



GROUP DECISION SUPPORT SYSTEM (GDSS)
PENENTUAN LOKASI PENEMPATAN ANJUNGAN
TUNAI MANDIRI MENGGUNAKAN METODE *ENTROPY*,
VIKOR DAN BORDA

SKRIPSI

oleh :

Nur Lailiana

112410101025

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015



GROUP DECISION SUPPORT SYSTEM (GDSS)
PENENTUAN LOKASI PENEMPATAN ANJUNGAN
TUNAI MANDIRI MENGGUNAKAN METODE *ENTROPY*,
VIKOR DAN BORDA

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi dan mencapai gelar Sarjana Komputer

oleh :

Nur Lailiana

112410101025

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT, yang telah memberikan kelancaran dan kemudahan dalam menyelesaikan skripsi ini;
2. Ayahanda Muryat dan Ibunda tercinta Kartini;
3. Adik Angga Noviansyah, Kakek Tukirin beserta keluarga, Kakek Musban beserta keluarga dan Kakak Pujiyanto beserta keluarga;
4. Saudara-saudaraku beserta seluruh keluarga besar;
5. Bangun Rizki Awangditama beserta keluarga yang selalu memberi motivasi;
6. Sahabatku bersama doa dan bantuannya;
7. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
8. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Allah akan meninggikan orang-orang yang beriman di antara kamu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat”. (*QS. Al-Mujadalah : 11*)¹



¹ Departemen Agama Republik Indonesia. 1998. *Al Qur'an dan Terjemahannya*. Semarang: PT Kumudasmoro Grafindo

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Nur Lailiana

NIM : 112410101025

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "*Group Decision Support System (GDSS) Penentuan Lokasi Penempatan Anjungan Tunai Mandiri Menggunakan Metode Entropy, VIKOR dan Borda*" adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 25 Juni 2015

Yang menyatakan,

Nur Lailiana

NIM 112410101025

SKRIPSI

***GROUP DECISION SUPPORT SYSTEM (GDSS) PENENTUAN LOKASI
PENEMPATAN ANJUNGAN TUNAI MANDIRI MENGGUNAKAN
METODE ENTROPY, VIKOR DAN BORDA***

oleh :

Nur Lailiana

NIM 112410101025

Pembimbing :

Dosen Pembimbing Utama : Anang Andrianto, ST., MT.

Dosen Pembimbing Pendamping : Nelly Oktavia Adiwijaya, S.Si., MT.

PENGESAHAN PEMBIMBING

Skripsi berjudul “*Group Decision Support System (GDSS) Penentuan Lokasi Penempatan Anjungan Tunai Mandiri Menggunakan Metode Entropy, VIKOR dan Borda*”, telah diuji dan disahkan pada :

hari, tanggal : Kamis, 25 Juni 2015

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

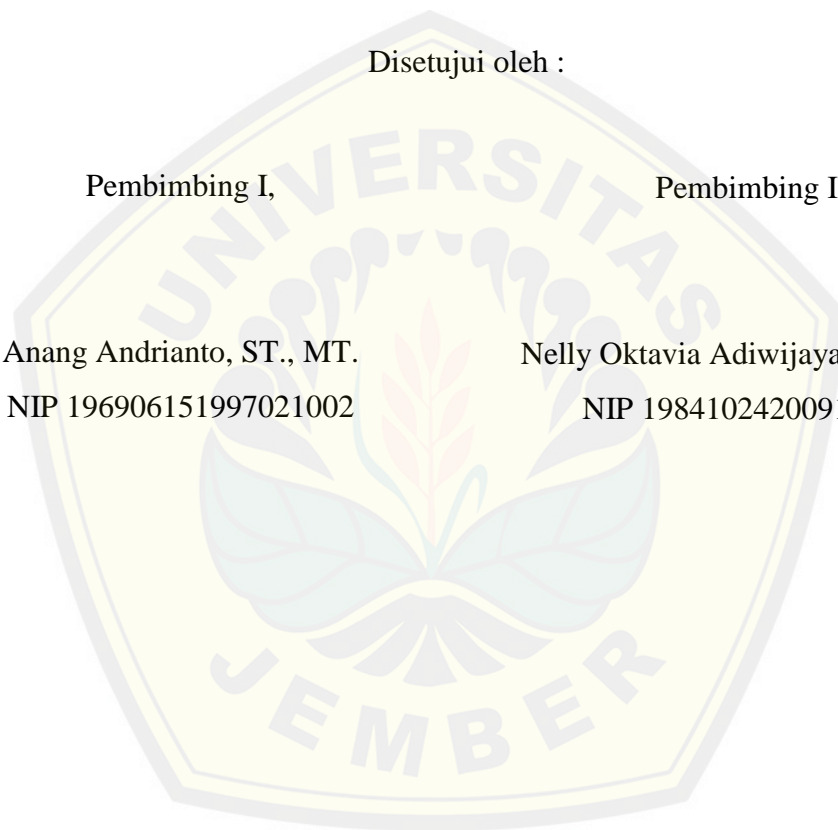
Pembimbing II,

Anang Andrianto, ST., MT.

Nelly Oktavia Adiwijaya, S.Si., MT.

NIP 196906151997021002

NIP 198410242009122008



PENGESAHAN

Skripsi berjudul “*Group Decision Support System (GDSS) Penentuan Lokasi Penempatan Anjungan Tunai Mandiri Menggunakan Metode Entropy, VIKOR dan Borda*”, telah diuji dan disahkan pada :

hari, tanggal : Kamis, 25 Juni 2015

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji :

Penguji I,

Penguji II,

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D
NIP 196704201992011001

Muhammad Arief Hidayat, S.Kom., M.Kom
NIP 198101232010121003

Mengesahkan
Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D
NIP 196704201992011001

RINGKASAN

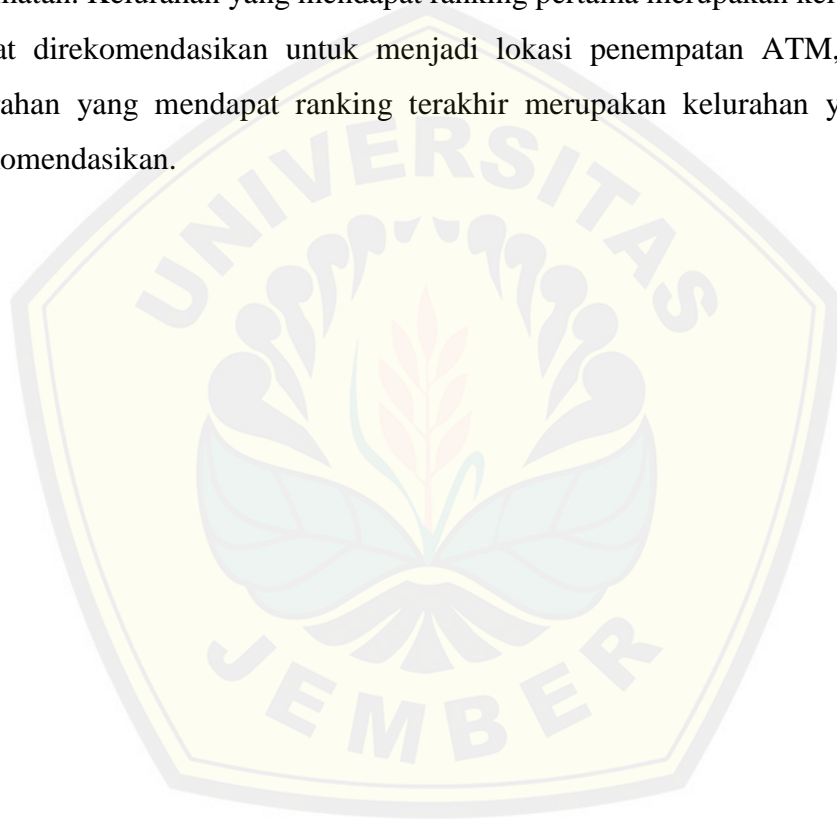
Group Decision Support System (GDSS) Penentuan Lokasi Penempatan Anjungan Tunai Mandiri Menggunakan Metode Entropy, VIKOR dan Borda;
Nur Lailiana; 112410101025; 463 halaman; Program Studi Sistem Informasi Universitas Jember.

Anjungan Tunai Mandiri (ATM) merupakan alat kasir otomatis yang menggantikan sebagian besar fungsi dari seorang *teller* tanpa dibatasi oleh waktu operasional bank. Banyaknya ATM di Kabupaten Jember membuat nasabah dapat melakukan transaksi pembayaran di berbagai lokasi. Akan tetapi tidak semua ATM dikunjungi oleh nasabah setiap harinya dikarenakan lokasi ATM tersebut jauh dari tempat tinggal dan tempat kerja nasabah atau nasabah merasa kurang nyaman dalam bertransaksi melalui ATM tersebut. Hal tersebut akan menyebabkan kerugian bagi pihak bank. Penentuan lokasi penempatan ATM menjadi sangatlah penting untuk mencegah kerugian-kerugian tersebut. PT. BRI (Persero) Tbk Cabang Jember merupakan salah satu bank yang mengalami kesulitan dalam menentukan lokasi penempatan ATM.

Pada penelitian ini dibangun *Group Decision Support System (GDSS) Penentuan Lokasi Penempatan ATM* menggunakan kriteria-kriteria yang ditentukan oleh para pengambil keputusan. Metode yang digunakan dalam sistem ini adalah metode *Entropy*, VIKOR dan Borda. Metode *Entropy* digunakan untuk menentukan bobot kriteria, metode VIKOR digunakan untuk perankingan lokasi oleh setiap pengambilan keputusan dan metode Borda digunakan untuk menghasilkan perankingan lokasi berdasarkan keputusan kelompok. Sistem ini memvisualisasikan hasil perankingan lokasi dan persebaran ATM BRI serta kriteria lainnya menggunakan Sistem Informasi Geografis.

Penelitian ini dilaksanakan dalam tiga tahap yaitu tahap pengumpulan data, tahap analisis dan tahap pengembangan sistem. Tahap pengumpulan data dilakukan melalui studi literatur dan wawancara kepada narasumber. Tahap analisis dilakukan dengan menentukan koordinat lokasi, *digitasi* dan analisis data ke dalam metode

sistem. Tahap pengembangan sistem dilakukan dengan menggunakan model *waterfall* yang dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program dan pengujian sistem. Sistem ini dibangun menggunakan bahasa pemrograman *Page Hypertext Pre-Processor (PHP)* dan *framework Code Igniter (CI)*. Berdasarkan hasil pengujian terhadap sistem, GDSS dapat memberikan rekomendasi lokasi penempatan ATM berdasarkan penilaian kelompok berupa perankingan kelurahan dari sebuah kecamatan. Kelurahan yang mendapat ranking pertama merupakan kelurahan yang sangat direkomendasikan untuk menjadi lokasi penempatan ATM, sedangkan kelurahan yang mendapat ranking terakhir merupakan kelurahan yang kurang direkomendasikan.



PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “*Group Decision Support System (GDSS) Penentuan Lokasi Penempatan Anjungan Tunai Mandiri Menggunakan Metode Entropy, VIKOR dan Borda*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Anang Andrianto S.T.,MT., selaku Dosen Pembimbing Utama dan Nelly Oktavia Adiwijaya, S.Si., MT. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Anang Andrianto S.T.,MT., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ayahanda Muryat dan Ibu Kartini yang telah sangat memotivasi penulis;
6. Adik Angga Noviansyah, Kakek Tukirin beserta keluarga, Kakek Musban beserta keluarga dan Kakak Pujianto beserta keluarga;
7. Saudara-saudaraku beserta seluruh keluarga besar;
8. Bangun Rizki Awangditama beserta keluarga yang selalu memberi motivasi;
9. Sahabat seperjuangan yang selalu menemani dan memberikan semangat Kadek, Ruka, Mita, Dinda, Devi, Piter, Jefta, Maulana, Emas dan Febri;
10. Keluarga besar NEFOTION semua mahasiswa Program Studi Sistem Informasi angkatan 2011 yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
11. Keluarga besar Himpunan Mahasiswa Sistem Informasi (HIMASIF) periode 2012-2013 Arbi Darmaja, Fitriyana Dewi, Ridwan Galih, Bangun R.

Awangditama, A. Haidar Rusdi, Feby Dinda Risvi, Ratih Pravitasari, Qilbaaini E.M, Anwar Syarifuddin F, Tri Rofiana, Gusti Dwi S, Hasan Jindan, Nindy Meidita E, Annisa Nandadiri, Siti Masruroh, Febrianto Rama Anji, Rizka R. Fitriah, Hani Almira, Pandu Dwi Luhur P, Risqa Farihatul J, Ahmad Syafiq Kamil, Dody Arief P, Selamat Hariyanto, Yudha Alif dan Ngesti Dania dan HIMASIF periode 2013-2014 Pheiter Indera Kusuma, Nindi Norya Tindra, Emas Achmad Mirza, Rizki Herdatullah, Yudi Candra K, Lintang Harianti, Ghausilia R, Tri Setyoko, Qilbaaini Effendy M, Fachrur Rijal, Risqa Farihatul Jannah, Ainul Ulfah Maulinda, Haris Arfan R, Rizki Vadilla, M. Rizki Firdaus, Rozi Rikza Akbar, Febrianto Rama A, M.Ainul Yaqin, M. Fieqih Hidayatullah, M.Fariz Aminullah, Khoirus Sahal, Dhevi Indriawati, Diah Ayu S, Yudha Herlambanag C.P, Hipolitus Kresna, Dany Dwi K dan Eky Angliara.

12. Keluarga besar kos-kosan 39 Erna Fifin, Kurnia, Rima, Wheny, Livara dan penghuni lain yang tidak dapat disebutkan satu persatu;
13. Keluarga besar BIDIKMISI Universitas Jember;
14. Keluarga besar Paskibra Universitas Jember angkatan 2011;
15. Direktorat Pembelajaran dan Kemahasiswaan Ditjen Dikti yang telah memberikan beasiswa bidikmisi kepada penulis sehingga dapat menyelesaikan pendidikan S1;
16. PT. Bank Rakyat Indonesia (Persero) Tbk Cabang Jember;
17. Bapak Edy, Ibu Enny, Icha, Fira, Ammar dan Mbak Seh yang telah banyak membantu penulis di Kabupaten Jember;
18. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 25 Juni 2015

Penulis

DAFTAR ISI

PERSEMBAHAN.....	ii
MOTO	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN.....	vii
RINGKASAN	viii
PRAKATA.....	x
DAFTAR ISI.....	xii
DAFTAR TABEL	xv
DAFTAR GAMBAR.....	xix
DAFTAR LAMPIRAN	xxii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan dan Manfaat	4
1.3.1 Tujuan	5
1.3.2 Manfaat	5
1.4 Ruang Lingkup	6
1.5 Sistematika Penulisan	6
BAB 2. TINJUAN PUSTAKA.....	8
2.1 <i>Decision Support System (DSS)</i>	8
2.2 <i>Group Decision Support System (GDSS)</i>	9
2.3 Sistem Informasi Geografis	10
2.3.1 <i>Google Earth</i>	10
2.3.2 <i>Google Maps API</i>	11
2.3.3 <i>Open Street Maps</i>	12
2.3.4 <i>Quantum GIS</i>	13
2.4 Metode <i>Entropy</i>	13

2.5	Metode VIKOR	15
2.6	Metode Borda	17
BAB 3. METODOLOGI PENELITIAN		18
3.1	Jenis Penelitian	18
3.2	Metode Penelitian	18
3.3	Tempat dan Waktu Penelitian	18
3.4	Tahapan Penelitian	19
3.5.1	Tahap Pengumpulan Data	20
3.5.2	Tahap Analisis	25
3.5.3	Tahap Pengembangan Sistem	64
BAB 4. PENGEMBANGAN SISTEM		70
4.1	<i>Statement Of Purpose</i>	70
4.2	Analisis Kebutuhan Sistem	71
4.2.1	Kebutuhan Fungsional	71
4.2.2	Kebutuhan Non-Fungsional	72
4.3	Desain Sistem	72
4.3.1	<i>Business Process</i>	72
4.3.2	<i>Usecase Diagram</i>	74
4.3.3	<i>Usecase Skenario</i>	77
4.3.4	<i>Activity Diagram</i>	95
4.3.5	<i>Sequence Diagram</i>	105
4.3.6	<i>Class Diagram</i>	130
4.3.7	<i>Entity Relationship Diagram</i>	130
4.4	Penulisan Kode Program	133
4.5	Pengujian Sistem	150
4.5.1	<i>White Box Testing</i>	150
4.5.2	<i>Black Box Testing</i>	176
BAB 5. HASIL DAN PEMBAHASAN		182
5.1	Hasil Penerapan Metode <i>Entropy</i> , VIKOR dan Borda Pada Proses Pengambilan Keputusan	182

5.2 Hasil Pembuatan GDSS Penentuan Lokasi Penempatan ATM	185
5.2.1 Fitur <i>Login</i>	185
5.2.2 Halaman Depan Admin	186
5.2.3 Fitur Mengelola Data <i>User</i>	187
5.2.4 Fitur Mengelola Data Kecamatan	187
5.2.5 Fitur Mengelola Data Kelurahan	188
5.2.6 Fitur Mengelola Data Kriteria	188
5.2.7 Fitur Mengelola Analisa DSS	189
5.3 Hasil Penerapan Metode <i>Entropy</i>, VIKOR dan Borda Pada Sistem	189
5.3.1 Pemilihan Kriteria dan Pemberian Bobot Awal Kriteria..	190
5.3.2 Perhitungan Bobot Kriteria Menggunakan Metode <i>Entropy</i>	191
5.3.3 Perhitungan Perankingan Lokasi Menggunakan Metode VIKOR	200
5.3.4 Perhitungan Keputusan Kelompok Menggunakan Metode Borda	211
5.4 Hasil Implementasi Sistem Informasi Geografis Pada Sistem	217
5.4.1 Implementasi <i>Google Maps</i> Pada Sistem	217
5.4.2 Implementasi <i>Open Street Maps</i> Pada Sistem	225
5.5 Hasil Perbandingan Proses Pengambilan Keputusan Secara Konvensional dan Menggunakan Sistem	228
BAB 6. PENUTUP	231
6.1 Kesimpulan	231
6.2 Saran	232
DAFTAR PUSTAKA	233
LAMPIRAN	236

DAFTAR TABEL

Tabel 3.1 Kriteria Pengambilan Keputusan oleh Pelaksana IT dan <i>E-Channel</i> ...	21
Tabel 3.2 Kriteria Pengambilan Keputusan oleh Asisten Manajer Pemasaran Dana	22
Tabel 3.3 Kriteria Pengambilan Keputusan oleh Pimpinan Bank.....	22
Tabel 3.4 Data Batasan Kriteria oleh Pimpinan Bank	23
Tabel 3.5 Data ATM BRI.....	24
Tabel 3.6 Kriteria dan Bobot Awal Pelaksana Bidang IT dan E-Channel pada analisa Kecamatan Kaliwates.....	32
Tabel 3.7 Kriteria dan Bobot Awal Asisten Manajer Pemasaran Dana pada analisa Kecamatan Kaliwates	32
Tabel 3.8 Kriteria dan Bobot Awal Pimpinan Bank pada Analisa Kecamatan Kaliwates.....	32
Tabel 3.9 Batasan Kriteria oleh Pelaksana IT dan <i>E-Channel</i>	33
Tabel 3.10 Batasan Kriteria oleh Asisten Manajer Pemasaran Dana.....	33
Tabel 3.11 Batasan Kriteria oleh Pimpinan Bank	34
Tabel 3.12 Metrik Nilai Kriteria Asli Pelaksana IT dan <i>E-Channel</i>	34
Tabel 3.13 Metrik Nilai Kriteria Asli Asisten Manajer Pemasaran Dana.....	34
Tabel 3.14 Metrik Nilai Kriteria Asli Pimpinan Bank.....	35
Tabel 3.15 Metrik Kriteria Pelaksana IT dan <i>E-Channel</i>	35
Tabel 3.16 Metrik Kriteria Asisten Manajer Pemasaran Dana	35
Tabel 3.17 Metrik Kriteria Pimpinan Bank.....	36
Tabel 3.18 Metrik Kriteria Ternormalisasi Pelaksana IT dan <i>E-Channel</i>	37
Tabel 3.19 Metrik Kriteria Ternormalisasi Asisten Manajer Pemasaran Dana	37
Tabel 3.20 Metrik Kriteria Ternormalisasi Pimpinan Bank.....	38
Tabel 3.21 Metrik Probabilitas Kriteria Pelaksana IT dan E-Channel.....	39
Tabel 3.22 Metrik Probabilitas Kriteria Asisten Manajer Pemasaran Dana	39
Tabel 3.23 Metrik Probabilitas Kriteria Pimpinan Bank	39
Tabel 3.24 Hasil Nilai <i>Entropy</i> Kriteria Pelaksana IT dan <i>E-Channel</i>	40
Tabel 3.25 Hasil Nilai <i>Entropy</i> Kriteria Asisten Manajer Pemasaran Dana.....	40

Tabel 3.26 Hasil Nilai <i>Entropy</i> Kriteria Pimpinan Bank	41
Tabel 3.27 Hasil Perhitungan Lamda Kriteria Pelaksana IT dan <i>E-Channel</i>	42
Tabel 3.28 Hasil Perhitungan Lamda Kriteria Asisten Manajer Pemasaran Dana	42
Tabel 3.29 Hasil Perhitungan Lamda Kriteria Pimpinan Bank.....	42
Tabel 3.30 Hasil Perhitungan Lamda Dikalikan Bobot Awal Kriteria Pelaksana IT dan <i>E-Channel</i>	43
Tabel 3.31 Hasil Pembobotan <i>Entropy</i> Kriteria dari Pelaksana IT dan <i>E-Channel</i>	44
Tabel 3.32 Hasil Pembobotan <i>Entropy</i> Kriteria dari Asisten Manajer Pemasaran Dana	44
Tabel 3.33 Hasil Pembobotan <i>Entropy</i> Kriteria dari Pimpinan Bank.....	44
Tabel 3.34 Nilai Solusi Ideal Positif dan Negatif Kriteria dari Pelaksana IT dan <i>E-Channel</i>	46
Tabel 3.35 Nilai Solusi Ideal Positif dan Negatif Kriteria dari Asisten Manajer Pemasaran Dana	46
Tabel 3.36 Nilai Solusi Ideal Positif dan Negatif Kriteria dari Pimpinan Bank ...	46
Tabel 3.37 Metrik Nilai Ideal Terbobot dari Pelaksana IT dan <i>E-Channel</i>	48
Tabel 3.38 Metrik Nilai Ideal Terbobot dari Asisten Manajer Pemasaran Dana..	48
Tabel 3.39 Metrik Nilai Ideal Terbobot dari Pimpinan Bank	48
Tabel 3.40 Nilai Solusi Tertinggi dan Terendah Kelurahan dari Pelaksana IT dan <i>E-Channel</i>	49
Tabel 3.41 Nilai Solusi Tertinggi dan Terendah Kelurahan dari Asisten Manajer Pemasaran Dana	49
Tabel 3.42 Nilai Solusi Tertinggi dan Terendah Kelurahan dari Pimpinan Bank	50
Tabel 3.43 Nilai Rasio Perankingan (Q) Kelurahan dari Pelaksana IT dan <i>E-Channel</i>	51
Tabel 3.44 Nilai Rasio Perankingan (Q) Kelurahan dari Asisten Manajer Pemasaran Dana	51
Tabel 3.45 Nilai Rasio Perankingan (Q) Kelurahan dari Pimpinan Bank	51
Tabel 3.46 Hasil <i>Decreasing Order</i> Nilai S dari Pelaksana IT dan <i>E-Channel</i>	52
Tabel 3.47 Hasil <i>Decreasing Order</i> Nilai R dari Pelaksana IT dan <i>E-Channel</i> ...	52

Tabel 3.48 Hasil <i>Decreasing Order</i> Nilai Q dari Pelaksana IT dan <i>E-Channel</i> ...	52
Tabel 3.49 Hasil <i>Decreasing Order</i> Nilai S dari Asisten Manajer Pemasaran Dana	53
Tabel 3.50 Hasil <i>Decreasing Order</i> Nilai R dari Asisten Manajer Pemasaran Dana	53
Tabel 3.51 Hasil <i>Decreasing Order</i> Nilai Q dari Asisten Manajer Pemasaran Dana	53
Tabel 3.52 Hasil <i>Decreasing Order</i> Nilai S dari Pimpinan Bank.....	54
Tabel 3.53 Hasil <i>Decreasing Order</i> Nilai R dari Pimpinan Bank	54
Tabel 3.54 Hasil <i>Decreasing Order</i> Nilai Q dari Pimpinan Bank	54
Tabel 3.55 Hasil Perankingan VIKOR Penilaian Pelaksana IT dan <i>E-Channel</i> ...	56
Tabel 3.56 Perhitungan Kondisi Kedua Perankingan Kelurahan Berdasarkan Penilaian Asisten Manajer.....	57
Tabel 3.57 Hasil Perankingan VIKOR Penilaian Asisten Manajer Pemasaran Dana	58
Tabel 3.58 Perhitungan Kondisi Kedua Perankingan Kelurahan Berdasarkan Penilaian Pimpinan Bank	58
Tabel 3.59 Hasil Perankingan VIKOR Penilaian Pimpinan Bank	59
Tabel 3.60 Hasil Perankingan Lokasi Masing-Masing Pengambil Keputusan.....	59
Tabel 3.61 <i>Point</i> Borda Setiap Kelurahan oleh Setiap Pengambil Keputusan	60
Tabel 3.62 Hasil Perhitungan Metode Borda.....	61
Tabel 3.63 Hasil Perankingan Lokasi Penempatan ATM pada Kecamatan Kaliwates.....	61
Tabel 4.1 Definisi Aktor	75
Tabel 4.2 Definisi <i>Usecase</i>	76
Tabel 4.3 <i>Usecase</i> Skenario Mengelola Analisa DSS	78
Tabel 4.4 <i>Usecase</i> Skenario Memroses Pengambilan Keputusan	82
Tabel 4.5 Kode Program <i>function</i> proses1_tambah_bobot_awal().....	133
Tabel 4.6 Kode Program <i>function</i> metode_entropy()	134
Tabel 4.7 Kode Program <i>function</i> metode_vikor()	138
Tabel 4.8 Kode Program <i>function</i> metode_borda().....	146

Tabel 4.9 <i>Test Case</i> Fitur Mengelola Analisa DSS	153
Tabel 4.10 <i>Test Case</i> Fitur Memilih Kriteria dan Menentukan Bobot Awal.....	158
Tabel 4.11 <i>Test Case</i> Perhitungan Metode <i>Entropy</i>	169
Tabel 4.12 Hasil Pengujian <i>Black Box Usecase</i> Mengelola Analisa DSS.....	177
Tabel 5.1 Kode Program Penyimpanan Nilai Kepentingan ke Dalam Metrik 2 Dimensi	191
Tabel 5.2 Kode Program Pembuatan Metrik Kriteria	191
Tabel 5.3 Kode Program Pembuatan Metrik Kriteria Ternormalisasi	192
Tabel 5.4 Kode Program Pembuatan Metrik Probabilitas Kriteria	194
Tabel 5.5 Kode Program Pembuatan Metrik Natural Logaritma Probabilitas Kriteria	195
Tabel 5.6 Kode Program Menghitung Nilai <i>Entropy</i>	195
Tabel 5.7 Kode Program Perhitungan Lamda Kriteria dan Bobot <i>Entropy</i>	196
Tabel 5.8 Kode Program Menyimpan Bobot <i>Entropy</i> ke <i>Database</i>	199
Tabel 5.9 Kode Program Menghitung Nilai Solusi Ideal Positif dan Negatif Setiap Kriteria	200
Tabel 5.10 Kode Program Menghitung Nilai Solusi Tertinggi dan Terendah Setiap Alternatif	201
Tabel 5.11 Kode Program Menghitung Indeks VIKOR	202
Tabel 5.12 Kode Program Mengurutkan Nilai S,R dan Q dengan <i>Decreasing</i> <i>Order</i>	204
Tabel 5.13 Kode Program Pemeriksaan Kondisi Pemberian Ranking Kelurahan	207
Tabel 5.14 Kode Program Menyimpan Hasil Perankingan Lokasi ke <i>Database</i>	211
Tabel 5.15 Kode Program Penentuan <i>Point Borda</i>	212
Tabel 5.16 Kode Program Perhitungan <i>Borda Count</i>	213
Tabel 5.17 Kode Program Pemberian Ranking Kelurahan Metode Borda.....	214
Tabel 5.18 Kode Program Menampilkan Peta Persebaran ATM dan Kategori..	218
Tabel 5.19 Kode Program Fitur Melihat Hasil GDSS	221

DAFTAR GAMBAR

Gambar 2.1 <i>Request</i> URL Pada <i>Google Maps API</i>	12
Gambar 2.2 Tahapan Perhitungan Metode <i>Entropy</i>	14
Gambar 2.3 Tahapan Perhitungan Metode VIKOR.....	16
Gambar 2.4 Contoh Pemberian Ranking Alternatif Pada Metode VIKOR	16
Gambar 3.1 Diagram Alir Tahapan Penelitian.....	19
Gambar 3.2 Diagram <i>Block</i> Tahapan Analisis.....	25
Gambar 3.3 Hasil Koordinat Lokasi ATM BRI.....	26
Gambar 3.4 Hasil <i>Digitasi</i> Kecamatan di Kabupaten Jember.....	27
Gambar 3.5 Hasil <i>Digitasi</i> Kelurahan di Kabupaten Jember	27
Gambar 3.6 Diagram Alir GDSS Penentuan Lokasi ATM.....	31
Gambar 3.7 Alur Pengambilan Keputusan pada Sistem	63
Gambar 3.8 Model <i>Waterfall</i> (Sommerville, 2001).....	64
Gambar 3.9 Contoh Listing Program	67
Gambar 3.10 Contoh Grafik Alir	67
Gambar 4.1 <i>Business Process</i> GDSS Penentuan Lokasi Penempatan ATM.....	73
Gambar 4.2 <i>Usecase</i> Diagram GDSS Penentuan Lokasi Penempatan ATM	74
Gambar 4.3 <i>Activity</i> Diagram Mengelola Analisa DSS	97
Gambar 4.4 <i>Activity</i> Diagram Memroses Pengambilan Keputusan.....	101
Gambar 4.5 <i>Sequence</i> Diagram Mengelola Analisa DSS	113
Gambar 4.6 <i>Sequence</i> Diagram Memroses Pengambilan Keputusan	122
Gambar 4.7 <i>Class</i> Diagram GDSS Penentuan Lokasi Penempatan ATM.....	131
Gambar 4.8 ERD GDSS Penentuan Lokasi Penempatan ATM.....	132
Gambar 4.9 <i>Listing</i> Program Menambah Analisa DSS	150
Gambar 4.10 <i>Listing</i> Program Mengubah Analisa DSS	151
Gambar 4.11 <i>Listing</i> Program Menghapus Analisa DSS.....	151
Gambar 4.12 Diagram Alir Mengelola Analisa DSS.....	152
Gambar 4.13 Grafik Alir Fitur Mengelola Analisa DSS.....	152
Gambar 4.14 <i>Listing</i> Program Fitur Memilih Kriteria dan Menentukan Bobot Awal Kriteria.....	156

Gambar 4.15 Diagram Alir Fitur Memilih Kriteria dan Menentukan Bobot Awal	156
Gambar 4.16 Grafik Alir Fitur Memilih Kriteria dan Menentukan Bobot Awal	157
Gambar 4.17 <i>Listing</i> Program Perhitungan Metode <i>Entropy</i>	162
Gambar 4.18 Diagram Alir Perhitungan Metode <i>Entropy</i>	163
Gambar 4.19 Grafik Alir Perhitungan Metode <i>Entropy</i>	164
Gambar 5.1 Tampilan Halaman <i>Login</i>	186
Gambar 5.2 Tampilan Halaman Depan Admin	186
Gambar 5.3 Tampilan Fitur Mengelola Data <i>User</i>	187
Gambar 5.4 Tampilan Fitur Mengelola Data Kecamatan	187
Gambar 5.5 Tampilan Fitur Mengelola Data Kelurahan	188
Gambar 5.6 Tampilan Fitur Mengelola Data Kriteria	188
Gambar 5.7 Tampilan Fitur Mengelola Analisa DSS	189
Gambar 5.8 Tampilan Pemilihan Kriteria dan dan Pemberian Bobot Awal Kriteria	190
Gambar 5.9 Tampilan Metrik Kriteria Pada Sistem	192
Gambar 5.10 Tampilan Metrik Kriteria Ternormalisasi Pada Sistem	193
Gambar 5.11 Tampilan Metrik Probabilitas Kriteria Pada Sistem	194
Gambar 5.12 Tampilan Metrik Natural Logaritma Probabilitas Kriteria Pada Sistem	195
Gambar 5.13 Tampilan Nilai <i>Entropy</i> Setiap Kriteria Pada Sistem	196
Gambar 5.14 Tampilan Nilai Lamda dan Bobot <i>Entropy</i> Pada Sistem	197
Gambar 5.15 Hasil Pembobotan <i>Entropy</i> Pelaksana IT dan <i>E-Channel</i> Pada Sistem	198
Gambar 5.16 Hasil Pembobotan <i>Entropy</i> Asisten Manajer Pemasaran Dana Pada Sistem	198
Gambar 5.17 Hasil Pembobotan <i>Entropy</i> Pimpinan Bank Pada Sistem	199
Gambar 5.18 Tampilan Nilai Solusi Ideal Positif dan Negatif Pada Sistem	201
Gambar 5.19 Tampilan Hasil Perhitungan Nilai S, R dan Q Pada Sistem	203
Gambar 5.20 Tampilan Urutan Nilai S, R dan Q Pada Sistem	206

Gambar 5.21 Hasil Perankingan Lokasi Pelaksana IT dan <i>E-Channel</i> Pada Sistem	210
Gambar 5.22 Hasil Perankingan Lokasi Asisten Manajer Pemasaran Dana Pada Sistem.....	210
Gambar 5.23 Hasil Perankingan Lokasi Pimpinan Bank Pada Sistem	210
Gambar 5.24 Tampilan Metrik Ranking Kelurahan Pada Sistem.....	212
Gambar 5.25 Tampilan Metrik <i>Point</i> Borda Pada Sistem.....	213
Gambar 5.26 Hasil Keputusan Kelompok Analisa Kecamatan Kaliwates Pada Sistem.....	215
Gambar 5.27 Tampilan Hasil Keputusan Kelompok Dalam Bentuk Tabel dan Grafik	216
Gambar 5.28 Tampilan Hasil Keputusan Kelompok Dalam Bentuk Peta.....	216
Gambar 5.29 Tampilan Fitur Melihat Informasi Peta.....	220
Gambar 5.30 Tampilan Fitur Melihat Informasi Peta Pada Kecamatan dan Kategori Tertentu	220
Gambar 5.31 Tampilan Fitur Melihat Hasil GDSS.....	224
Gambar 5.32 Tampilan <i>Info Window</i> Kelurahan Pada Fitur Melihat Hasil GDSS	225
Gambar 5.33 Tampilan <i>Open Street Map</i> Pada Fitur Menambah Data Kecamatan	226
Gambar 5.34 Tampilan Tipe dan Koordinat Gambar Menggunakan OSM.....	226
Gambar 5.35 Tampilan Koordinat Area Dalam Bentuk <i>Geometry</i> Pada OSM..	227
Gambar 5.36 Tampilan Hasil Fitur Menambah Kecamatan Menggunakan OSM	227

DAFTAR LAMPIRAN

Lampiran A. <i>Usecase</i> Skenario.....	236
Lampiran B. <i>Activity</i> Diagram	280
Lampiran C. <i>Sequence</i> Diagram	308
Lampiran D. Kode Program.....	373
Lampiran E. Pengujian <i>Black Box</i>	441
Lampiran F. Hasil Pembuatan Sistem.....	459



BAB 1. PENDAHULUAN

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.

1.1 Latar Belakang

Anjungan Tunai Mandiri (ATM) merupakan alat kasir otomatis tanpa orang, ditempatkan di dalam atau di luar pekarangan bank yang sanggup mengeluarkan uang tunai dan menangani transaksi-transaksi perbankan secara rutin, seperti penyetoran, penarikan uang, transfer antar rekening, pelunasan atau pembayaran tagihan kartu kredit (Emirson, 2002). ATM menggantikan sebagian besar fungsi dari seorang *teller* dan sangat membantu nasabah dalam mendapatkan pelayanan bank tanpa dibatasi oleh waktu operasional bank.

Banyaknya ATM di Kabupaten Jember membuat nasabah dapat melakukan transaksi pembayaran dengan cepat dan nyaman di berbagai lokasi. Lokasi penempatan ATM biasanya merupakan tempat keramaian, unit-unit bank dan tempat institusi yang bekerjasama dengan pihak bank. Akan tetapi tidak semua ATM dikunjungi oleh nasabah setiap harinya. Terdapat ATM yang jarang dikunjungi oleh nasabah dikarenakan lokasi ATM tersebut jauh dari tempat tinggal dan tempat kerja nasabah atau nasabah merasa kurang nyaman dalam bertransaksi melalui ATM tersebut. Hal tersebut akan menyebabkan kerugian bagi pihak bank yaitu kerugian dari pembangunan ATM yang kurang efisien dalam penempatannya. Kurangnya total transaksi yang dilakukan mesin ATM juga akan memberikan kerugian pada pihak bank. Penentuan lokasi penempatan ATM menjadi sangatlah penting untuk mencegah kerugian-kerugian tersebut.

PT. BRI (Persero) Tbk Cabang Jember merupakan salah satu bank yang mengalami kesulitan dalam menentukan lokasi penempatan ATM. Penentuan lokasi penempatan ATM masih dilakukan secara konvensional, sehingga membutuhkan waktu dalam melakukan proses penyeleksian. Permasalahan

selanjutnya adalah bagaimana pengambil keputusan melakukan penilaian terhadap lokasi yang tepat, apakah lokasi sudah memenuhi kriteria yang ditetapkan sebagai lokasi penempatan ATM atau belum. Proses penentuan lokasi penempatan ATM tidak dilakukan oleh satu pihak saja melainkan dilakukan oleh 3 pihak yaitu Pelaksana Bidang IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan PT. BRI (Persero) Tbk Cabang Jember. Para pengambil keputusan memiliki kriteria dan penilaian yang berbeda sehingga menghasilkan keputusan yang berbeda. Dari keputusan yang berbeda tersebut akan sangat sulit disatukan untuk menjadi sebuah keputusan bersama yang akan direkomendasikan untuk sebuah lokasi penempatan ATM.

Penelitian sebelumnya berjudul “Sistem Pendukung Keputusan Kelompok Penentuan Kelayakan Lokasi Pemukiman” yang dilakukan oleh Mutammimul Ula dan SN. Azhari. Mereka membuat sebuah *Group Decision Support System* (GDSS) penentuan kelayakan lokasi pemukiman pada pesisir pantai Kota Lhokseumawe. GDSS digunakan dalam penelitian tersebut karena pengambilan keputusan dilakukan oleh tiga pihak yang memiliki kriteria dan penilaian yang berbeda dan manfaatnya adalah untuk meningkatkan kualitas keputusan sehingga penilaian yang digunakan lebih objektif. (Ula & Azhari, 2013)

Penelitian lainnya dengan judul “Sistem Pengambilan Keputusan Penempatan ATM (*Automated Teller Machine*) Berdasarkan Penyebaran Nasabah di Surabaya Menggunakan GIS” dilakukan oleh Purba Setyo Baskoro, Ahmad Syauqi Ahsan dan Arif Basofi. Sistem Informasi Geografis pada penelitian tersebut sangat cocok digunakan untuk memvisualisasikan dan menentukan letak strategis dan berpotensi untuk pengembangan ATM (Baskoro, Ahsan, & Basofi, 2011).

Berdasarkan penelitian tersebut dapat disimpulkan bahwa untuk mengakomodasi kriteria dan penilaian pengambil keputusan yang berbeda-beda sehingga menghasilkan penilaian yang objektif maka GDSS dapat digunakan untuk menentukan lokasi penempatan ATM pada PT. BRI (Persero) Tbk Cabang Jember. Sedangkan GIS dapat digunakan untuk memvisualisasikan kriteria, penyebaran ATM BRI di Kabupaten Jember dan hasil keputusan kelompok agar informasi yang dihasilkan bisa digambarkan secara jelas dan menarik.

Secara umum proses pengambilan keputusan pada PT. BRI (Persero) Tbk Cabang Jember terbagi menjadi dua tahapan yaitu pengambilan keputusan secara individu oleh masing-masing pengambil keputusan dan tahap penentuan keputusan kelompok. Metode pengambilan keputusan dapat digunakan untuk membantu menyelesaikan kedua tahapan tersebut. Pengambilan keputusan oleh pengambil keputusan menghasilkan ranking kelurahan yang cocok untuk menjadi lokasi penempatan ATM dapat menggunakan metode pembobotan kriteria dan metode perankingan alternatif. Penggunaan metode *voting* dapat digunakan untuk pengambilan keputusan kelompok.

Metode pembobotan kriteria yang sering digunakan pada GDSS adalah metode *Entropy*. Penelitian tentang GDSS yang menggunakan metode *Entropy* antara lain penelitian yang dilakukan oleh Mutammimul Ula dan SN. Azhari yang sudah dijelaskan sebelumnya serta penelitian pembuatan GDSS penentuan lokasi KKN oleh Elfitri Dwi Rahardianti. Pada penelitian tersebut metode *Entropy* dapat menghasilkan bobot kriteria berdasarkan karakteristik data sekaligus dapat mengakomodasi penilaian subyektif dari pengambil keputusan. (Rahardianti & Azhari, 2013)

Metode perankingan alternatif yang digunakan pada GDSS merupakan salah satu jenis dari MCDM (*Multiple Criteria Decision Making*) yaitu pemilihan alternatif terbaik dari beberapa alternatif yang ada berdasarkan kriteria tertentu. Salah satu metode MCDM yang dapat digunakan untuk menghasilkan ranking alternatif setiap pengambil keputusan adalah metode perankingan VIKOR karena VIKOR fokus pada ranking dan seleksi serta dapat mengatasi situasi dimana pengambil keputusan tidak dapat menentukan preferensinya pada saat awal desain sistem. (Huang, Tzeng, & Liu, 2009)

Metode *voting* yang sering digunakan untuk menghasilkan keputusan kelompok pada GDSS adalah metode Borda. Metode Borda ini digunakan pada penelitian yang dilakukan oleh Bhakti Susilo dan SN Azhari dengan judul "GDSS Penentuan Lokasi Shelter Baru Transjogja Menggunakan Metode Brown-Gibson dan Borda" serta pada penelitian yang dilakukan oleh Mahdi dengan judul "Sistem Pendukung Keputusan Kelompok Pemilihan Dayah Terbaik Menggunakan Metode

Promethee dan Metode Borda”. Metode Borda selalu menghasilkan hasil yang pasti dan menggunakan informasi peringkat yang menyeluruh dan sistematis serta mengatasi metode *voting* lainnya dimana pilihan yang tidak berada pada ranking pertama akan secara otomatis dihapus. (Reilly, 2002)

Berdasarkan kelebihan-kelebihan metode tersebut dan tingkat penggunaannya dalam GDSS maka peneliti memutuskan untuk menggunakan metode *Entropy*, VIKOR dan Borda pada penelitian ini. Metode *Entropy* digunakan untuk menentukan bobot kriteria, metode VIKOR digunakan untuk perankingan lokasi oleh setiap pengambilan keputusan dan metode Borda digunakan untuk menghasilkan keputusan kelompok. Sistem ini diharapkan dapat membantu para pengambil keputusan pada PT. BRI (Persero) Tbk Cabang Jember untuk menghasilkan keputusan kelompok dalam menentukan lokasi penempatan ATM yang sesuai dengan kriteria.

1.2 Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang mendefinisikan beberapa permasalahan yang harus diselesaikan dalam penulisan ini, yaitu:

1. Bagaimana menerapkan metode *Entropy*, VIKOR dan Borda pada proses pengambilan keputusan penentuan lokasi penempatan Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember?
2. Bagaimana merancang dan membangun sebuah *Group Decision Support System* (GDSS) penentuan lokasi penempatan Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember menggunakan metode *Entropy*, VIKOR dan Borda?

1.3 Tujuan dan Manfaat

Tujuan dan manfaat dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan.

1.3.1 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Menerapkan metode *Entropy*, VIKOR dan Borda pada proses pengambilan keputusan penentuan lokasi penempatan Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember.
2. Merancang dan membangun sebuah *Group Decision Support System* (GDSS) penentuan lokasi penempatan Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember menggunakan metode *Entropy*, VIKOR dan Borda.

1.3.2 Manfaat

Manfaat yang ingin didapatkan dari penelitian ini adalah :

1. Manfaat Akademis
Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.
2. Manfaat bagi peneliti
 - a. Mengetahui bagaimana proses penerapan metode *Entropy*, VIKOR dan Borda pada proses pengambilan keputusan penentuan lokasi penempatan ATM pada PT. BRI (Persero) Tbk Cabang Jember.
 - b. Membantu instansi dalam pengambilan keputusan penentuan lokasi penempatan ATM.
3. Manfaat bagi objek penelitian
 - a. Memberikan inovasi baru kepada instansi tempat penelitian mengenai penggunaan *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM.
 - b. Membantu instansi untuk melakukan pengambilan keputusan kelompok mengenai penentuan lokasi penempatan ATM secara cepat dan lebih objektif.

1.4 Ruang Lingkup

Ruang lingkup dalam penulisan ini merupakan batasan – batasan masalah dalam penulisan. Penulis memberi batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penulisan dan pembuatan sistem informasi. Berikut adalah batasan masalah yang dicantumkan:

1. Persebaran ATM BRI, ATM kompetitor, tempat keramaian dan kriteria lain yang berhubungan dengan pengambilan keputusan serta hasil dari pengambilan keputusan divisualisasikan ke dalam peta menggunakan Sistem Informasi Geografis.
2. Sistem ini menentukan lokasi penempatan ATM PT. BRI (Persero) Tbk Cabang Jember hanya dalam lingkup Kabupaten Jember.
3. Penentuan lokasi penempatan ATM ditentukan oleh para pengambil keputusan, sistem hanya memberikan rekomendasi lokasi penempatan ATM yang sesuai dengan kriteria.
4. Perhitungan bobot kriteria pada sistem ini menggunakan metode *Entropy*, perankingan rekomendasi lokasi setiap pengambil keputusan menggunakan metode VIKOR dan *voting* kelompok menggunakan metode Borda.

1.5 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

1. Pendahuluan
Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup studi dan sistematika penulisan.
2. Tinjauan Pustaka
Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.
3. Metodologi Penelitian
Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.
4. Pengembangan Sistem
Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.

6. Penutup

Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.



BAB 2. TINJUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku literatur, jurnal dan internet. Berikut merupakan teori-teori yang digunakan dan dibahas dalam penelitian:

2.1 *Decision Support System (DSS)*

Pengambilan keputusan adalah pemilihan beberapa tindakan alternatif yang ada untuk mencapai satu atau beberapa tujuan yang telah ditetapkan. Untuk membantu proses pengambilan keputusan diperlukan sebuah sistem pendukung keputusan. Sistem pendukung keputusan merupakan kumpulan dari aplikasi perangkat lunak dan perangkat keras yang saling terintegrasi yang merupakan tulang punggung dari sebuah proses pengambilan keputusan. (Waghmode & P.P.Jamsandekar, 2004)

Komponen Sistem Pengambil Keputusan dapat dibangun dari subsistem berikut ini:

1. Subsistem Manajemen Data (*Data Management System*), meliputi basis data-basis data yang berisi data yang relevan dengan keadaan dan dikelola oleh *software* yang disebut DBMS (*Database Management System*).
2. Subsistem Manajemen Model (*Model Management Subsystem*), berupa sebuah paket *software* yang berisi model-model finansial, statistik, *management science*, atau model kuantitatif, yang menyediakan kemampuan analisa dan *software management* yang sesuai.
3. Subsistem Manajemen Pengetahuan (*Knowledge Management Subsystem*) merupakan subsistem (*optional*) yang dapat mendukung subsistem lain atau berlaku sebagai komponen yang berdiri sendiri (*independent*).

4. Subsistem Antarmuka Pengguna (*User Interface Subsystem*), merupakan subsistem yang dapat dipakai oleh *user* untuk berkomunikasi dan memberi perintah (menyediakan *user interface*).
5. Pengguna (*user*), termasuk di dalamnya adalah pengguna (*user*), *manager*, dan pengambil keputusan. (Turban, McLean, & Wetherbe, 2005)

2.2 *Group Decision Support System (GDSS)*

Group Decision Support System (GDSS) merupakan sebuah sistem berbasis komputer yang mendukung sekelompok orang yang tergabung dalam satu tugas atau sasaran yang sama dan memiliki satu sarana tertentu yang berfungsi saling menghubungkan orang-orang yang ada dalam kelompok tersebut (McLeod & Schell, 2007).

Menurut Turban Efraim karakteristik yang dimiliki oleh GDSS antara lain:

1. GDSS mempunyai tujuan untuk mendukung proses pengambilan keputusan secara berkelompok dengan menyediakan otomatisasi subproses menggunakan alat teknologi informasi.
2. GDSS merupakan sistem informasi yang dirancang secara khusus, bukan hanya konfigurasi komponen-komponen sistem yang sudah ada. GDSS dapat dirancang untuk menyelesaikan salah satu tipe masalah atau level kelompok pengambil keputusan yang bervariasi.
3. GDSS mendorong pembentukan ide-ide, pemecahan masalah dan kebebasan berekspresi yang didalamnya terdapat mekanisme pencegah pengembangan perilaku negatif dari kelompok seperti konflik yang merusak, salah komunikasi dan perdebatan masalah dalam kelompok. (Turban, 2011)

Beberapa keuntungan dari sebuah sistem pendukung keputusan kelompok atau *Group Decision Support System (GDSS)*, diantaranya dapat mendukung aktivitas kelompok, memberikan kecepatan hasil akhir keputusan yang akan diperoleh, setiap pengambil keputusan dapat mengkalkulasikan keputusan secara *online* pada saat yang bersamaan atau tidak bersamaan. (Pressman, 2002)

2.3 Sistem Informasi Geografis

Sistem Informasi Geografis merupakan suatu sistem informasi yang digunakan untuk memasukkan, menyimpan, memanggil kembali, mengolah, menganalisa dan menghasilkan data bereferensi geografi atau data geospasial, untuk mendukung pengambilan keputusan dalam perencanaan dan pengolahan seperti penggunaan lahan, sumber daya alam, lingkungan transportasi, perencanaan fasilitas kota dan pelayanan umum lainnya. Komponen kunci dalam SIG adalah sistem komputer, data geospasial (data atribut) dan pengguna. SIG menghasilkan 2 jenis data yaitu data spasial dan data atribut. Data spasial merupakan data yang menggambarkan aspek keruangan seperti letak suatu wilayah yang terdiri dari titik, garis dan area. Data atribut merupakan data yang menggambarkan aspek deskripsi atau penjelasan tentang fenomena dalam bentuk kata-kata, angka atau tabel (Nurdiansyah, Basofi, & Fariza, 2010). Beberapa perangkat lunak atau layanan yang digunakan untuk mengimplementasikan Sistem Informasi Geografis yang digunakan di dalam penelitian dijelaskan pada sub-bab di bawah ini.

2.3.1 *Google Earth*

Google Earth adalah sebuah layanan peta berbasis citra satelit yang disediakan oleh mesin pencari terkenal *Google*. *Google Earth* memetakan bumi dari gambar citra yang dikumpulkan dari pemetaan satelit, fotografi udara dan *globe GIS 3D*. Citra yang digunakan oleh aplikasi ini adalah data resolusi rendah (MODIS), menengah (Land-sat) dan juga resolusi tinggi dan sangat tinggi. Dengan citra resolusi tinggi tersebut, kenampakan objek menjadi sangat jelas dan objek mudah diidentifikasi. (Winarso, Trisakti, & K.Harsanugraha, 2010)

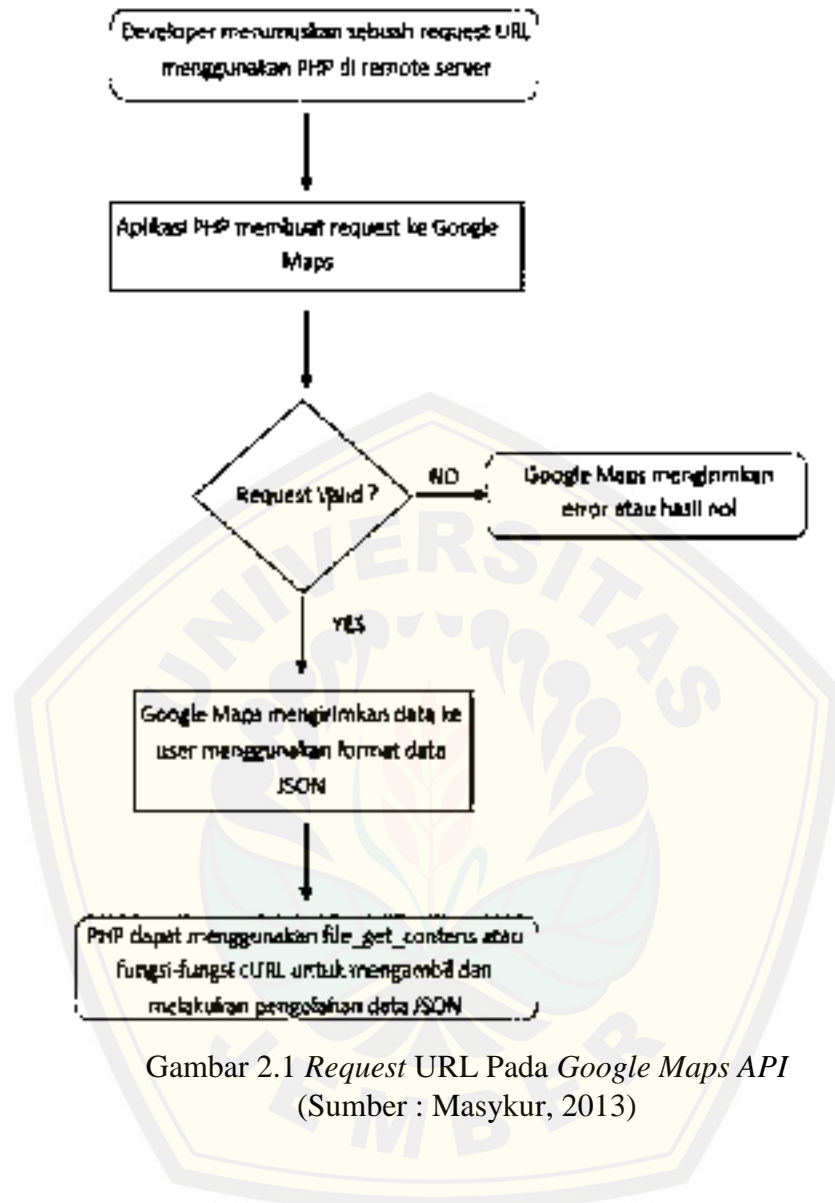
Ketelitian citra *Google Earth* pada penelitian Gatot Winarso, Bambang Trisakti dan Wawan K. Harsanugraha dengan judul “*Google Earth* sebagai Referensi Alternatif Titik Kontrol Tanah” diuji dengan menentukan penyimpangan yang terjadi antara koordinat pada citra *Google Earth* dengan koordinat hasil pengukuran *Ground Control Point* (GCP). Rata-rata penyimpangan antara koordinat objek pada citra digital *Google Earth* dengan koordinat GCP pengukuran lapangan adalah sebesar 5,41 m. Dengan penyimpangan rata-rata tersebut

disimpulkan bahwa *Google Earth* dapat digunakan untuk menentukan koordinat suatu lokasi.

2.3.2 *Google Maps API*

Google Maps adalah layanan gratis yang sudah disediakan oleh *Google* yang dapat digunakan untuk mengimplementasikan Sistem Informasi Geografis berbasis *website*. Salah satu keuntungan dari penggunaan *Google Maps* adalah layanan yang *free* dan bisa dikembangkan sesuai dengan keinginan karena *Google* sendiri sudah menyediakan *library* bagi para pengembang yang ingin memanfaatkan layanan *Google Maps* tersebut. *Google Maps* menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia. *Google Maps* juga menawarkan pencarian suatu tempat dan rute perjalanan. (Masykur, 2014)

Google Maps API adalah sebuah layanan (*service*) atau *library* berbentuk *javascript* yang diberikan oleh *Google* kepada para pengguna untuk memanfaatkan *Google Map* dalam mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta dan menambah konten melalui berbagai jenis *service* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi *enterprise* di dalam *websitenya*. Pengguna dapat memanfaatkan layanan-layanan yang ditawarkan oleh *Google Maps* setelah melakukan registrasi dan mendapatkan *Google Maps API Key*. *Google* memberikan layanan ini secara gratis di seluruh dunia. Untuk membangun aplikasi yang memanfaatkan *Google Maps* maka di *website* maka akan digunakan *Google Maps Javascript API v3* yang memiliki keunggulan lebih cepat dari versi sebelumnya (Developers, 2015). Data dari *Google Maps API* yang dikirimkan ke *user* setelah *user* melakukan *request* ke *Google Maps* merupakan data yang mempunyai format JSON. JSON merupakan format pertukaran data yang ringan dan memiliki format yang sederhana dan *universal* sehingga semua bahasa pemrograman *modern* mendukung struktur data ini. Terdapat beberapa langkah dalam proses *request URL* pada *Google Maps API*. Bagan *request URL* pada *Google Maps API* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Request URL Pada Google Maps API (Sumber : Masykur, 2013)

2.3.3 Open Street Maps

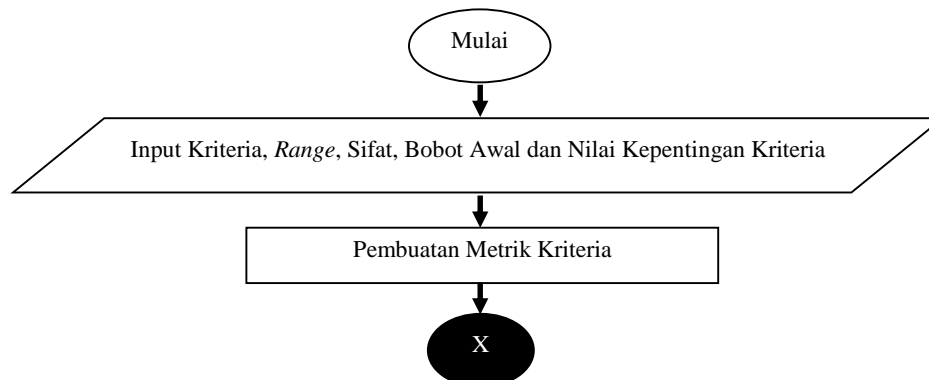
Open Street Maps (OSM) adalah layanan yang menyediakan informasi geospasial dengan tema yang beragam seperti infrastruktur transportasi meliputi jalan, jalur kereta api, sungai, serta informasi lain seperti tempat-tempat penting, bangunan, fitur alam dan penggunaan lahan, garis pantai dan batas administratif. Terdapat beberapa fungsi di OSM antara lain *editing*, memanipulasi dan menyimpan peta terhadap informasi geospasial. (Niam, Suprayogi, & Awaludin, 2014)

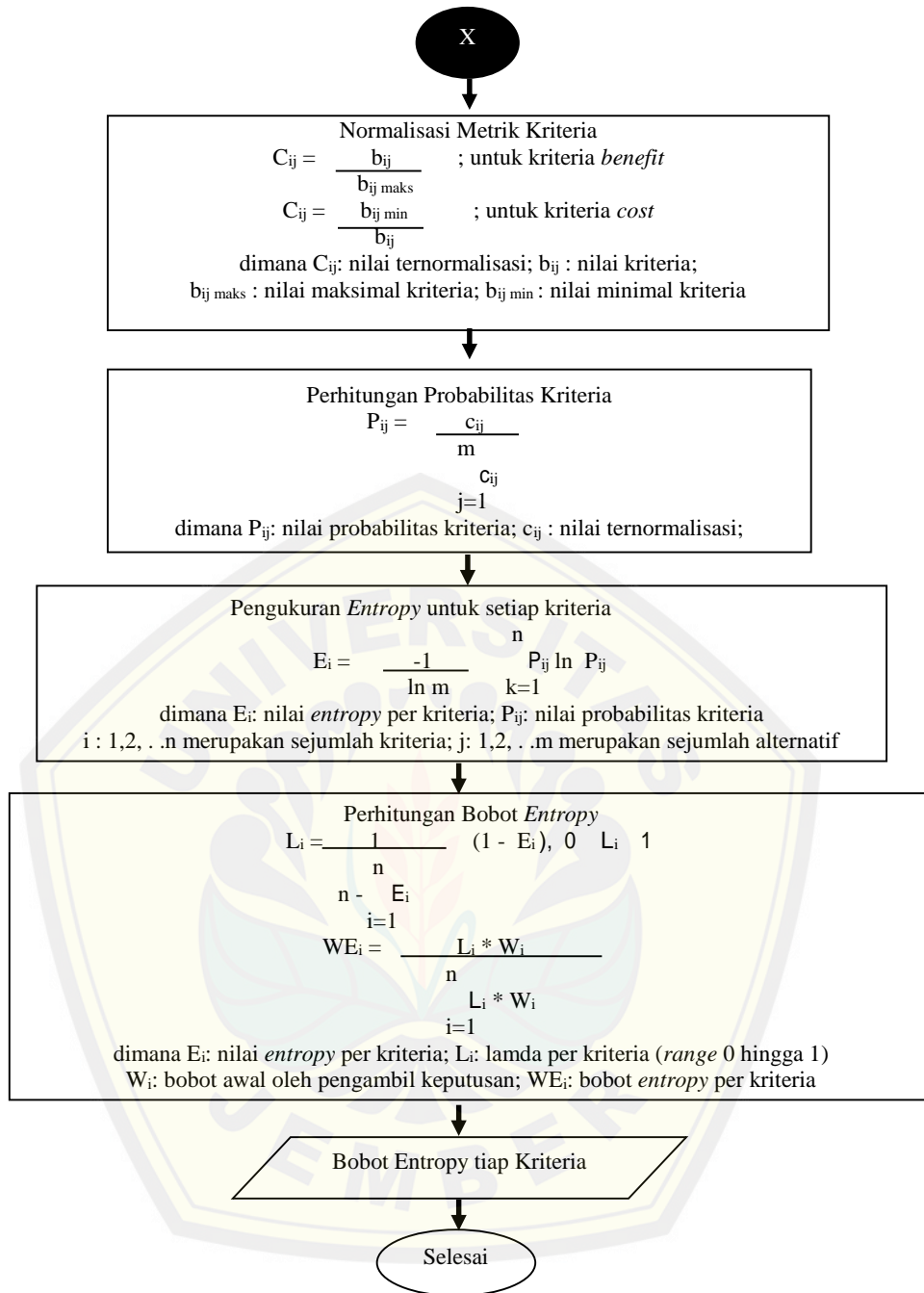
2.3.4 *Quantum* GIS

Quantum GIS adalah perangkat lunak berbasis desktop pada SIG yang digunakan untuk menyediakan data, melihat, mengedit, dan kemampuan analisis. Data-data tentang koordinat lokasi diolah dengan menggunakan *Quantum* GIS sehingga bisa disimpan dalam format shp. *Quantum* GIS juga dapat digunakan untuk proses *digitasi*. *Digitasi* merupakan proses pengkonversian data spasial pada peta ke dalam format digital. Data-data yang dikumpulkan akan dimaukkan dan diubah menjadi sebuah data spasial sehingga dapat diolah menjadi suatu peta digital oleh *Quantum* GIS. (Muliawan, 2014)

2.4 Metode *Entropy*

Metode *entropy* merupakan metode untuk menentukan suatu bobot kriteria. Kelebihan dari metode *entropy* dibandingkan metode pembobotan lainnya adalah metode *entropy* menggunakan pendekatan subyektif dan obyektif sehingga menghasilkan bobot kriteria berdasarkan karakteristik data sekaligus dapat mengakomodasi preferensi subyektif dari pengambil keputusan. Sedangkan metode pembobotan lainnya kebanyakan hanya menggunakan pendekatan subyektif (bobot berasal dari preferensi pengambil keputusan) atau pendekatan obyektif (bobot berasal dari perhitungan sistem). Selain memberikan obyektifitas pembobotan kriteria, satuan maupun batasan data yang digunakan pada metode ini tidak perlu sama dan metode *entropy* juga dapat memecahkan masalah tentang diskriminasi diantara sekumpulan data (Jamila & Hartati, 2011). Langkah-langkah dari perhitungan metode *entropy* digambarkan pada Gambar 2.2.





Gambar 2.2 Tahapan Perhitungan Metode Entropy

Kriteria yang digunakan pada perhitungan metode *Entropy* memiliki 2 sifat yaitu *benefit* dan *cost*. Kriteria *benefit* adalah kriteria dimana pengambil keputusan menginginkan nilai maksimum di antara seluruh nilai alternatif sedangkan kriteria *cost* merupakan kriteria dimana pengambil keputusan menginginkan nilai minimum di antara seluruh nilai alternatif. (Wirjayasaputra & Hartati, 2012)

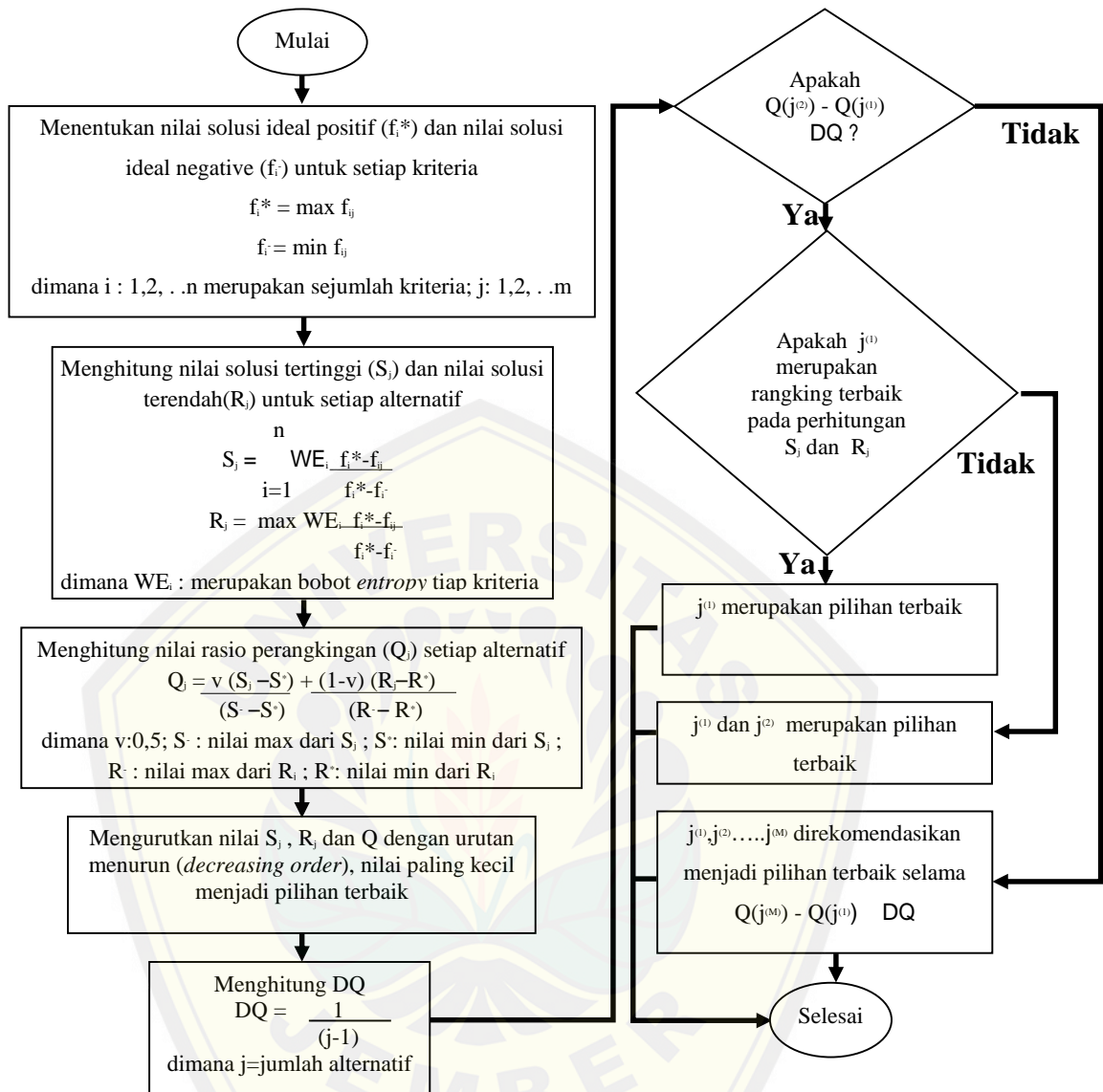
Hasil dari perhitungan *entropy* merupakan bobot *entropy* sebenarnya dari sebuah kriteria. Pada dasarnya, kriteria dengan variasi nilai tertinggi akan mendapatkan bobot tertinggi. Artinya, kriteria tersebut dianggap mampu untuk membedakan performansi tiap alternatif. (Triyanti & Gadis, 2008)

2.5 Metode VIKOR

Metode VIKOR (*Vlsekriterijumska Optimizacija Kompromisno Resenje in Serbia*) merupakan salah satu metode MCDM (*Multi Criteria Decision Making*) yang memiliki prosedur perhitungan sederhana dengan pertimbangan kedekatan antar alternatif yang ideal maupun tidak ideal. Metode ini dapat digunakan untuk meranking banyak kriteria baik itu kualitatif maupun kuantitatif. Metode VIKOR dibangun untuk optimasi multikriteria pada sistem yang kompleks (Opricovic & Tzeng, 2004). Tahapan perhitungan metode VIKOR dijelaskan pada Gambar 2.3.

Nilai v pada rumus untuk mencari nilai rasio perankingan merupakan bobot dari *majority of criteria*, biasanya diberi nilai 0.5. Pemberian ranking pada alternatif mengikuti konsep ranking seperti pada Gambar 2.4. Apabila ada 4 alternatif dengan terdapat 2 alternatif yang mempunyai ranking 1 maka urutan ranking alternatif keseluruhan adalah 1,1,3 dan 4. (Bondor, Kacso, Lenghel, Istrate, & Muresan, 2013)

Hasil dari metode VIKOR berupa urutan perankingan alternatif mulai dari ranking terbaik sampai terendah. Keistimewaan VIKOR adalah dapat digunakan untuk pengambil keputusan dengan kriteria yang lebih dari satu, khususnya situasi dimana pengambil keputusan tidak dapat menentukan preferensinya pada saat awal desain sistem. Solusi yang ditawarkan pada metode VIKOR adalah pertimbangan antara nilai utilitas maksimal grup (S_j) dan nilai regret minimal individu (R_j) yang saling bertentangan (Huang, Tzeng, & Liu, 2009).



Gambar 2.3 Tahapan Perhitungan Metode VIKOR

Parameters	Position	Parameters	Position	Parameters	Position
Serum adiponectin	1	Gender	6	Total cholesterol	12
Triglycerides	2	HbA1c	8	Waist circumference	
Duration of diabetes	3	Fasting glucose	9	BMI	15
SBP					
Age					
Metabolic syndrome	6	CRP	12	Catalase	17
		LDL cholesterol	12	HDL cholesterol	17
				MDA	18

SBP = systolic blood pressure; DBP = diastolic blood pressure; CRP = C-reactive protein; LDL = Low Density Lipoprotein; HDL = High Density Lipoprotein; BMI = body mass index; MDA = malondialdehyde; HbA1c = glycated hemoglobin

Gambar 2.4 Contoh Pemberian Ranking Alternatif Pada Metode VIKOR

(Sumber : Bondor, Kacso, Lenghel, Istrate, & Muresan, 2013)

2.6 Metode Borda

Borda merupakan suatu metode yang digunakan pada pengambilan keputusan kelompok untuk pemilihan *single winner* atau pun *multiple winner*, dimana pemberi suara (*voters*) melakukan perankingan terhadap kandidat yang disusun berdasarkan pilihan (*preference*) (Ratna, 2000). Metode Borda selalu menghasilkan hasil yang pasti dan menggunakan informasi peringkat yang menyeluruh dan sistematis serta mengatasi metode *voting* lainnya dimana pilihan yang tidak berada pada ranking pertama akan secara otomatis dihapus (Reilly, 2002).

Borda menentukan pemenang yang mempunyai *point* terbanyak. Borda memberikan suatu jumlah *point* tertentu untuk masing-masing kandidat sesuai dengan ranking yang telah diatur oleh masing-masing pengambil keputusan. Pemenang akan ditentukan oleh banyaknya jumlah *point* yang dikumpulkan atau diperoleh dari masing-masing kandidat. Pemilihan dan perhitungan dilakukan dengan cara memberikan sejumlah *point* kepada masing-masing kandidat misalnya ada 5 kandidat pada suatu pemilihan maka masing-masing kandidat menerima 5 *point* untuk kandidat ranking pertama, 4 *point* untuk kandidat ranking kedua dan seterusnya, kandidat terakhir menerima 1 *point*, dengan kata lain dimana n kandidat akan menerima n *point* untuk pilihan pertama, $n-1$ untuk pilihan kedua, $n-2$ untuk pilihan 3 dan $n-4$ untuk pilihan 5. (Mahdi, 2013)

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang jenis penelitian, metode penelitian, tempat dan waktu penelitian serta tahapan penelitian yang dilakukan untuk membangun *Group Decision Support System* (GDSS) Penentuan Lokasi Penempatan Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember.

3.1 Jenis Penelitian

Jenis penelitian yang dilakukan merupakan penelitian kualitatif dan kuantitatif. Penelitian kualitatif dalam penelitian ini adalah pada tahap analisis studi kasus dan pengumpulan data yaitu dengan melakukan wawancara kepada pengambil keputusan mengenai indikator untuk menentukan lokasi penempatan ATM sesuai kriteria pada PT. BRI (Persero) Tbk Cabang Jember. Penggunaan metode kuantitatif adalah pada tahap pengumpulan dan analisis data dalam bentuk angka serta penelitian ini mengkaji teori yang sudah ada sebelumnya.

3.2 Metode Penelitian

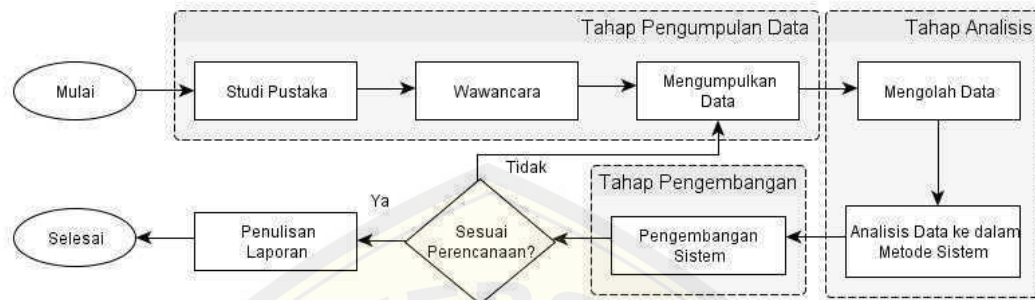
Metode penelitian yang digunakan pada penelitian ini adalah penelitian survei. Penelitian ini merupakan penelitian survei karena memenuhi beberapa karakter utama penelitian survei yaitu informasi dikumpulkan dari sekelompok besar orang untuk mendeskripsikan beberapa aspek atau karakteristik tertentu seperti kemampuan, sikap, kepercayaan, pengetahuan dari populasi dan juga informasi dikumpulkan melalui pengajuan pertanyaan.

3.3 Tempat dan Waktu Penelitian

Penelitian dilakukan di PT. Bank Rakyat Indonesia (Persero) Tbk Cabang Jember yang beralamat di Jl. Ahmad Yani No.1 Jember. Waktu penelitian dilakukan selama 4 (empat) bulan, dimulai pada bulan Maret 2015 sampai dengan bulan Juni 2015.

3.4 Tahapan Penelitian

Penelitian ini dilakukan dalam beberapa tahap diantaranya tahap pengumpulan data, tahap analisis dan tahap pengembangan sistem. Tahapan penelitian digambarkan dalam diagram alir seperti pada Gambar 3.1.



Gambar 3.1 Diagram Alir Tahapan Penelitian

Tahap awal dari penelitian ini adalah tahap pengumpulan data. Peneliti melakukan studi pustaka yang terkait dengan masalah yang ada. Wawancara kepada pihak-pihak terkait untuk mendapatkan data-data yang diperlukan dalam pembuatan system dilakukan setelah melakukan studi pustaka. Tahap kedua setelah tahap pengumpulan data adalah tahap analisis. Tahap analisis dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data lalu menganalisis data ke dalam metode yang digunakan oleh sistem. Tahap ketiga dari penelitian adalah tahap pengembangan sistem yang dimulai dari menganalisis kebutuhan sampai melakukan pengujian pada sistem. Tahap selanjutnya adalah memeriksa apakah pengujian pada tahap pengembangan sistem sesuai dengan perencanaan atau tidak. Penelitian akan kembali ke tahap pengumpulan data apabila hasil pengujian tidak sesuai perencanaan sedangkan jika hasil pengujian telah sesuai dengan apa yang direncanakan maka akan berlanjut ke tahap penulisan laporan dan penelitian telah selesai. Penjelasan lebih detail tentang tahapan penelitian akan dijelaskan pada sub-bab berikut ini.

3.5.1 Tahap Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan melalui beberapa teknik pengumpulan data yaitu:

1. Studi Literatur

Studi Literatur yang digunakan berasal dari buku, jurnal, internet, dokumen-dokumen yang terkait dengan *Decision Support System*, *Group Decision Support System*, Sistem Informasi Geografis, *Google Maps API*, *Openstreet Maps*, *Google Earth*, *Quantum GIS*, metode *Entropy*, metode VIKOR dan metode Borda.

2. Wawancara

Wawancara dilakukan kepada pengambil keputusan penentuan lokasi penempatan ATM di PT. BRI (Persero) Tbk Cabang Jember. Para pengambil keputusan tersebut antara lain Pelaksana Bidang IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan Bank. Data-data dari hasil wawancara yang digunakan untuk membangun sistem adalah data pengambil keputusan, data kriteria pengambilan keputusan, data batasan kriteria, dan data lokasi ATM BRI di Kabupaten Jember.

Data-data yang dikumpulkan dari hasil studi literatur dan wawancara meliputi data pengambil keputusan, data kriteria pengambilan keputusan, data batasan kriteria, data ATM BRI di Kabupaten Jember dan data nilai kriteria. Penjelasan tentang data-data tersebut dijelaskan pada sub-bab di bawah ini.

3.5.1.1 Data Pengambil Keputusan

Data pengambil keputusan merupakan biodata diri dari masing-masing pengambil keputusan. Biodata dari Pelaksana Bidang IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan PT. BRI (Persero) Tbk Cabang Jember terdiri dari nama lengkap, *personal number*, tanggal lahir, nama ibu, jenis kelamin, alamat, jabatan, dan nomor hp. Pengambil keputusan yang berasal dari pelaksana Bidang IT dan *E-Channel* maupun bagian Pemasaran Dana dapat berganti sesuai

kebijakan dari pimpinan PT. BRI (Persero) Tbk Cabang Jember. Data pengambil keputusan ini digunakan untuk mengatur akses masuk ke dalam sistem dan mengatur proses pengambilan keputusan.

3.5.1.2 Data Kriteria Pengambilan Keputusan

Data kriteria pengambilan keputusan merupakan data kriteria-kriteria yang digunakan dalam menentukan lokasi penempatan ATM. Kriteria dari masing-masing pengambil keputusan berbeda-beda dan dapat berubah sesuai dengan keinginan dari pengambil keputusan. Kriteria yang ditetapkan oleh para pengambil keputusan dari hasil wawancara merupakan kriteria awal yang masih digunakan saat ini dalam pengambilan keputusan. Kriteria pengambilan keputusan oleh Pelaksana IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan PT. BRI (Persero) Tbk Cabang Jember dapat dilihat pada Tabel 3.1, Tabel 3.2 dan Tabel 3.3.

Tabel 3.1 Kriteria Pengambilan Keputusan oleh Pelaksana IT dan *E-Channel*

No.	Kode Kriteria	Kriteria	Sifat Kriteria	Keterangan
1.	CA1	Jumlah ATM BRI	<i>Cost</i>	Merupakan jumlah ATM BRI yang ada di setiap kelurahan
2.	CA2	Jumlah Transaksi ATM	<i>Benefit</i>	Merupakan jumlah transaksi ATM yang ada di setiap kelurahan
3.	CA3	Jumlah Tempat Keramaian	<i>Benefit</i>	Merupakan jumlah tempat keramaian (perumahan, pusat perbelanjaan, rumah sakit, pantai) yang ada di setiap kelurahan
4.	CA4	Jumlah Penduduk	<i>Benefit</i>	Merupakan jumlah penduduk dari setiap kelurahan
5.	CA5	Jumlah Nasabah	<i>Benefit</i>	Merupakan jumlah nasabah dari Bank BRI setiap kelurahan
6.	CA6	Jumlah ATM Kompetitor	<i>Benefit</i>	Merupakan jumlah ATM kompetitor dari Bank BRI (Bank BCA

dan Bank Mandiri)
setiap kelurahan

Tabel 3.2 Kriteria Pengambilan Keputusan oleh Asisten Manajer Pemasaran Dana

No.	Kode Kriteria	Kriteria	Sifat Kriteria	Keterangan
1.	CB1	Jumlah Transaksi ATM	<i>Benefit</i>	Merupakan jumlah transaksi ATM yang ada di setiap kelurahan
2.	CB2	Jumlah Nasabah yang Mempunyai Kartu ATM	<i>Benefit</i>	Merupakan jumlah nasabah ATM BRI yang mempunyai kartu ATM yang ada di setiap kelurahan
3.	CB3	Jumlah Tempat Keramaian	<i>Benefit</i>	Merupakan jumlah tempat keramaian (perumahan, pusat perbelanjaan, rumah sakit, pantai) yang ada di setiap kelurahan

Tabel 3.3 Kriteria Pengambilan Keputusan oleh Pimpinan Bank

No.	Kode Kriteria	Kriteria	Sifat Kriteria	Keterangan
1.	CC1	Jumlah Transaksi ATM	<i>Benefit</i>	Merupakan jumlah transaksi ATM yang ada di setiap kelurahan
2.	CC2	Jumlah Instansi yang Bekerjasama	<i>Benefit</i>	Merupakan jumlah instansi yang bekerjasama dengan Bank BRI yang ada di setiap kelurahan
3.	CC3	Jumlah Tempat Keramaian	<i>Benefit</i>	Merupakan jumlah tempat keramaian (perumahan, pusat perbelanjaan, rumah sakit, pantai) yang ada di setiap kelurahan

Terdapat beberapa kriteria yang diajukan oleh para pengambil keputusan pada Tabel 3.1, Tabel 3.2 dan Tabel 3.3. Pelaksana IT dan *E-Channel* mempunyai

6 kriteria dengan satu kriteria CA1 bersifat *cost*, sedangkan Asisten Manajer Pemasaran Dana dan Pimpinan PT. BRI (Persero) Tbk Cabang Jember masing-masing mempunyai 3 kriteria yang semuanya bersifat *benefit*.

3.5.1.3 Data Batasan Kriteria

Data batasan kriteria merupakan batasan-batasan kriteria yang telah ditentukan oleh para pengambil keputusan yang menggambarkan *range* dari nilai masing-masing kriteria dengan tingkat kepentingan yang berbeda. Setiap kriteria yang ditentukan oleh para pengambil keputusan selalu memiliki batasan dan nilai kepentingan. Nilai kepentingan menggambarkan apakah nilai tersebut penting atau tidak penting dalam perhitungan pengambilan keputusan. Nilai kepentingan dimulai dari nilai 1 sampai 4 sesuai dengan kesepakatan antar pengambil keputusan. Data batasan kriteria yang telah ditentukan oleh Pimpinan Bank dapat dilihat pada Tabel 3.4.

Tabel 3.4 Data Batasan Kriteria oleh Pimpinan Bank

No.	Kode Kriteria	Kriteria	Batasan	Nilai Kepentingan
1.	CC1	Jumlah Transaksi	0 x 5999	1
		ATM	6000 x 10000	2
			10001 x 15000	3
			15001 x 1000000	4
2.	CC2	Jumlah Instansi yang Bekerjasama	0 x 1	1
			2 x 3	2
			4 x 5	3
			6 x 500	4
3.	CC3	Jumlah Tempat Keramaian	0 x 1	1
			2 x 3	2
			4 x 5	3
			6 x 100000	4

3.5.1.4 Data ATM BRI

Data ATM BRI merupakan data lokasi ATM yang dimiliki Bank BRI di Kabupaten Jember. Bank BRI memiliki 113 buah ATM yang tersebar di berbagai daerah di Kabupaten Jember. Beberapa data ATM BRI dapat dilihat pada Tabel 3.5.

Tabel 3.5 Data ATM BRI

No	User ID	Lokasi	Alamat
1.	21958	Alfamart Basuki Rahmat	Jl. Basuki Rachmad 155 Tegal Besar - Kaliwates - Jember
2.	21959	Alfamart Dr. Soebandi	Jl. Dr. Soebandi 259 Patrang - Jember
3.	21960	Alfamart Karimata	Jl. Karimata 77 Sumpersari - Jember
4.	21969	Indomaret Tanggul 2	Jl. HOS Cokro Aminoto 3 Tanggul - Jember
5.	21970	Indomaret Baratan	Jl. Slamet Riyadi Jember

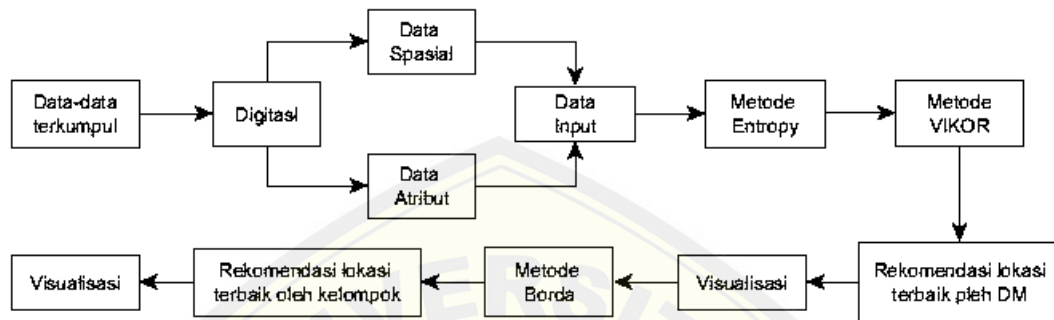
3.5.1.5 Data Nilai Kriteria

Data nilai kriteria merupakan nilai dari data-data yang berkaitan dengan kriteria pengambilan keputusan. Data-data tersebut antara lain data jumlah transaksi ATM BRI setiap kelurahan, data tempat keramaian yaitu perumahan, pusat perbelanjaan, rumah sakit dan pantai yang ada di Kabupaten Jember, data jumlah penduduk setiap kelurahan, data jumlah nasabah Bank BRI setiap kelurahan, data ATM bank kompetitor yaitu ATM Bank BCA dan Bank Mandiri, data jumlah nasabah Bank BRI yang mempunyai kartu ATM serta data jumlah instansi yang bekerja sama dengan Bank BRI pada setiap kelurahan di Kabupaten Jember.

Data nilai kriteria yang dikumpulkan berasal dari PT. BRI (Persero) Tbk Cabang Jember, Badan Pusat Statistik (BPS) Kabupaten Jember dan internet. Data yang dikumpulkan dari PT. BRI (Persero) Tbk Cabang Jember adalah data jumlah transaksi ATM BRI, data jumlah nasabah Bank BRI, data jumlah nasabah Bank BRI yang mempunyai kartu ATM dan data jumlah instansi yang bekerja sama dengan Bank BRI. Pengumpulan data dari BPS Kabupaten Jember adalah data kecamatan, data kelurahan, dan data jumlah penduduk setiap kelurahan. Data yang diambil dari internet antara lain data ATM bank kompetitor dan data tempat keramaian di Kabupaten Jember. Data ATM Mandiri diambil dari *website* resmi Bank Mandiri yaitu www.bankmandiri.co.id dan data ATM BCA diambil dari www.bca.co.id. Data rumah sakit dan tempat wisata diambil dari *website* milik Kabupaten Jember yaitu www.jemberkab.go.id sedangkan data perumahan diambil dari www.perumahanmu.com.

3.5.2 Tahap Analisis

Tahap analisis dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data. Langkah selanjutnya adalah menganalisa data dengan menggunakan metode *Entropy*, VIKOR dan Borda. Proses analisis dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram *Block* Tahapan Analisis

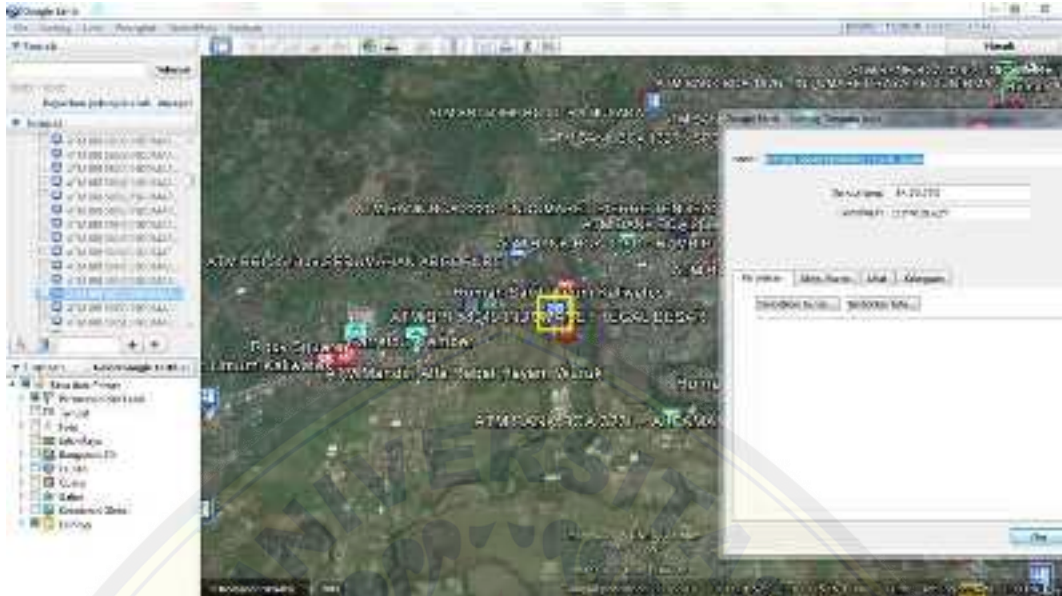
Langkah pertama dalam tahap analisis data ini adalah mengelompokkan data ke dalam dua jenis data yaitu data *spatial* dan data *atribut* karena data yang digunakan untuk membangun sistem ini ada yang akan diolah dan akan ditampilkan ke dalam peta. Cara untuk mengelompokkan data menjadi data *spatial* dan *atribut* adalah dengan mendapatkan koordinat lokasi data dan memrosesnya melalui proses digitasi. Tahapan analisis data sesuai dengan Gambar 3.2 akan dijelaskan pada subbab berikut ini.

3.5.2.1 Penentuan Koordinat Lokasi

Koordinat lokasi menunjukkan suatu lokasi di dalam peta. Data yang akan ditampilkan ke dalam peta adalah data ATM BRI, data ATM bank kompetitor, data tempat keramaian, data kecamatan dan data kelurahan. Penentuan koordinat lokasi data dalam penelitian ini menggunakan *software Google Earth*.

Langkah-langkah menentukan koordinat lokasi pada *Google Earth* adalah mencari lokasi data yang akan ditentukan koordinatnya pada permukaan bumi, lalu memberikan tanda titik jika lokasi itu merupakan titik, tanda garis jika lokasi merupakan garis dan tanda area jika lokasi berbentuk area. Setelah itu *Google Earth* otomatis akan menampilkan koordinat lokasi dari titik, garis maupun area tersebut

hasil dari pengambilan koordinat oleh citra satelit. Hasil dari penentuan koordinat lokasi data menggunakan *Google Earth* dapat dilihat pada Gambar 3.3.



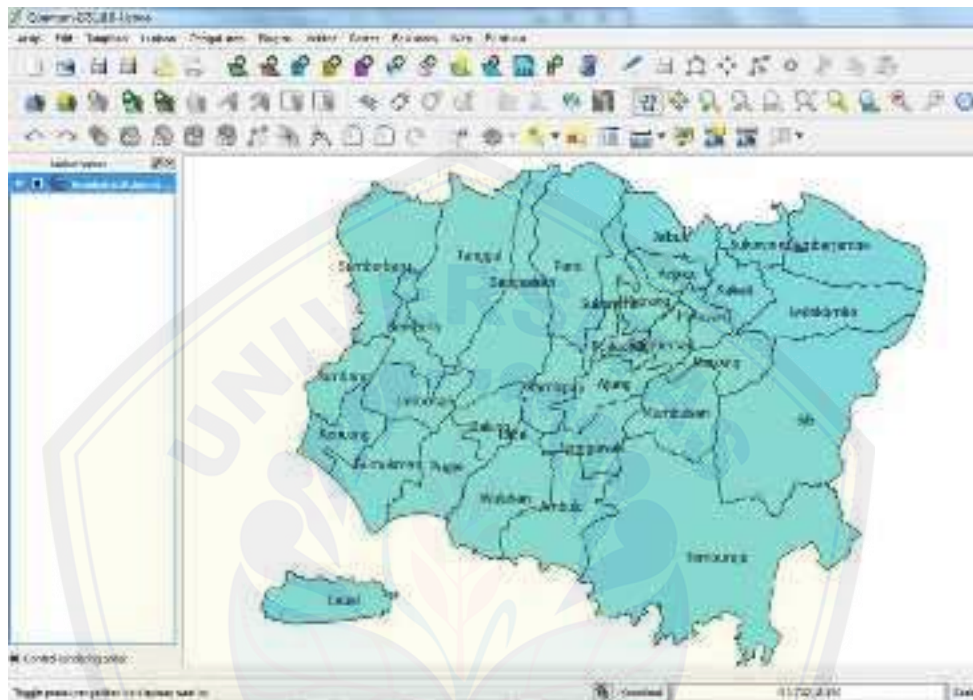
Gambar 3.3 Hasil Koordinat Lokasi ATM BRI

Hasil penentuan koordinat lokasi ATM BRI 58049 Indomaret Tegal Besar pada Gambar 3.3 adalah garis lintang $8^{\circ}11'0.37''S$ dan garis bujur $113^{\circ}40'28.62''T$. Baik lokasi yang berbentuk titik, garis maupun area semuanya memiliki koordinat dan cara penentuannya sama dengan langkah-langkah yang sudah dijelaskan di atas. Data yang dihasilkan oleh *Google Earth* masih dalam format kml sedangkan data yang dapat dikonversi agar dapat dimasukkan ke dalam *database* harus data dengan format shp sehingga digunakan *software* Quantum GIS untuk mengkonversi data tersebut. Quantum GIS di dalam penelitian ini juga digunakan untuk *digitasi* peta kecamatan dan kelurahan di Kabupaten Jember untuk mendapatkan data kecamatan dan kelurahan dalam bentuk digital.

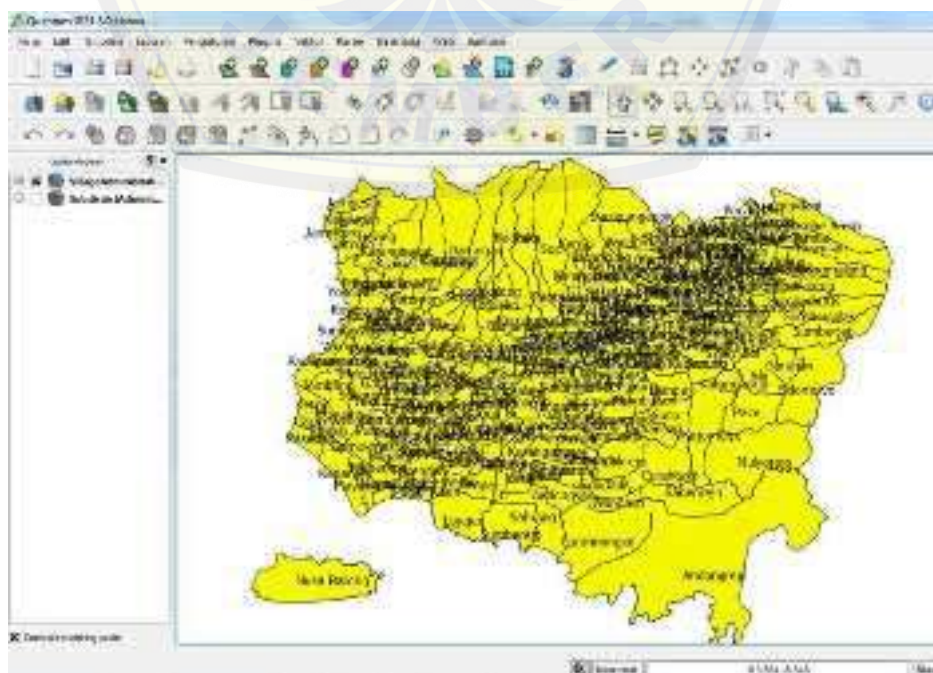
3.5.2.2 Digitasi

Digitasi mengkonversi data spasial pada peta ke dalam format digital. Objek-objek tertentu seperti jalan, rumah, sawah dan lain-lain dapat diubah ke dalam format digital dengan menggunakan citra satelit resolusi tinggi. Langkah awal dalam tahap digitasi ini adalah menambahkan peta Kabupaten Jember.

Pengambilan area kecamatan dan kelurahan menggunakan fungsi *shapefile* dari Quantum GIS, kemudian peneliti membuat area kecamatan dan kelurahan di atas peta Kabupaten Jember yang sudah memiliki koordinat lokasi yang jelas. Hasil dari digitasi kecamatan dan kelurahan yang ada di Kabupaten Jember digambarkan pada Gambar 3.4 dan Gambar 3.5.



Gambar 3.4 Hasil *Digitasi* Kecamatan di Kabupaten Jember



Gambar 3.5 Hasil *Digitasi* Kelurahan di Kabupaten Jember

Data kelurahan dan kecamatan hasil dari digitasi serta data ATM BRI, data ATM bank kompetitor dan data tempat keramaian akan dikonversi ke dalam format data shp menggunakan Quantum GIS tersebut. Data dengan format shp akan dikonversi menggunakan *software PostGIS Shapefile Loader Exporter* sehingga data-data tersebut dapat dimasukkan ke dalam basis data yang digunakan untuk membangun sistem.

3.5.2.3 Data Hasil Digitasi

Data-data dari hasil *digitasi* terdiri dari 2 jenis data yaitu data spasial dan data atribut.. Hasil analisis data melalui proses pengumpulan data, penentuan koordinat lokasi dan proses *digitasi* menghasilkan data *spasial* dan data atribut yang digunakan untuk membangun *Group Decision Support System* (GDSS) Penentuan Lokasi Penempatan Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember.

Data atribut yang diperlukan dalam penelitian ini antara lain:

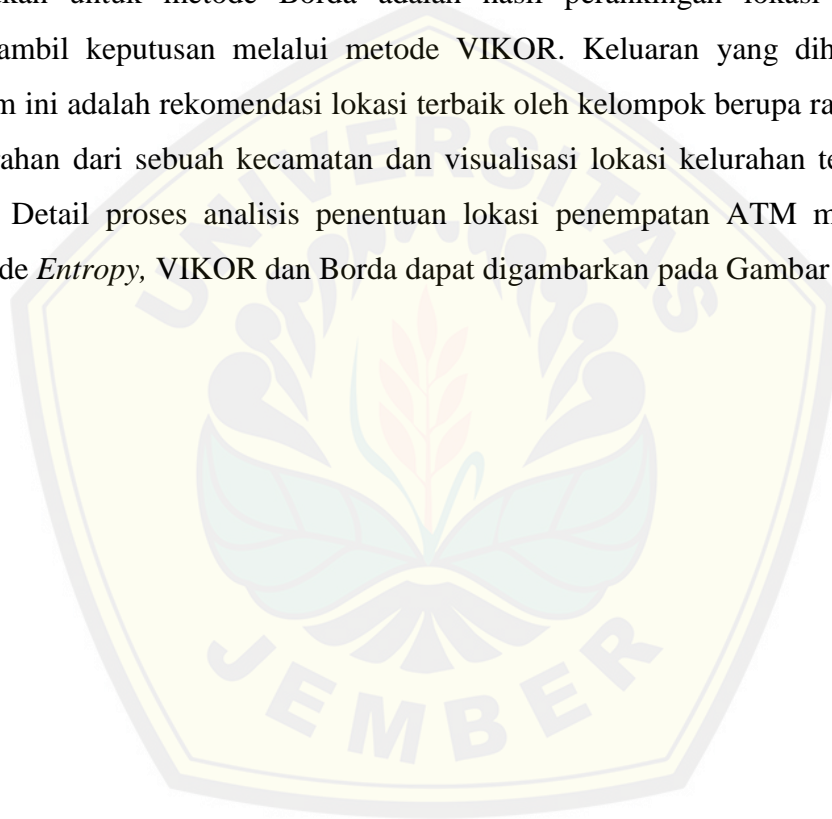
1. Data kecamatan dan kelurahan di Kabupaten Jember
2. Data ATM BRI di setiap kelurahan di Kabupaten Jember
3. Data ATM bank kompetitor (Bank BCA dan Mandiri) di setiap kelurahan di Kabupaten Jember
4. Data tempat keramaian di setiap kelurahan di Kabupaten Jember
5. Data jumlah transaksi ATM BRI setiap kelurahan di Kabupaten Jember
6. Data jumlah nasabah Bank BRI setiap kelurahan di Kabupaten Jember
7. Data jumlah nasabah Bank BRI yang mempunyai kartu ATM setiap kelurahan di Kabupaten Jember
8. Data jumlah instansi yang bekerja sama dengan Bank BRI setiap kelurahan di Kabupaten Jember
9. Data jumlah penduduk setiap kelurahan di Kabupaten Jember

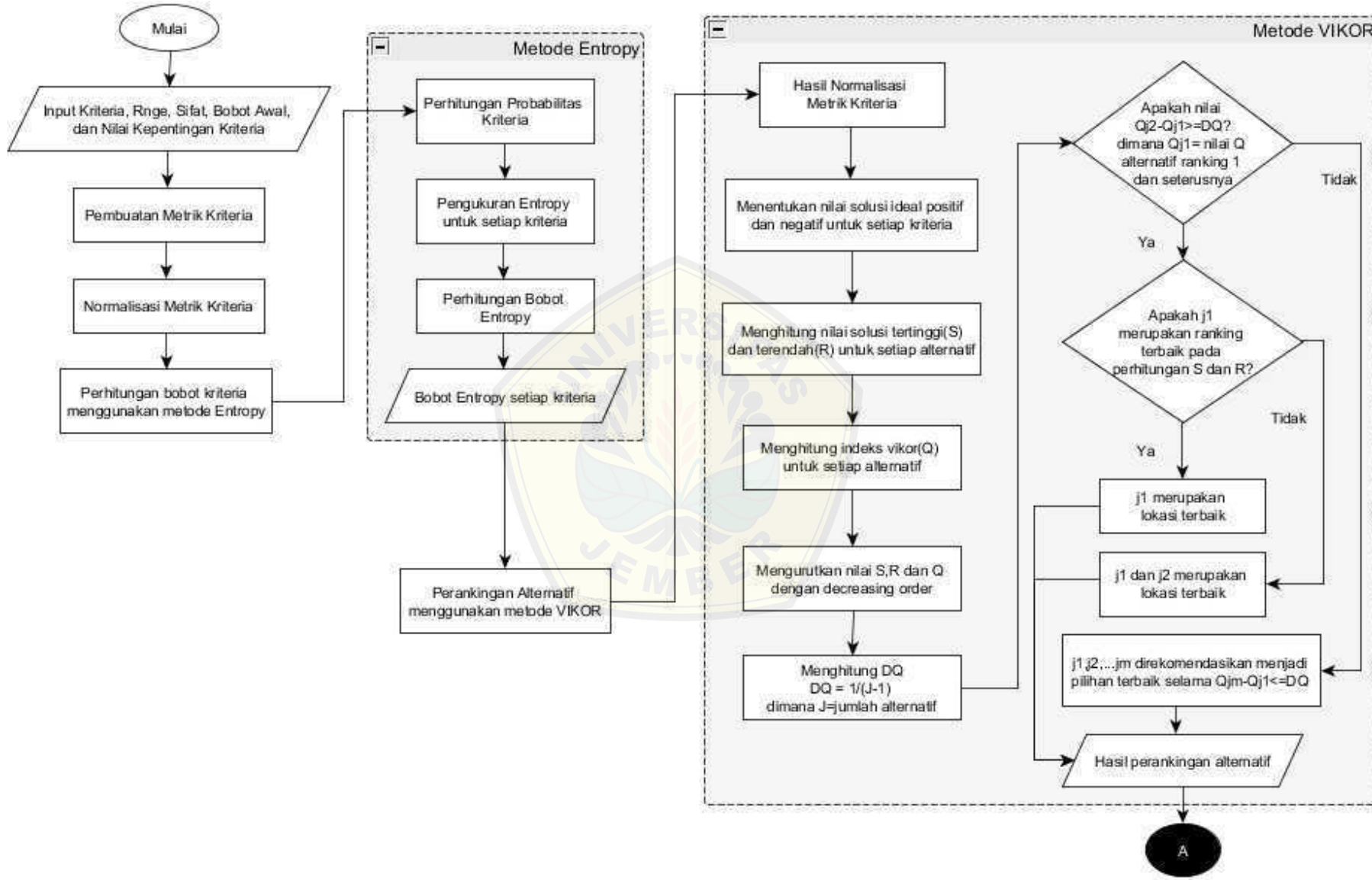
Data spasial yang diperlukan dalam penelitian ini antara lain:

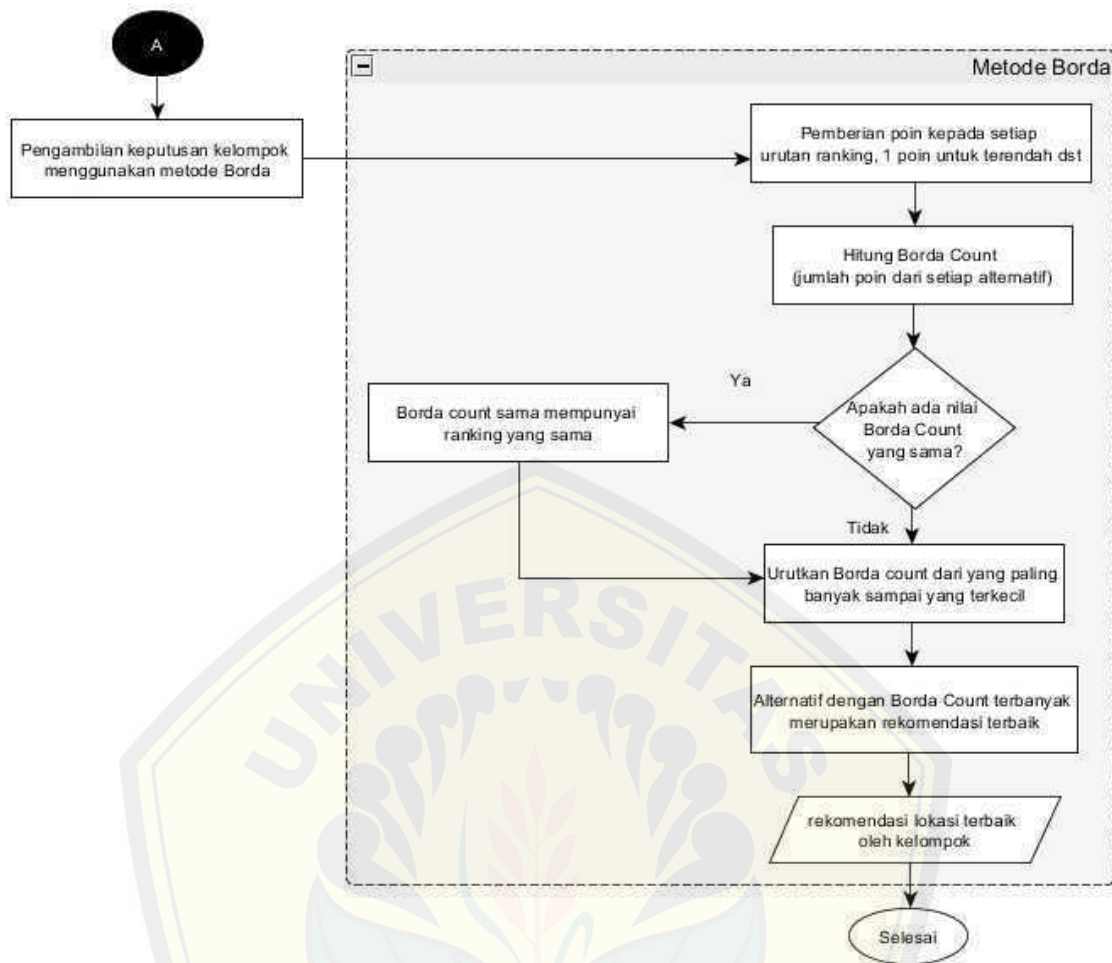
1. Peta Kabupaten Jember
2. Peta lokasi ATM BRI, ATM BCA dan ATM Mandiri di Kabupaten Jember

3. Peta lokasi perumahan, pusat perbelanjaan, rumah sakit dan pantai di Kabupaten Jember

Data spasial dan data atribut yang sudah dikumpulkan akan dianalisis ke dalam metode yang dipakai oleh sistem yaitu metode *Entropy*, VIKOR dan Borda. Data masukan untuk metode *Entropy* adalah data kriteria, nilai kepentingan kriteria dan bobot awal dari pengambil keputusan. Data masukan untuk metode VIKOR adalah bobot kriteria dari hasil perhitungan metode *Entropy*. Sedangkan data masukan untuk metode Borda adalah hasil perankingan lokasi oleh setiap pengambil keputusan melalui metode VIKOR. Keluaran yang dihasilkan dari sistem ini adalah rekomendasi lokasi terbaik oleh kelompok berupa ranking lokasi kelurahan dari sebuah kecamatan dan visualisasi lokasi kelurahan tersebut pada peta. Detail proses analisis penentuan lokasi penempatan ATM menggunakan metode *Entropy*, VIKOR dan Borda dapat digambarkan pada Gambar 3.6.







Gambar 3.6 Diagram Alir GDSS Penentuan Lokasi ATM

3.5.2.4 Analisis Data ke dalam Metode *Entropy*

Data-data masukan yang digunakan pada metode *Entropy* adalah data kriteria, data batasan kriteria dan bobot awal kriteria yang ditentukan oleh para pengambil keputusan. Langkah-langkah perhitungan metode *Entropy* adalah sebagai berikut:

1. Penentuan kriteria dan bobot awal oleh pengambil keputusan

Contoh perhitungan metode *Entropy* ini akan menganalisa data pada Kecamatan Kaliwates. Jumlah dari bobot awal harus sama dengan 1. Kriteria dan bobot awal yang telah ditentukan oleh Pelaksana Bidang IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan Bank untuk Kecamatan Kaliwates dapat dilihat pada Tabel 3.6, Tabel 3.7 dan Tabel 3.8

Tabel 3.6 Kriteria dan Bobot Awal Pelaksana Bidang IT dan E-Channel pada analisa Kecamatan Kaliwates

No.	Kode Kriteria	Kriteria	Sifat Kriteria	Bobot Awal
1.	CA1	Jumlah ATM BRI	Cost	0,4
2.	CA2	Jumlah Transaksi ATM	Benefit	0,3
3.	CA3	Jumlah Tempat Keramaian	Benefit	0,1
4.	CA4	Jumlah Penduduk	Benefit	0,05
5.	CA5	Jumlah Nasabah	Benefit	0,05
6.	CA6	Jumlah ATM Kompetitor	Benefit	0,1

Tabel 3.7 Kriteria dan Bobot Awal Asisten Manajer Pemasaran Dana pada analisa Kecamatan Kaliwates

No.	Kode Kriteria	Kriteria	Sifat Kriteria	Bobot Awal
1.	CB1	Jumlah Transaksi ATM	Benefit	0,7
2.	CB2	Jumlah Nasabah yang Mempunyai Kartu ATM	Benefit	0,2
3.	CB3	Jumlah Tempat Keramaian	Benefit	0,1

Tabel 3.8 Kriteria dan Bobot Awal Pimpinan Bank pada Analisa Kecamatan Kaliwates

No.	Kode Kriteria	Kriteria	Sifat Kriteria	Bobot Awal
1.	CC1	Jumlah Transaksi ATM	Benefit	0,4
2.	CC2	Jumlah Instansi yang Bekerjasama	Benefit	0,4
3.	CC3	Jumlah Tempat Keramaian	Benefit	0,2

2. Pembuatan Metrik Kriteria

Metrik kriteria merupakan metrik nilai kepentingan dari setiap kelurahan terhadap kriteria yang telah ditentukan. Nilai kepentingan didapatkan dengan mencocokkan nilai dengan batasan kriteria yang sudah ditentukan. Setiap batasan mempunyai nilai kepentingan yang berbeda-beda. Nilai kepentingan yang masuk ke batasan kriteria tertentu akan mendapatkan nilai kepentingan sesuai nilai kepentingan milik batasan kriteria tersebut. Batasan kriteria dan nilai kriteria asli

sebelum dimasukkan ke dalam nilai kepentingan dapat dilihat pada Tabel 3.9, Tabel 3.10, Tabel 3.11, Tabel 3.12, Tabel 3.13 dan Tabel 3.14.

Tabel 3.9 Batasan Kriteria oleh Pelaksana IT dan *E-Channel*

No.	Kode Kriteria	Kriteria	Batasan	Nilai Kepentingan
1.	CA1	Jumlah ATM BRI	0 x 1	1
			x=2	2
			x=3	3
			x 4	4
2.	CA2	Jumlah Transaksi ATM	0 x 5999	1
			6000 x 10000	2
			10001 x 15000	3
			x 15001	4
3.	CA3	Jumlah Tempat Keramaian	0 x 1	1
			2 x 3	2
			4 x 5	3
			x 6	4
4.	CA4	Jumlah Penduduk	0 x 5000	1
			5001 x 6000	2
			6001 x 7000	3
			x 7001	4
5.	CA5	Jumlah Nasabah	0 x 20	1
			21 x 30	2
			31 x 40	3
			x 41	4
6.	CA6	Jumlah ATM Kompetitor	0 x 1	1
			2 x 3	2
			4 x 5	3
			x 6	4

Tabel 3.10 Batasan Kriteria oleh Asisten Manajer Pemasaran Dana

No.	Kode Kriteria	Kriteria	Batasan	Nilai Kepentingan
1.	CB1	Jumlah Transaksi ATM	0 x 5999	1
			6000 x 10000	2
			10001 x 15000	3
			x 15001	4
2.	CB2	Jumlah Nasabah yang	0 x 10	1
			11 x 20	2
			21 x 30	3

		Mempunyai Kartu ATM	x	31	4
3.	CB3	Jumlah Tempat Keramaian	0	x 1	1
			2	x 3	2
			4	x 5	3
			x	6	4

Tabel 3.11 Batasan Kriteria oleh Pimpinan Bank

No.	Kode Kriteria	Kriteria	Batasan	Nilai Kepentingan
1.	CC1	Jumlah Transaksi ATM	0 x 5999 6000 x 10000 10001 x 15000 x 15001	1 2 3 4
2.	CC2	Jumlah Instansi yang Bekerjasama	0 x 1 2 x 3 4 x 5 x 6	1 2 3 4
3.	CC3	Jumlah Tempat Keramaian	0 x 1 2 x 3 4 x 5 x 6	1 2 3 4

Tabel 3.12 Metrik Nilai Kriteria Asli Pelaksana IT dan E-Channel

Kelurahan	CA1	CA2	CA3	CA4	CA5	CA6
Jember Kidul	4	5000	1	6000	20	15
Kaliwates	4	6500	3	7000	35	7
Kebon Agung	1	4500	0	5500	24	0
Kepatihan	4	6000	4	6500	40	6
Mangli	2	5000	1	7800	33	1
Sempusari	2	6500	4	5700	43	4
Tegal Besar	3	6000	3	8000	30	4

Tabel 3.13 Metrik Nilai Kriteria Asli Asisten Manajer Pemasaran Dana

Kelurahan	CB1	CB2	CB3
Jember Kidul	5000	15	15
Kaliwates	6500	40	7
Kebon Agung	4500	20	0
Kepatihan	6000	30	5

Mangli	5000	35	1
Sempusari	6500	45	2
Tegal Besar	6000	40	4

Tabel 3.14 Metrik Nilai Kriteria Asli Pimpinan Bank

Kelurahan	CC1	CC2	CC3
Jember Kidul	5000	1	15
Kaliwates	6500	2	7
Kebon Agung	4500	2	0
Kepatihan	6000	0	5
Mangli	5000	1	1
Sempusari	6500	1	2
Tegal Besar	6000	2	4

Langkah selanjutnya yaitu menentukan nilai kepentingan kriteria dilihat dari batasan kriteria yang ada. Pada Tabel 3.12, kriteria CA1 yaitu jumlah ATM BRI pada kelurahan Jember Kidul adalah 4 buah. Jumlah ATM BRI bernilai 4 pada **Tabel 3.9** masuk ke dalam batasan $x \leq 4$ sehingga nilai kepentingan dari nilai tersebut adalah 4. Cara mendapatkan nilai kepentingan untuk data nilai yang lain juga sama seperti itu sehingga hasil dari metrik kriteria oleh para pengambil keputusan dapat dilihat pada Tabel 3.15, Tabel 3.16 dan Tabel 3.17.

Tabel 3.15 Metrik Kriteria Pelaksana IT dan *E-Channel*

Kelurahan	CA1	CA2	CA3	CA4	CA5	CA6
Jember Kidul	4	1	1	2	1	4
Kaliwates	4	2	2	3	3	4
Kebon Agung	1	1	1	2	2	1
Kepatihan	4	2	3	3	3	4
Mangli	2	1	1	4	3	1
Sempusari	2	2	3	2	4	3
Tegal Besar	3	2	2	4	2	3

Tabel 3.16 Metrik Kriteria Asisten Manajer Pemasaran Dana

Kelurahan	CB1	CB2	CB3
Jember Kidul	1	2	1

Kaliwates	2	4	2
Kebon Agung	1	2	1
Kepatihan	2	3	3
Mangli	1	4	1
Sempusari	2	4	3
Tegal Besar	2	4	2

Tabel 3.17 Metrik Kriteria Pimpinan Bank

Kelurahan	CC1	CC2	CC3
Jember Kidul	1	1	1
Kaliwates	2	2	2
Kebon Agung	1	2	1
Kepatihan	2	1	3
Mangli	1	1	1
Sempusari	2	1	3
Tegal Besar	2	2	2

3. Normalisasi Metrik Kriteria

Proses normalisasi mengacu pada sifat kriteria, apakah berupa kriteria *benefit* atau kriteria *cost*. Cara normalisasi kriteria *benefit* dan *cost* menggunakan rumus [3.1] dan [3.2].

$$C_{ij} = \frac{b_{ij}}{b_{ij \text{ maks}}} ; \text{ untuk kriteria } \textit{benefit} \tag{3.1}$$

$$C_{ij} = \frac{b_{ij \text{ min}}}{b_{ij}} ; \text{ untuk kriteria } \textit{cost} \tag{3.2}$$

dimana

C_{ij} : nilai ternormalisasi; b_{ij} : nilai kriteria; $b_{ij \text{ maks}}$: nilai maksimal kriteria; $b_{ij \text{ min}}$: nilai minimal kriteria.

Kriteria yang bersifat *cost* adalah kriteria CA1 yaitu jumlah ATM BRI. Perhitungan normalisasi untuk kriteria jumlah ATM BRI pada kelurahan Jember Kidul menggunakan rumus [3.2] adalah sebagai berikut:

Diketahui : $b_{ij \text{ min}} = 1$ $b_{ij} = 4$

Maka $C_{ij} = \frac{b_{ij \text{ min}}}{b_{ij}} = C_{ij} = \frac{1}{4} = 0,25$

Hasil dari perhitungan normalisasi kriteria jumlah ATM BRI pada kelurahan Jember Kidul yang mempunyai sifat *cost* adalah 0,25. Sedangkan perhitungan untuk kriteria yang mempunyai sifat *benefit* CA2 yaitu jumlah transaksi ATM pada kelurahan Jember Kidul menggunakan rumus [3.1] adalah sebagai berikut:

$$\text{Diketahui : } b_{ij} = 4 \quad b_{ij \text{ maks}} = 4$$

$$\text{Maka } C_{ij} = \frac{b_{ij}}{b_{ij \text{ maks}}} = C_{ij} = \frac{4}{4} = 1$$

Hasil dari perhitungan normalisasi kriteria jumlah transaksi ATM pada kelurahan Jember Kidul yang mempunyai sifat *benefit* adalah 1. Perhitungan pada kriteria lainnya juga sama dengan langkah-langkah di atas sehingga menghasilkan metrik kriteria ternormalisasi sesuai pada Tabel 3.18, Tabel 3.19 dan Tabel 3.20.

Tabel 3.18 Metrik Kriteria Ternormalisasi Pelaksana IT dan *E-Channel*

Kelurahan	CA1	CA2	CA3	CA4	CA5	CA6
Jember Kidul	0,25	0,5	0,33333	0,5	0,25	1
Kaliwates	0,25	1	0,66667	0,75	0,75	1
Kebon Agung	1	0,5	0,33333	0,5	0,5	0,25
Kepatihan	0,25	1	1	0,75	0,75	1
Mangli	0,5	0,5	0,33333	1	0,75	0,25
Sempusari	0,5	1	1	0,5	1	0,75
Tegal Besar	0,33333	1	0,66667	1	0,5	0,75

Tabel 3.19 Metrik Kriteria Ternormalisasi Asisten Manajer Pemasaran Dana

Kelurahan	CB1	CB2	CB3
Jember Kidul	0,5	0,5	0,33333
Kaliwates	1	1	0,66667
Kebon Agung	0,5	0,5	0,33333
Kepatihan	1	0,75	1
Mangli	0,5	1	0,33333
Sempusari	1	1	1
Tegal Besar	1	1	0,66667

Tabel 3.20 Metrik Kriteria Ternormalisasi Pimpinan Bank

Kelurahan	CC1	CC2	CC3
Jember Kidul	0,5	0,5	0,33333
Kaliwates	1	1	0,66667
Kebon Agung	0,5	1	0,33333
Kepatihan	1	0,5	1
Mangli	0,5	0,5	0,33333
Sempusari	1	0,5	1
Tegal Besar	1	1	0,66667

4. Pengukuran *Entropy* untuk Setiap Kriteria

Tahap selanjutnya dari perhitungan metode bobot *entropy* adalah pengukuran *entropy* untuk setiap kriteria. Langkah pertama adalah menghitung probabilitas kriteria. Proses perhitungan probabilitas kriteria merupakan pembagian nilai data yang ternormalisasi dengan jumlah nilai data ternormalisasi menggunakan rumus [3.3].

$$P_{ij} = \frac{c_{ij}}{\sum_{j=1}^m c_{ij}} \tag{3.3}$$

dimana P_{ij} : nilai probabilitas kriteria; c_{ij} : nilai ternormalisasi.

Perhitungan probabilitas untuk kriteria CA1 yaitu jumlah ATM BRI pada kelurahan Jember Kidul menggunakan rumus [3.3] adalah sebagai berikut:

Diketahui :

$$c_{ij} = 0,5$$

$$\sum_{j=1}^7 c_{ij} = 0,25 + 0,25 + 1 + 0,25 + 0,5 + 0,5 + 0,33333 = 3,08333$$

$$\text{Maka } P_{ij} = \frac{c_{ij}}{\sum_{j=1}^7 c_{ij}} = \frac{0,5}{3,08333} = 0,08108117$$

Hasil dari perhitungan probabilitas untuk jumlah ATM BRI pada kelurahan Jember Kidul adalah 0,08108117. Perhitungan probabilitas pada kriteria lainnya juga sama dengan langkah-langkah di atas sehingga menghasilkan metrik probabilitas kriteria yang dapat dilihat pada Tabel 3.21, Tabel 3.22 dan Tabel 3.23.

Tabel 3.21 Metrik Probabilitas Kriteria Pelaksana IT dan E-Channel

Alternatif	CA1	CA2	CA3	CA4	CA5	CA6
Jember Kidul	0,08108117	0,09091	0,07692	0,1	0,05556	0,2
Kaliwates	0,08108117	0,18182	0,15385	0,15	0,16667	0,2
Kebon Agung	0,32432467	0,09091	0,07692	0,1	0,11111	0,05
Kepatihan	0,08108117	0,18182	0,23077	0,15	0,16667	0,2
Mangli	0,16216234	0,09091	0,07692	0,2	0,16667	0,05
Sempusari	0,16216234	0,18182	0,23077	0,1	0,22222	0,15
Tegal Besar	0,10810822	0,18182	0,15385	0,2	0,11111	0,15

Tabel 3.22 Metrik Probabilitas Kriteria Asisten Manajer Pemasaran Dana

Alternatif	CB1	CB2	CB3
Jember Kidul	0,09090909	0,08696	0,07692
Kaliwates	0,18181818	0,17391	0,15385
Kebon Agung	0,09090909	0,08696	0,07692
Kepatihan	0,18181818	0,13043	0,23077
Mangli	0,09090909	0,17391	0,07692
Sempusari	0,18181818	0,17391	0,23077
Tegal Besar	0,18181818	0,17391	0,15385

Tabel 3.23 Metrik Probabilitas Kriteria Pimpinan Bank

Alternatif	CC1	CC2	CC3
Jember Kidul	0,09090909	0,1	0,07692
Kaliwates	0,18181818	0,2	0,15385
Kebon Agung	0,09090909	0,2	0,07692
Kepatihan	0,18181818	0,1	0,23077
Mangli	0,09090909	0,1	0,07692
Sempusari	0,18181818	0,1	0,23077
Tegal Besar	0,18181818	0,2	0,15385

Berdasarkan nilai probabilitas kriteria pada Tabel 3.21, Tabel 3.22 dan Tabel 3.23 maka akan diukur nilai *entropy* terhadap setiap kriteria menggunakan rumus [3.4].

$$E_i = - \frac{1}{\ln m} \sum_{k=1}^n P_{ij} \ln P_{ij}$$

[3.4]

dimana

E_i : nilai *entropy* per kriteria; P_{ij} : nilai probabilitas kriteria; $i: 1,2,\dots,n$ merupakan sejumlah kriteria; $j: 1,2,\dots,m$ merupakan sejumlah alternatif .

Pengukuran *entropy* untuk kriteria CA1 yaitu jumlah ATM BRI pada kelurahan Jember Kidul menggunakan rumus [3.4] adalah sebagai berikut:

Diketahui :

$$m = 7; P_{ij} = 0,08108117$$

$$\begin{aligned} \text{Maka } E_i &= - \frac{1}{\ln m} \sum_{k=1}^n P_{ij} \ln P_{ij} \\ &= - \frac{1}{\ln 7} \sum_{k=1}^6 0,08108117 \ln 0,08108117 \\ &= 0,92851 \end{aligned}$$

Hasil dari pengukuran *entropy* untuk kriteria jumlah ATM BRI yang ditentukan oleh Pelaksana IT dan *E-Channel* adalah 0,92851. Pengukuran *entropy* pada kriteria lainnya juga sama dengan langkah-langkah di atas sehingga menghasilkan nilai *entropy* setiap kriteria yang dapat dilihat pada Tabel 3.24, Tabel 3.25 dan Tabel 3.26

Tabel 3.24 Hasil Nilai *Entropy* Kriteria Pelaksana IT dan *E-Channel*

No.	Kriteria	Nilai <i>Entropy</i>
1.	Jumlah ATM BRI	0,92851
2.	Jumlah Transaksi ATM	0,97321
3.	Jumlah Tempat Keramaian	0,94795
4.	Jumlah Penduduk	0,9783
5.	Jumlah Nasabah	0,9656
6.	Jumlah ATM Kompetitor	0,94268

Tabel 3.25 Hasil Nilai *Entropy* Kriteria Asisten Manajer Pemasaran Dana

No.	Kriteria	Nilai <i>Entropy</i>
1.	Jumlah Transaksi ATM	0,97321

2.	Jumlah Nasabah yang Mempunyai Kartu ATM	0,98014
3.	Jumlah Tempat Keramaian	0,94795

Tabel 3.26 Hasil Nilai *Entropy* Kriteria Pimpinan Bank

No.	Kriteria	Nilai <i>Entropy</i>
1.	Jumlah Transaksi ATM	0,97321
2.	Jumlah Instansi yang Bekerjasama	0,96957
3.	Jumlah Tempat Keramaian	0,94795

5. Perhitungan Bobot *Entropy*

Pengambil keputusan telah memberikan bobot awal terhadap setiap kriteria. Bobot awal yang telah ditentukan setiap kriteria berhubungan secara langsung dengan rata-rata informasi yang dibangkitkan dengan sekumpulan nilai *entropy* setiap kriteria. Perhitungan bobot *entropy* yang sebenarnya menggunakan rumus [3.5] dan [3.6].

$$L_i = \frac{1}{n - \sum_{j=1}^n E_j} (1 - E_i) \tag{3.5}$$

$$WE_i = \frac{L_i * W_i}{\sum_{j=1}^n L_j * W_j} \tag{3.6}$$

dimana

E_i : nilai *entropy* per kriteria; L_i : lamda per kriteria (*range* 0 hingga 1)

W_i : bobot awal oleh pengambil keputusan; WE_i : bobot *entropy* per kriteria

Pengukuran *lamda* untuk kriteria CA1 yaitu kriteria jumlah ATM BRI yang ditentukan oleh Pelaksana IT dan *E-Channel* menggunakan rumus [3.5] adalah sebagai berikut:

Diketahui :

$$E_i = 0,92851 ; n = 6 ;$$

$$\begin{aligned} \sum_{j=1}^6 E_j &= 0,92851 + 0,97321 + 0,94795 + 0,9783 + 0,9656 + 0,94268 \\ &= 5,73625 \end{aligned}$$

$$\begin{aligned} \text{Maka } L_i &= \frac{1}{n - \sum_{j=1}^6 E_j} (1 - E_i) \\ &= \frac{1}{6 - \sum_{j=1}^6 5,73625} (1 - 0,92851) \\ &= 0,271052133 \end{aligned}$$

Hasil dari perhitungan untuk kriteria jumlah ATM BRI yang ditentukan oleh Pelaksana IT dan *E-Channel* adalah 0,271052133. Perhitungan lamda pada kriteria lainnya juga sama dengan langkah-langkah di atas sehingga menghasilkan nilai lamda kriteria yang dapat dilihat pada Tabel 3.27, Tabel 3.28 dan Tabel 3.29.

Tabel 3.27 Hasil Perhitungan Lamda Kriteria Pelaksana IT dan *E-Channel*

No.	Kriteria	Nilai Lamda
1.	Jumlah ATM BRI	0,271052133
2.	Jumlah Transaksi ATM	0,10157346
3.	Jumlah Tempat Keramaian	0,197345972
4.	Jumlah Penduduk	0,082274882
5.	Jumlah Nasabah	0,13042654
6.	Jumlah ATM Kompetitor	0,217327014

Tabel 3.28 Hasil Perhitungan Lamda Kriteria Asisten Manajer Pemasaran Dana

No.	Kriteria	Nilai Lamda
1.	Jumlah Transaksi ATM	0,271428571
2.	Jumlah Nasabah yang Mempunyai Kartu ATM	0,201215805
3.	Jumlah Tempat Keramaian	0,527355623

Tabel 3.29 Hasil Perhitungan Lamda Kriteria Pimpinan Bank

No.	Kriteria	Nilai Lamda
1.	Jumlah Transaksi ATM	0,245172508
2.	Jumlah Instansi yang Bekerjasama	0,278484488
3.	Jumlah Tempat Keramaian	0,476343004

Hasil dari perhitungan lamda setiap kriteria tersebut akan dihitung menggunakan rumus [3.6] untuk mendapatkan bobot *entropy* sebenarnya.

Perhitungan bobot *entropy* untuk kriteria CA1 yaitu kriteria jumlah ATM BRI yang ditentukan oleh Pelaksana IT dan *E-Channel* menggunakan rumus [3.6] adalah sebagai berikut:

Diketahui :

$$L_i = 0,271052133 ; W_i = 0,4;$$

$$\text{Maka } L_i * W_i = 0,271052133 \times 0,4 = 0,10842$$

Langkah-langkah perhitungan dari nilai lamda dikalikan bobot awal untuk kriteria lain yang ditentukan oleh Pelaksana IT dan *E-Channel* sama dengan langkah-langkah perhitungan di atas. Hasil perhitungan lamda dikalikan bobot awal untuk semua kriteria oleh Pelaksana IT dan *E-Channel* dapat dilihat pada Tabel 3.30.

Tabel 3.30 Hasil Perhitungan Lamda Dikalikan Bobot Awal Kriteria Pelaksana IT dan *E-Channel*

Kriteria	Lamda	Bobot Awal	Lamda x Bobot Awal
Jumlah ATM BRI	0,323545822	0,4	0,108421
Jumlah Transaksi ATM	0,093423072	0,3	0,030472
Jumlah Tempat Keramaian	0,181510671	0,1	0,019735
Jumlah Penduduk	0,075673037	0,05	0,004114
Jumlah Nasabah	0,119960943	0,05	0,006521
Jumlah ATM Kompetitor	0,205886456	0,1	0,021733

Langkah perhitungan bobot *entropy* selanjutnya untuk kriteria jumlah ATM BRI yang ditentukan oleh Pelaksana IT dan *E-Channel* menggunakan rumus [3.6] adalah sebagai berikut:

Diketahui :

$$\sum_{j=1}^6 L_i * W_i = 0,10842 + 0,03047 + 0,01973 + 0,00411 + 0,00652 + 0,02173 = 0,191 ;$$

$$L_i * W_i = 0,271052133 \times 0,4 = 0,10842$$

$$\text{Maka } WE_i = \frac{L_i * W_i}{\sum_{j=1}^6 L_i * W_i} = \frac{0,10842}{0,191} = 0,56765$$

Hasil dari pembobotan *entropy* untuk kriteria jumlah ATM BRI yang ditentukan oleh Pelaksana IT dan *E-Channel* adalah 0,628345. Perhitungan bobot *entropy* pada kriteria lainnya juga sama dengan langkah-langkah di atas sehingga menghasilkan bobot *entropy* setiap kriteria yang dapat dilihat pada Tabel 3.31, Tabel 3.32 dan Tabel 3.33

Tabel 3.31 Hasil Pembobotan *Entropy* Kriteria dari Pelaksana IT dan *E-Channel*

No.	Kriteria	Bobot <i>Entropy</i>
1.	Jumlah ATM BRI	0,567648
2.	Jumlah Transaksi ATM	0,159539
3.	Jumlah Tempat Keramaian	0,103322
4.	Jumlah Penduduk	0,021538
5.	Jumlah Nasabah	0,034143
6.	Jumlah ATM Kompetitor	0,113784

Tabel 3.32 Hasil Pembobotan *Entropy* Kriteria dari Asisten Manajer Pemasaran Dana

No.	Kriteria	Bobot <i>Entropy</i>
1.	Jumlah Transaksi ATM	0,671428
2.	Jumlah Nasabah yang Mempunyai Kartu ATM	0,142213
3.	Jumlah Tempat Keramaian	0,186359

Tabel 3.33 Hasil Pembobotan *Entropy* Kriteria dari Pimpinan Bank

No.	Kriteria	Bobot <i>Entropy</i>
1.	Jumlah Transaksi ATM	0,321822
2.	Jumlah Instansi yang Bekerjasama	0,365548
3.	Jumlah Tempat Keramaian	0,312632

Hasil perhitungan bobot *entropy* pada Tabel 3.31, Tabel 3.32 dan Tabel 3.33 merupakan hasil akhir perhitungan metode *Entropy*. Pada dasarnya data yang mempunyai *range* nilai yang besar (relatif terhadap kriteria itu sendiri) dan

mempunyai variasi nilai yang tinggi untuk tiap alternatif, akan memperoleh bobot yang tinggi sehingga kriteria tersebut dianggap mampu untuk membedakan performansi tiap alternatif. Bobot *Entropy* yang dimiliki setiap kriteria akan digunakan pada perhitungan perankingan metode VIKOR.

3.5.2.5 Analisis Data ke dalam Metode VIKOR

Metode VIKOR (*Vlse Kriterijumska Optimizacija Kompromisno Resenje in Serbia*) merupakan metode yang memiliki prosedur perhitungan sederhana dengan pertimbangan kedekatan antar alternatif yang ideal maupun tidak ideal. Data masukan pada metode VIKOR ini adalah metrik kriteria ternormalisasi dan bobot *entropy* yang sudah dihitung pada tahapan perhitungan metode *Entropy* sebelumnya. Langkah-langkah perhitungan metode VIKOR adalah sebagai berikut:

1. Menentukan nilai solusi ideal positif dan negatif untuk setiap kriteria

Tahap pertama dalam perhitungan VIKOR adalah menghitung nilai solusi ideal positif (f_i^*) dan nilai solusi ideal negatif (f_i^-) untuk setiap kriteria berdasarkan hasil normalisasi metrik kriteria pada pada Tabel 3.18, Tabel 3.19 dan Tabel 3.20 menggunakan rumus [3.7] dan [3.8].

$$f_i^* = \max f_{ji} \quad [3.7]$$

$$f_i^- = \min f_{ji} \quad [3.8]$$

dimana $i: 1,2,\dots,n$ merupakan sejumlah kriteria; $j: 1,2,\dots,m$ merupakan sejumlah alternatif.

Perhitungan nilai solusi ideal positif (f_i^*) dan nilai solusi ideal negatif (f_i^-) untuk kriteria CA1 yaitu kriteria jumlah ATM BRI yang ditentukan oleh Pelaksana IT dan *E-Channel* menggunakan rumus [3.7] dan [3.8] adalah sebagai berikut :

$$\begin{aligned}
 f_i^* &= \max f_{ji} \\
 &= \max (0,25;0,25;1;0,25;0,5;0,5;0,33333) \\
 &= 1 \\
 f_i^- &= \min f_{ji} \\
 &= \min (0,25;0,25;1;0,25;0,5;0,5;0,33333) \\
 &= 0,25
 \end{aligned}$$

Hasil dari perhitungan nilai solusi ideal positif (f_i^*) dan nilai solusi ideal negatif (f_i^-) untuk kriteria jumlah ATM BRI yang ditentukan oleh Pelaksana IT dan *E-Channel* adalah 0,90722. perhitungan nilai solusi ideal positif dan nilai solusi ideal negatif pada kriteria lainnya juga sama dengan langkah-langkah di atas sehingga menghasilkan nilai solusi ideal positif dan nilai solusi ideal negatif setiap kriteria yang dapat dilihat pada Tabel 3.34, Tabel 3.35 dan Tabel 3.36.

Tabel 3.34 Nilai Solusi Ideal Positif dan Negatif Kriteria dari Pelaksana IT dan *E-Channel*

No.	Kriteria	f_i^*	f_i^-
1.	Jumlah ATM BRI	1	0,25
2.	Jumlah Transaksi ATM	1	0,5
3.	Jumlah Tempat Keramaian	1	0,33333
4.	Jumlah Penduduk	1	0,5
5.	Jumlah Nasabah	1	0,25
6.	Jumlah ATM Kompetitor	1	0,25

Tabel 3.35 Nilai Solusi Ideal Positif dan Negatif Kriteria dari Asisten Manajer Pemasaran Dana

No.	Kriteria	f_i^*	f_i^-
1.	Jumlah Transaksi ATM	1	0,5
2.	Jumlah Nasabah yang Mempunyai Kartu ATM	1	0,5
3.	Jumlah Tempat Keramaian	1	0,33333

Tabel 3.36 Nilai Solusi Ideal Positif dan Negatif Kriteria dari Pimpinan Bank

No.	Kriteria	f_i^*	f_i^-
1.	Jumlah Transaksi ATM	1	0,5

2.	Jumlah Instansi yang Bekerjasama	1	0,5
3.	Jumlah Tempat Keramaian	1	0,33333

2. Menghitung nilai solusi tertinggi dan terendah untuk setiap alternatif

Bobot *entropy* yang telah dihasilkan pada perhitungan metode *Entropy* akan dimasukkan ke dalam rumus [3.9] dan [3.10] untuk menentukan nilai solusi tertinggi (S_j) dan nilai solusi terendah (R_j) untuk setiap alternatif berdasarkan metrik kriteria ternormalisasi sesuai pada Tabel 3.18, Tabel 3.19 dan Tabel 3.20. Langkah ini digunakan untuk mencari jarak antara masing-masing alternatif dengan solusi ideal positif.

$$S_j = \sum_{i=1}^n WE_i \frac{f_i^* - f_{ji}}{f_i^* - f_i^-} \tag{3.9}$$

$$R_j = \max WE_i \frac{f_i^* - f_{ji}}{f_i^* - f_i^-} \tag{3.10}$$

dimana WE_i merupakan bobot *entropy* setiap kriteria.

Langkah awal dalam perhitungan nilai solusi tertinggi dan terendah adalah menghitung nilai ideal positif dan negatif dikalikan dengan bobot *entropy*. Perhitungan nilai ideal positif dan negatif dikalikan dengan bobot *entropy* untuk kelurahan Jember Kidul dengan kriteria CA1 yaitu jumlah ATM BRI yang ditentukan oleh Pelaksana IT dan *E-Channel* adalah sebagai berikut:

Diketahui:

$$f_i^* = 1 ; f_i^- = 0,25; f_{ji} = 0,25 ; WE_i = 0,567648 ;$$

$$\text{Maka } WE_i \frac{f_i^* - f_{ji}}{f_i^* - f_i^-} = 0,567648 \times \frac{(1 - 0,25)}{(1 - 0,25)} = 0,567648$$

Perhitungan nilai ideal positif dan negatif dikalikan dengan bobot *entropy* untuk nilai kriteria pada kelurahan lain sama dengan langkah-langkah di atas sehingga menghasilkan metrik nilai ideal terbobot yang dapat dilihat pada Tabel 3.37, Tabel 3.38 dan Tabel 3.39.

Tabel 3.37 Metrik Nilai Ideal Terbobot dari Pelaksana IT dan *E-Channel*

Kelurahan	CA1	CA2	CA3	CA4	CA5	CA6
Jember Kidul	0,567648	0,159539	0,103322	0,021538	0,034143	0
Kaliwates	0,567648	0	0,05166	0,010769	0,011381	0
Kebon Agung	0	0,159539	0,103322	0,021538	0,022762	0,113784
Kepatihan	0,567648	0	0	0,010769	0,011381	0
Mangli	0,378432	0,159539	0,103322	0	0,011381	0,113784
Sempusari	0,378432	0	0	0,021538	0	0,037928
Tegal Besar	0,504578523	0	0,05166	0	0,022762	0,037928

Tabel 3.38 Metrik Nilai Ideal Terbobot dari Asisten Manajer Pemasaran Dana

Kelurahan	CB1	CB2	CB3
Jember Kidul	0,671428	0,142213	0,186359
Kaliwates	0	0	0,093178
Kebon Agung	0,671428	0,142213	0,186359
Kepatihan	0	0,0711065	0
Mangli	0,671428	0	0,186359
Sempusari	0	0	0
Tegal Besar	0	0	0,093178

Tabel 3.39 Metrik Nilai Ideal Terbobot dari Pimpinan Bank

Kelurahan	CC1	CC2	CC3
Jember Kidul	0,321822	0,365548	0,312632
Kaliwates	0	0	0,156314
Kebon Agung	0,321822	0	0,312632
Kepatihan	0	0,365548	0
Mangli	0,321822	0,365548	0,312632
Sempusari	0	0,365548	0
Tegal Besar	0	0	0,156314

Setelah menghitung nilai ideal positif dan negatif dikalikan dengan bobot *entropy* maka selanjutnya adalah mencari nilai solusi tertinggi dan terendah untuk setiap kelurahan. Perhitungan mencari nilai solusi tertinggi dan terendah untuk kelurahan Jember Kidul menggunakan rumus [3.9] dan [3.10] adalah sebagai berikut:

$$\begin{aligned}
 S_j &= \sum_{i=1}^6 WE_i \frac{f_i^* - f_{ji}}{f_i^* - f_i^-} \\
 &= 0,567648 + 0,159539 + 0,103322 + 0,021538 + 0,034143 + 0 \\
 &= 0,88619
 \end{aligned}$$

$$\begin{aligned}
 R_j &= \max WE_i \frac{f_i^* - f_{ji}}{f_i^* - f_i^-} \\
 &= \max (0,567648; 0,159539; 0,103322; 0,021538; 0,034143; 0) \\
 &= 0,567648
 \end{aligned}$$

Hasil dari perhitungan nilai solusi tertinggi dan terendah untuk kelurahan Jember Kidul yang ditentukan oleh Pelaksana IT dan *E-Channel* adalah 0,88619 dan 0,567648. Perhitungan nilai solusi tertinggi dan terendah untuk kelurahan lainnya juga sama dengan langkah-langkah di atas sehingga menghasilkan nilai solusi tertinggi dan terendah setiap kelurahan yang dapat dilihat pada Tabel 3.40, Tabel 3.41 dan Tabel 3.42.

Tabel 3.40 Nilai Solusi Tertinggi dan Terendah Kelurahan dari Pelaksana IT dan *E-Channel*

Kelurahan	S_j	R_j
Jember Kidul	0,88619	0,567648
Kaliwates	0,641458	0,567648
Kebon Agung	0,420945	0,159539
Kepatihan	0,589798	0,567648
Mangli	0,766458	0,378432
Sempusari	0,437898	0,378432
Tegal Besar	0,616929	0,504579

Tabel 3.41 Nilai Solusi Tertinggi dan Terendah Kelurahan dari Asisten Manajer Pemasaran Dana

Kelurahan	S_j	R_j
Jember Kidul	1	0,671428
Kaliwates	0,093178	0,093178
Kebon Agung	1	0,671428
Kepatihan	0,071107	0,071107
Mangli	0,857787	0,671428
Sempusari	0	0
Tegal Besar	0,093178	0,093178

Tabel 3.42 Nilai Solusi Tertinggi dan Terendah Kelurahan dari Pimpinan Bank

Kelurahan	S_j	R_j
Jember Kidul	1,000002	0,365548
Kaliwates	0,156314	0,156314
Kebon Agung	0,634454	0,321822
Kepatihan	0,365548	0,365548
Mangli	1,000002	0,365548
Sempusari	0,365548	0,365548
Tegal Besar	0,156314	0,156314

3. Menghitung indeks VIKOR untuk setiap alternatif

Setelah mendapatkan nilai solusi optimal dan solusi terendah, maka tahap selanjutnya adalah menghitung indeks VIKOR. Nilai rasio perankingan (Q_j) merupakan nilai yang digunakan sebagai acuan dalam proses perankingan terhadap tiap-tiap alterntaif yang dapat dihitung menggunakan rumus [3.11] di bawah ini.

$$Q_j = \frac{v (S_j - S^*)}{(S^- - S^*)} + \frac{(1 - v) (R_j - R^*)}{(R^- - R^*)} \tag{3.11}$$

dimana v : merupakan bobot dari strategi utilitas maksimum grup dengan $v=0,5$; S_j : nilai solusi tertinggi alternatif; R_j : nilai solusi terendah alternatif; S^- : nilai maksimal dari S_j ; S^* : nilai minimal dari S_j ; R^* : nilai minimal dari R_j ; R^- : nilai maksimal dari R_j ;

Perhitungan indeks VIKOR untuk kelurahan Jember Kidul yang ditentukan oleh Pelaksana IT dan *E-Channel* adalah sebagai berikut:

Diketahui:

$$S_j = 0,88619 ; S^* = 0,420945 ; S^- = 0,88619 ; R_j = 0,567648 ; R^* = 0,159539 ; R^- = 0,567648$$

Maka

$$Q_j = \frac{v (S_j - S^*)}{(S^- - S^*)} + \frac{(1 - v) (R_j - R^*)}{(R^- - R^*)} = \frac{0,5 (0,88619 - 0,420945)}{(0,88619 - 0,420945)} + \frac{(1 - 0,5) (0,567648 - 0,159539)}{(0,567648 - 0,159539)} = 1$$

Hasil dari perhitungan indeks VIKOR untuk kelurahan Jember Kidul yang ditentukan oleh Pelaksana IT dan *E-Channel* adalah 1. Perhitungan indeks VIKOR untuk kelurahan lainnya juga sama dengan langkah-langkah di atas sehingga menghasilkan indeks VIKOR/nilai rasio perankingan (Q_j) setiap kelurahan yang dapat dilihat pada Tabel 3.43, Tabel 3.44 dan Tabel 3.45.

Tabel 3.43 Nilai Rasio Perankingan (Q) Kelurahan dari Pelaksana IT dan *E-Channel*

Kelurahan	Q_j
Jember Kidul	1
Kaliwates	0,7369833
Kebon Agung	0
Kepatihan	0,68146333
Mangli	0,63950191
Sempusari	0,28639383
Tegal Besar	0,63335075

Tabel 3.44 Nilai Rasio Perankingan (Q) Kelurahan dari Asisten Manajer Pemasaran Dana

Kelurahan	Q_j
Jember Kidul	1
Kaliwates	0,115977
Kebon Agung	1
Kepatihan	0,088505
Mangli	0,928894
Sempusari	0
Tegal Besar	0,115977

Tabel 3.45 Nilai Rasio Perankingan (Q) Kelurahan dari Pimpinan Bank

Kelurahan	Q_j
Jember Kidul	1,000001
Kaliwates	0
Kebon Agung	0,678873
Kepatihan	0,624
Mangli	1,000001
Sempusari	0,624
Tegal Besar	0

4. Mengurutkan nilai Q_j , S_j dan R_j dengan *decreasing order*

Untuk mendapatkan nilai ranking dari setiap alternatif, maka nilai dari Q_j , S_j dan R_j yang paling kecil dari semua alternatif akan menjadi ranking terbaik dan yang paling besar akan mendapatkan ranking yang terakhir. Hasil dari pengurutan nilai Q_j , S_j dan R_j secara *decreasing order* pada penilaian Pelaksana IT dan *E-Channel* dapat dilihat pada Tabel 3.46, Tabel 3.47 dan Tabel 3.48.

Tabel 3.46 Hasil *Decreasing Order* Nilai S dari Pelaksana IT dan *E-Channel*

Kelurahan	S
Kebon Agung	0,420945
Sempusari	0,437898
Kepatihan	0,589798
Tegal Besar	0,616929
Kaliwates	0,641458
Mangli	0,766458
Jember Kidul	0,88619

Tabel 3.47 Hasil *Decreasing Order* Nilai R dari Pelaksana IT dan *E-Channel*

Kelurahan	R
Kebon Agung	0,159539
Mangli	0,378432
Sempusari	0,378432
Tegal Besar	0,504579
Jember Kidul	0,567648
Kaliwates	0,567648
Kepatihan	0,567648

Tabel 3.48 Hasil *Decreasing Order* Nilai Q dari Pelaksana IT dan *E-Channel*

Kelurahan	Q
Kebon Agung	0
Sempusari	0,28639
Tegal Besar	0,63335
Mangli	0,6395
Kepatihan	0,68146
Kaliwates	0,73698
Jember Kidul	1

Penilaian oleh Pelaksana IT dan *E-Channel* pada Tabel 3.46, Tabel 3.47 dan Tabel 3.48 menunjukkan bahwa ranking pertama dari nilai Q_j , S_j dan R_j secara *decreasing order* adalah Kelurahan Kebon Agung. Sedangkan untuk penilaian oleh Asisten Manajer Pemasaran Dana dapat dilihat Tabel 3.49, Tabel 3.50 dan Tabel 3.51.

Tabel 3.49 Hasil *Decreasing Order* Nilai S dari Asisten Manajer Pemasaran Dana

Kelurahan	S
Sempusari	0
Kepatihan	0,071107
Kaliwates	0,093178
Tegal Besar	0,093178
Mangli	0,857787
Jember Kidul	1
Kebon Agung	1

Tabel 3.50 Hasil *Decreasing Order* Nilai R dari Asisten Manajer Pemasaran Dana

Kelurahan	R
Sempusari	0
Kepatihan	0,071107
Kaliwates	0,093178
Tegal Besar	0,093178
Jember Kidul	0,671428
Kebon Agung	0,671428
Mangli	0,671428

Tabel 3.51 Hasil *Decreasing Order* Nilai Q dari Asisten Manajer Pemasaran Dana

Kelurahan	Q
Sempusari	0
Kepatihan	0,08851
Kaliwates	0,11598
Tegal Besar	0,11598
Mangli	0,92889
Jember Kidul	1
Kebon Agung	1

Penilaian oleh Asisten Manajer Pemasaran Dana pada Tabel 3.49 dan Tabel 3.51 menunjukkan bahwa ranking pertama dari nilai Q_j , S_j dan R_j secara *decreasing order* adalah Kelurahan Sempusari. Hasil dari pengurutan nilai Q_j , S_j dan R_j secara

decreasing order dari penilaian Pimpinan Bank dapat dilihat pada Tabel 3.52, Tabel 3.53 dan Tabel 3.54.

Tabel 3.52 Hasil *Decreasing Order* Nilai S dari Pimpinan Bank

Kelurahan	S
Kaliwates	0,156314
Tegal Besar	0,156314
Kepatihan	0,365548
Sempusari	0,365548
Kebon Agung	0,634454
Jember Kidul	1,000002
Mangli	1,000002

Tabel 3.53 Hasil *Decreasing Order* Nilai R dari Pimpinan Bank

Kelurahan	R
Kepatihan	0,156314
Mangli	0,156314
Kaliwates	0,321822
Sempusari	0,365548
Tegal Besar	0,365548
Jember Kidul	0,365548
Kebon Agung	0,365548

Tabel 3.54 Hasil *Decreasing Order* Nilai Q dari Pimpinan Bank

Kelurahan	Q
Kaliwates	0
Tegal Besar	0
Kepatihan	0,624
Sempusari	0,624
Kebon Agung	0,67887
Jember Kidul	1
Mangli	1

Penilaian oleh Pimpinan Bank pada Tabel 3.52, Tabel 3.53 dan Tabel 3.54 menunjukkan bahwa ranking pertama dari nilai Q_j , dan S_j secara *decreasing order* adalah Kelurahan Kaliwates dan Kelurahan Tegal Besar. Sedangkan untuk ranking pertama dari nilai R_j secara *decreasing order* adalah Kelurahan Kepatihan.

5. Memeriksa Kondisi yang Harus Dipenuhi

Solusi terbaik dari alternatif ($j^{(1)}$) merupakan ranking yang terbaik dari nilai minimal Q_j dengan syarat harus memenuhi dua kondisi yaitu :

C1. “Keuntungan yang dapat diterima” :

$Q(j^{(2)}) - Q(j^{(1)}) \geq DQ$, dimana $j^{(2)}$ merupakan alternatif yang berada pada posisi 2 terbaik dari urutan perankingan Q_j , sedangkan $DQ = \frac{1}{(J-1)}$ dan J adalah jumlah dari alternatif yang ada.

C2. “Stabilitas pengambilan keputusan yang dapat diterima”:

Alternatif $j^{(1)}$ harus menjadi ranking yang terbaik pada S_j dan R_j . Hal ini akan membuat stabil proses pembuatan keputusan.

Tahapan selanjutnya dari analisa yang dilakukan para pengambil keputusan terhadap Kecamatan Kaliwates adalah memeriksa kondisi yang harus dipenuhi untuk menentukan kelurahan mana yang akan menjadi ranking terbaik, kedua dan seterusnya. Langkah awal untuk memeriksa kondisi tersebut adalah dengan menghitung nilai DQ dengan $DQ = \frac{1}{(J-1)}$. Perhitungan nilai DQ untuk penilaian Kecamatan Kaliwates dari Pelaksana IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan Bank BRI adalah sebagai berikut:

$$DQ = \frac{1}{(J-1)} = \frac{1}{(7-1)} = \frac{1}{6} = 0,166667$$

Hasil perhitungan DQ untuk penilaian Kecamatan Kaliwates dari Pelaksana IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan Bank BRI adalah 0,166667. Setelah menghitung DQ , maka langkah selanjutnya adalah memeriksa kondisi C1 dan C2 apakah terpenuhi atau tidak.

a. Pemeriksaan Kondisi pada penilaian Pelaksana IT dan *E-Channel*

C1. $Q(j^{(2)}) - Q(j^{(1)}) \geq DQ$

$0,28639 - 0 \geq 0,166667$ (Terpenuhi)

C2. Alternatif $j^{(1)}$ harus menjadi ranking yang terbaik pada S_j dan R_j

Kelurahan Sempusari menjadi ranking yang terbaik pada S_j dan R_j
(Terpenuhi)

Karena kondisi C1 dan C2 terpenuhi, maka ranking kelurahan yang dihasilkan oleh metode VIKOR untuk penilaian Pelaksana IT dan *E-Channel* dapat dilihat pada Tabel 3.55.

Tabel 3.55 Hasil Perankingan VIKOR Penilaian Pelaksana IT dan *E-Channel*

Kelurahan	Q	Ranking
Kebon Agung	0	1
Sempusari	0,28639	2
Tegal Besar	0,63335	3
Mangli	0,6395	4
Kepatihan	0,68146	5
Kaliwates	0,73698	6
Jember Kidul	1	7

b. Pemeriksaan Kondisi pada penilaian Asisten Manajer Pemasaran Dana

C1. $Q(j^{(2)}) - Q(j^{(1)}) \leq DQ$

$$0,08851 - 0 \leq 0,166667$$

$$0,08851 \leq 0,166667 \text{ (Tidak Terpenuhi)}$$

C2. Alternatif $j^{(1)}$ harus menjadi ranking yang terbaik pada S_j dan R_j

Kelurahan Sempusari menjadi ranking yang terbaik pada S_j dan R_j (Terpenuhi)

Karena kondisi C2 terpenuhi sedangkan C1 tidak terpenuhi maka perhitungan VIKOR akan dilanjutkan ke tahap 6.

c. Pemeriksaan Kondisi pada penilaian Pimpinan Bank

C1. $Q(j^{(2)}) - Q(j^{(1)}) \leq DQ$

$$0 - 0 \leq 0,166667$$

$$0 \leq 0,166667 \text{ (Tidak Terpenuhi)}$$

C2. Alternatif $j^{(1)}$ harus menjadi ranking yang terbaik pada S_j dan R_j

Kelurahan Sempusari menjadi ranking yang terbaik pada S_j , tetapi tidak menjadi yang terbaik pada R_j (Tidak Terpenuhi)

Karena kondisi C1 dan C2 tidak terpenuhi maka perhitungan VIKOR akan dilanjutkan ke tahap 6.

6. Jika salah satu kondisi C1 atau C2 tidak terpenuhi, maka tetap ada rekomendasi pilihan terbaik yaitu:

- (i) Alternatif $j^{(1)}$ dan $j^{(2)}$, jika hanya kondisi C2 yang tidak terpenuhi;
- (ii) Alternatif $j^{(1)}, j^{(2)}, \dots, j^{(M)}$ jika kondisi C1 tidak terpenuhi; dan $j^{(M)}$ ditentukan menggunakan persamaan $Q(j^{(M)}) - Q(j^{(1)}) < DQ$ untuk *maximum* M (posisi dari alternatif tersebut berdekatan).

Penentuan ranking kelurahan yang cocok untuk dibangun ATM baru pada Kecamatan Kaliwates berdasarkan penilaian Asisten Manajer Pemasaran Dana dan Pimpinan Bank sesuai pemeriksaan kondisi pada perhitungan VIKOR tahap ke 5 adalah sebagai berikut:

a. Penentuan Ranking Kelurahan berdasarkan Penilaian Asisten Manajer Pemasaran Dana

Pada perhitungan sebelumnya, kondisi C1 yaitu $Q(j^{(2)}) - Q(j^{(1)}) < DQ$ tidak terpenuhi sehingga rekomendasi lokasi terbaik adalah *point* ke (ii) pada tahap ke 6 yaitu alternatif $j^{(1)}, j^{(2)}, \dots, j^{(M)}$ jika kondisi C1 tidak terpenuhi; dan $j^{(M)}$ ditentukan menggunakan persamaan $Q(j^{(M)}) - Q(j^{(1)}) < DQ$ untuk *maximum* M (posisi dari alternatif tersebut berdekatan). Perhitungan $Q(j^{(M)}) - Q(j^{(1)}) < DQ$ untuk menentukan ranking kelurahan berdasarkan penilaian Asisten Manajer Pemasaran Dana dapat dilihat pada Tabel 3.56.

Tabel 3.56 Perhitungan Kondisi Kedua Perankingan Kelurahan Berdasarkan Penilaian Asisten Manajer

$Q(j^{(1)})$	$Q(j^{(M)})$	$Q(j^{(M)}) - Q(j^{(1)}) < DQ$	Ranking
0	$Q(j^{(3)}) = 0,11598$	$0,11598 < 0,166667$ (Terpenuhi)	1
0	$Q(j^{(4)}) = 0,11598$	$0,11598 < 0,166667$ (Terpenuhi)	1
0	$Q(j^{(5)}) = 0,92889$	$0,92889 < 0,166667$ (Tidak Terpenuhi)	5
0	$Q(j^{(6)}) = 1$	$1 < 0,166667$ (Tidak Terpenuhi)	6
0	$Q(j^{(7)}) = 1$	$1 < 0,166667$ (Tidak Terpenuhi)	6

Setelah pemeriksaan kondisi terpenuhi dan perhitungan DQ sudah dilakukan, maka selanjutnya sudah diketahui hasil dari perankingan VIKOR berdasarkan

penilaian Asisten Manajer Pemasaran Dana. Hasil perankingan VIKOR tersebut dapat dilihat pada Tabel 3.57.

Tabel 3.57 Hasil Perankingan VIKOR Penilaian Asisten Manajer Pemasaran Dana

Kelurahan	Q	Ranking
Sempusari	0	1
Kepatihan	0,08851	1
Kaliwates	0,11598	1
Tegal Besar	0,11598	1
Mangli	0,92889	5
Jember Kidul	1	6
Kebon Agung	1	6

b. Penentuan Ranking Kelurahan berdasarkan Penilaian Pimpinan Bank

Pada perhitungan sebelumnya, kondisi C1 dan C2 tidak terpenuhi sehingga rekomendasi lokasi terbaik adalah *point* ke (ii) pada tahap ke 6 yaitu alternatif $j^{(1)}$, $j^{(2)}$,, $j^{(M)}$ jika kondisi C1 tidak terpenuhi; dan $j^{(M)}$ ditentukan menggunakan persamaan $Q(j^{(M)}) - Q(j^{(1)}) < DQ$ untuk *maximum* M (posisi dari alternatif tersebut berdekatan). Perhitungan $Q(j^{(M)}) - Q(j^{(1)}) < DQ$ untuk menentukan ranking kelurahan berdasarkan penilaian Pimpinan Bank dapat dilihat pada Tabel 3.58.

Tabel 3.58 Perhitungan Kondisi Kedua Perankingan Kelurahan Berdasarkan Penilaian Pimpinan Bank

$Q(j^{(1)})$	$Q(j^{(M)})$	$Q(j^{(M)}) - Q(j^{(1)}) < DQ$	Ranking
0	$Q(j^{(3)}) = 0,624$	$0,624 < 0,166667$ (Tidak Terpenuhi)	3
0	$Q(j^{(4)}) = 0,624$	$0,624 < 0,166667$ (Tidak Terpenuhi)	3
0	$Q(j^{(5)}) = 0,67887$	$0,67887 < 0,166667$ (Tidak Terpenuhi)	5
0	$Q(j^{(6)}) = 1$	$1 < 0,166667$ (Tidak Terpenuhi)	6
0	$Q(j^{(7)}) = 1$	$1 < 0,166667$ (Tidak Terpenuhi)	6

Setelah pemeriksaan kondisi terpenuhi dan perhitungan DQ sudah dilakukan, maka selanjutnya sudah diketahui hasil dari perankingan VIKOR berdasarkan

penilaian Pimpinan Bank. Hasil perankingan VIKOR tersebut dapat dilihat pada Tabel 3.59.

Tabel 3.59 Hasil Perankingan VIKOR Penilaian Pimpinan Bank

Kelurahan	Q	Ranking
Kaliwates	0	1
Tegal Besar	0	1
Kepatihan	0,624	3
Sempusari	0,624	3
Kebon Agung	0,67887	5
Jember Kidul	1	6
Mangli	1	6

3.5.2.6 Analisis Data ke dalam Metode Borda

Metode Borda merupakan metode voting untuk menentukan keputusan kelompok. Metode Borda menentukan pemenang dari suatu pemilihan dengan memberikan suatu jumlah *point* tertentu untuk masing-masing kandidat sesuai dengan ranking yang telah diatur oleh masing-masing pengambil keputusan. Pemenang akan ditentukan oleh banyaknya jumlah *point* yang dikumpulkan atau diperoleh dari masing-masing kandidat. Langkah-langkah perhitungan metode Borda untuk analisa Kecamatan Kaliwates adalah sebagai berikut:

1. Mengumpulkan Hasil Perankingan Masing-Masing Pengambil Keputusan

Hasil perankingan lokasi oleh Pelaksana Bidang IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan Bank terhadap Kecamatan Kaliwates dapat dilihat pada Tabel 3.60.

Tabel 3.60 Hasil Perankingan Lokasi Masing-Masing Pengambil Keputusan

Kelurahan	Ranking DM1	Ranking DM2	Ranking DM3
Jember Kidul	7	6	6
Kaliwates	6	1	1
Kebon Agung	1	6	5
Kepatihan	5	1	3
Mangli	4	5	6
Sempusari	2	1	3
Tegal Besar	3	1	1

Keterangan

DM1 : Pelaksana Bidang IT dan *E-Channel* ;

DM2 : Asisten Manajer Pemasaran Dana ;

DM3 : Pimpinan Bank BRI.

2. Memberikan *Point Borda* Kepada Lokasi Sesuai Ranking yang Diperoleh

Langkah perhitungan metode Borda selanjutnya adalah memberikan *point Borda* kepada setiap lokasi sesuai dengan ranking yang ditentukan oleh pengambil keputusan. Proses pemberian *point Borda* adalah jika ada n kandidat akan menerima n *point* untuk pilihan pertama, $n-1$ untuk pilihan kedua, $n-2$ untuk pilihan 3 dan $n-4$ untuk pilihan 5. Kelurahan yang terdapat di Kecamatan Kaliwates berjumlah 7 buah sehingga ranking 1 mendapatkan *point* 7, ranking 2 mendapatkan *point* 6 dan seterusnya sampai ranking 7 mendapatkan *point* 1. Hasil dari pemberian *point Borda* kepada setiap kelurahan berdasarkan penilaian masing-masing pengambil keputusan dapat dilihat pada Tabel 3.61.

Tabel 3.61 *Point Borda* Setiap Kelurahan oleh Setiap Pengambil Keputusan

Kelurahan	<i>Point Borda</i> DM1	<i>Point Borda</i> DM2	<i>Point Borda</i> DM3
Jember Kidul	1	2	2
Kaliwates	2	7	7
Kebon Agung	7	2	3
Kepatihan	3	7	5
Mangli	4	3	2
Sempusari	6	7	5
Tegal Besar	5	7	7

3. Menghitung *Borda Count*

Perhitungan metode Borda selanjutnya adalah menghitung *Borda Count* dengan cara menjumlahkan setiap *point Borda* yang dimiliki oleh masing-masing kelurahan. Kelurahan yang mendapatkan *Borda Count* terbanyak mendapatkan ranking terbaik dan sebaliknya kelurahan dengan *Borda Count* paling sedikit akan

menjadi ranking rekomendasi yang terakhir. Hasil dari perhitungan Borda *Count* dapat dilihat pada Tabel 3.62.

Tabel 3.62 Hasil Perhitungan Metode Borda

Kelurahan	Ranking DM1	Ranking DM2	Ranking DM3	Borda Count	Ranking
Jember Kidul	1	2	2	5	7
Kaliwates	2	7	7	16	3
Kebon Agung	7	2	3	12	5
Kepatihan	3	7	5	15	4
Mangli	4	3	2	9	6
Sempusari	6	7	5	18	2
Tegal Besar	5	7	7	19	1

Hasil perhitungan metode Borda merupakan tahap terakhir dari analisis data ke dalam metode sistem. Hasil akhir dari analisa Kecamatan Kaliwates menggunakan kriteria yang sudah ditentukan oleh para pengambil keputusan menghasilkan suatu keputusan kelompok dengan menggunakan metode *Entropy*, VIKOR dan Borda yang dapat dilihat pada Tabel 3.63.

Tabel 3.63 Hasil Perankingan Lokasi Penempatan ATM pada Kecamatan Kaliwates

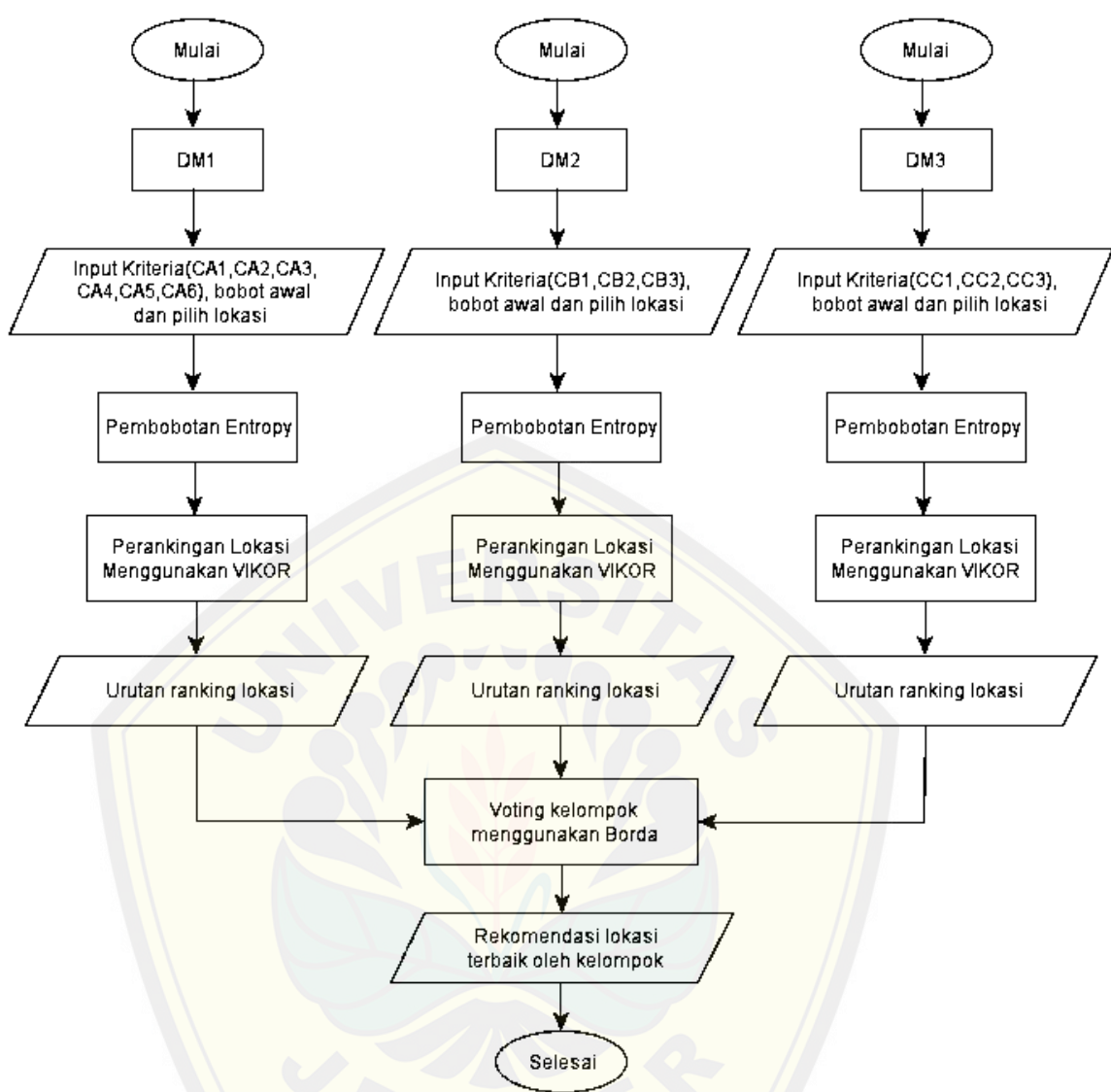
Kelurahan	Borda Count	Ranking
Tegal Besar	19	1
Sempusari	18	2
Kaliwates	16	2
Kepatihan	15	4
Kebon Agung	12	5
Mangli	9	6
Jember Kidul	5	7

3.5.2.7 Gambaran Umum Sistem

Sistem yang akan dibangun adalah *Group Decision Support System (GDSS)* penentuan lokasi penempatan Anjungan Tunai Mandiri menggunakan metode *Entropy*, VIKOR dan Borda yang akan digunakan untuk memudahkan pengambilan keputusan kelompok pada PT. BRI (Persero) Tbk Cabang Jember. Aktor yang

terlibat dalam sistem ini adalah admin, pengambil keputusan dan nasabah. Semua aktor dihubungkan oleh suatu koneksi jaringan internet, di mana semua datanya disimpan pada *database server* pusat *database* lokasi.

Tahap pertama dari penentuan lokasi penempatan ATM adalah setiap pengambil keputusan memasukkan nilai kriteria dan bobot sesuai dengan penilaian masing-masing ke dalam sistem. Sistem akan mengolah data dan bobot kriteria tersebut menggunakan metode pembobotan *Entropy*. Hasil dari perhitungan bobot kriteria menggunakan metode *Entropy* akan dimasukkan ke dalam perhitungan perankingan VIKOR. Perhitungan VIKOR menghasilkan urutan ranking alternatif berupa lokasi penempatan ATM yang sesuai dengan kriteria. Untuk mendapatkan hasil rekomendasi alternatif terbaik dari sebuah kelompok pengambil keputusan maka hasil VIKOR dari masing-masing pengambil keputusan akan dimasukkan ke dalam *group* menggunakan metode *voting* Borda. Perhitungan Borda digunakan untuk menggabungkan penilaian para pengambil keputusan, sehingga diperoleh hasil akhir berupa urutan perankingan alternatif oleh kelompok. Alur dari penentuan lokasi penempatan ATM ini dijelaskan pada Gambar 3.7.

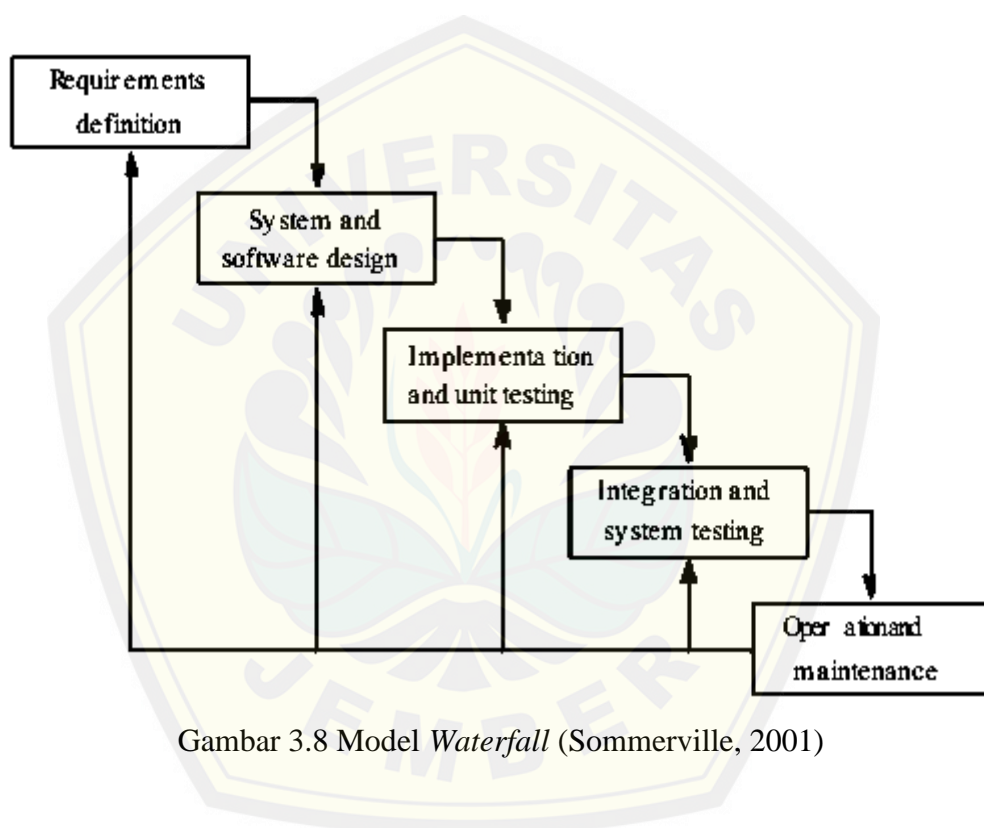


Ket:
 DM1=Pelaksana IT & E-Channel
 DM2=As. Manajer Pemasaran Dana
 DM3=Pimpinan Bank
 CA1=Jumlah ATM BRI
 CA2=Jumlah Transaksi ATM
 CA3=Jumlah Tempat Keramaian
 CA4=Jumlah Penduduk
 CA5=Jumlah Nasabah
 CA6=Jumlah ATM Kompetitor
 CB1=Jumlah Transaksi ATM
 CB2=Jumlah Nasabah Mempunyai Kartu ATM
 CB3=Jumlah Tempat Keramaian
 CC1=Jumlah Transaksi ATM
 CC2=Jumlah Instansi yang Bekerjasama
 CC3=Jumlah Tempat Keramaian

Gambar 3.7 Alur Pengambilan Keputusan pada Sistem

3.5.3 Tahap Pengembangan Sistem

Pengembangan sistem dilakukan setelah analisis data telah selesai dilakukan dan digunakan untuk membangun sistem. Pengembangan GDSS Penentuan Lokasi ATM ini akan menggunakan model *waterfall*. Model *waterfall* merupakan model yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, *test* dan pemeliharaan (Roger S. Pressman, 2002). Diagram alir dari model *Waterfall* digambarkan pada Gambar 3.8.



Gambar 3.8 Model *Waterfall* (Sommerville, 2001)

3.5.3.1. Analisis Kebutuhan

Tahap pertama pada proses perancangan perangkat lunak ini adalah analisis kebutuhan. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional sistem.

3.5.3.2. Desain Sistem

Pembuatan desain sistem menggunakan *Unified Modeling Language (UML)* yang dirancang menggunakan konsep *Object-Oriented Programming (OOP)*. Berikut Pemodelan UML yang digunakan antara lain :

1. *Business Process*

Business Proses digunakan untuk menggambarkan masukan data yang dibutuhkan sistem, keluaran dari sistem serta tujuan dari pembuatan sistem.

2. *Usecase Diagram*

Usecase adalah dokumentasi untuk menggambarkan fitur dan aktor yang terdapat pada sistem yang akan dibuat

3. *Usecase Scenario*

Usecase Scnario digunakan untuk menjelaskan atau menceritakan fitur atau isi yang ada di *usecase diagram*. *Usecase scenario* menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.

4. *Sequence Diagram*

Sequence diagram adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagiannya berupa pesan atau *message*.

5. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

6. *Class Diagram*

Class Diagram adalah sebuah spesifikasi yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

7. *Entity Relationship Diagram*

Entity Relationship Diagram merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

3.5.3.3.Implementasi

Pada tahap ini desain yang telah dibuat akan diimplementasikan ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

1. Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework Code Igniter (CI)*.
2. Manajemen basisdata menggunakan *DBMS PostgreSQL*.

3.5.3.4.Pengujian

Pada tahap ini dilakukan uji coba sistem yang telah dibuat dengan pengujian *black box* dan *white box*.

1. *Black Box Testing* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari sistem yang berkaitan dengan struktur internal atau kerja. Pada pengujian *black box* ini, sistem yang dibangun akan diuji dengan mengujikan langsung *running* sistem dan melakukan kegiatan pengujian dengan menganalisis proses masukan dan keluaran yang dihasilkan oleh sistem. Dalam pengujian *black box* juga dilakukan pengujian dengan cara memasukkan data normal dan data salah, dari masukan data inilah yang nantinya akan dilakukan analisis terhadap reaksi yang muncul pada sistem.
2. *White box testing* adalah cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Tahapan teknik pengujian jalur dasar meliputi:
 - a) *Listing Program*

Listing program merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program. Contoh penerapan tahapan ini dapat dilihat pada Gambar 3.9.

```

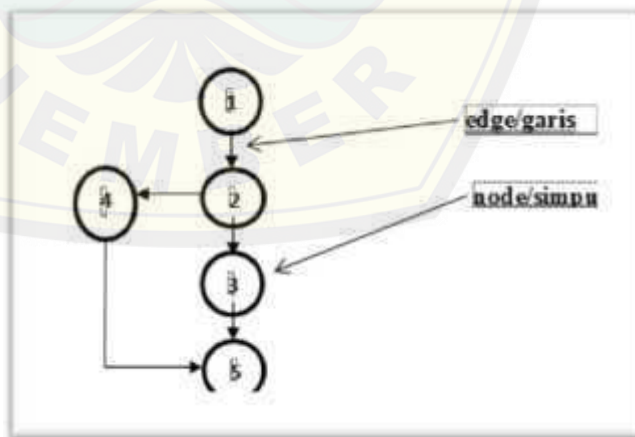
Spanjang = $_POST['p'];
$lebar = $_POST['l'];
if($spanjang == $lebar)
{
    $jenisBangun = 'Persegi';
}
else
{
    $jenisBangun = 'Persegi Panjang';
}

$luas = $spanjang * $lebar;
echo 'Luas bangun '.$jenisBangun.' adalah '.$luas;
    
```

Gambar 3.9 Contoh Listing Program
(Sumber : Roger S. Pressman, 2002)

b) Grafik Alir

Grafik alir merupakan notasi sederhana yang digunakan untuk merepresentasikan aliran kontrol. Aliran kkontrol yang digambarkan merupakan hasil penomoran dari *listing* program. Grafik alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program. Contoh penggambaran diagram alir dapat dilihat pada Gambar 3.10.



Gambar 3.10 Contoh Grafik Alir
(Sumber : Roger S. Pressman, 2002)

c) Kompleksitas Siklomatik

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program. Perhitungan kompleksitas siklomatik menggunakan rumus [3.12].

$$V(G) = E - N + 2$$

[3.12]

Keterangan :

 $V(G)$: Kompleksitas siklomatik E : Jumlah *edge* N : Jumlah *node*

Berdasarkan grafik alir yang ada pada tahapan kedua diketahui jumlah *edge* adalah 5 dan jumlah *node* adalah 5, sehingga dapat dihitung kompleksitas siklomatiknya $V(G) = E - N + 2 = 5 - 5 + 2 = 2$. Jadi jumlah jalur independen adalah 2 jalur.

d) Jalur Program Independen

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi. Dari perhitungan kompleksitas siklomatik, basis set yang dihasilkan dari jalur independen secara linier adalah 2 jalur, yaitu:

Jalur 1 : 1-2-3-5

Jalur 2 : 1-2-4-5

e) Pengujian Basis Set

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke

dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali. Dari tahap sebelumnya telah diketahui 2 basis set. Jika kemudian diuji dengan memasukkan data panjang = 5 dan lebar = 3, maka basis set jalur yang digunakan adalah 1-2-4-5. Dapat dilihat bahwa jalur telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan perangkat lunak, sistem ini telah memenuhi syarat.

3.5.3.5. Pemeliharaan

Perangkat lunak yang sudah selesai pasti akan mengalami perubahan. Perubahan tersebut biasanya berupa kesalahan atau *error* sehingga diperlukan perbaikan dan pemeliharaan kepada sistem. Tahap pemeliharaan ini akan dilakukan apabila sistem yang telah selesai diuji digunakan oleh PT. BRI (Persero) Tbk Cabang Jember. Pemeliharaan dilakukan sesuai dengan kebutuhan pengguna seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

BAB 4. PENGEMBANGAN SISTEM

Bab ini akan membahas tentang pengembangan *Group Decision Support System* (GDSS) Penentuan Lokasi Penempatan Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember. Tahap pengembangan sistem dilaksanakan berdasarkan model *waterfall*, dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program dan pengujian sistem.

4.1 *Statement Of Purpose*

Group Decision Support System (GDSS) penentuan lokasi penempatan ATM merupakan sistem pendukung keputusan kelompok yang digunakan untuk memberikan rekomendasi lokasi penempatan ATM sesuai dengan kriteria yang diberikan oleh para pengambil keputusan pada PT.BRI (Persero) Tbk Cabang Jember. Tujuan dikembangkan sistem ini adalah untuk membantu para pengambil keputusan menentukan lokasi penempatan ATM yang baru dan menggabungkan hasil keputusan dari beberapa pengambil keputusan menjadi satu keputusan akhir. Sistem ini dapat mengelola data pengambil keputusan, data lokasi, kriteria, batasan kriteria, nilai kriteria, informasi apa saja yang akan ditampilkan ke peta dan mengelola analisa pengambilan keputusan yang dilakukan. Sistem ini juga bisa melakukan perhitungan analisa pengambilan keputusan mulai dari pembobotan kriteria, perankingan lokasi penempatan yang terpilih dan menggabungkan keputusan menjadi satu keputusan kelompok. Sistem ini menampilkan persebaran ATM BRI di Kabupaten Jember dan memberikan informasi tentang kriteria lainnya seperti persebaran ATM Mandiri, ATM BCA dan tempat keramaian. Sistem juga dapat digunakan oleh masyarakat umum untuk mencari letak, alamat dan melihat persebaran ATM BRI.

4.2 Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem merupakan tahapan yang sangat penting dalam pengembangan sebuah sistem informasi. Seluruh kebutuhan penggunaan didefinisikan dan diformulasikan pada tahap kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan hal pokok yang harus dapat dilakukan oleh sistem dalam menerima masukan untuk diproses sehingga menghasilkan keluaran. Kebutuhan fungsional berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan fungsional dari *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM antara lain:

1. Sistem dapat menyimpan dan mengelola (*create,update,delete*) data pendukung pengambil keputusan seperti data *user* dan data lokasi kecamatan serta data kelurahan di Kabupaten Jember .
2. Sistem dapat menyimpan dan mengelola data kriteria, batasan kriteria dan nilai kriteria yang digunakan untuk proses pengambilan keputusan.
3. Sistem dapat menyimpan dan mengelola data kategori sebagai informasi yang akan ditampilkan ke dalam peta.
4. Sistem dapat menampilkan peta digital persebaran ATM BRI dan data kategori beserta informasinya.
5. Sistem dapat mengelola dan mengatur proses pengambilan keputusan yang akan dilakukan.
6. Sistem dapat menganalisis kelurahan mana yang cocok dari sebuah kecamatan yang dipilih untuk ditempatkan sebuah ATM baru berdasarkan data kriteria dan penilaian dari para pengambil keputusan.
7. Sistem dapat menampilkan hasil analisis pengambilan keputusan dari masing-masing pengambil keputusan maupun kelompok dalam bentuk grafik dan peta digital.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan hal yang dibutuhkan oleh sistem untuk mendukung aktivitas sistem sesuai dengan kebutuhan fungsional yang telah disusun. Kebutuhan nonfungsional menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Kebutuhan non-fungsional dari *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM antara lain:

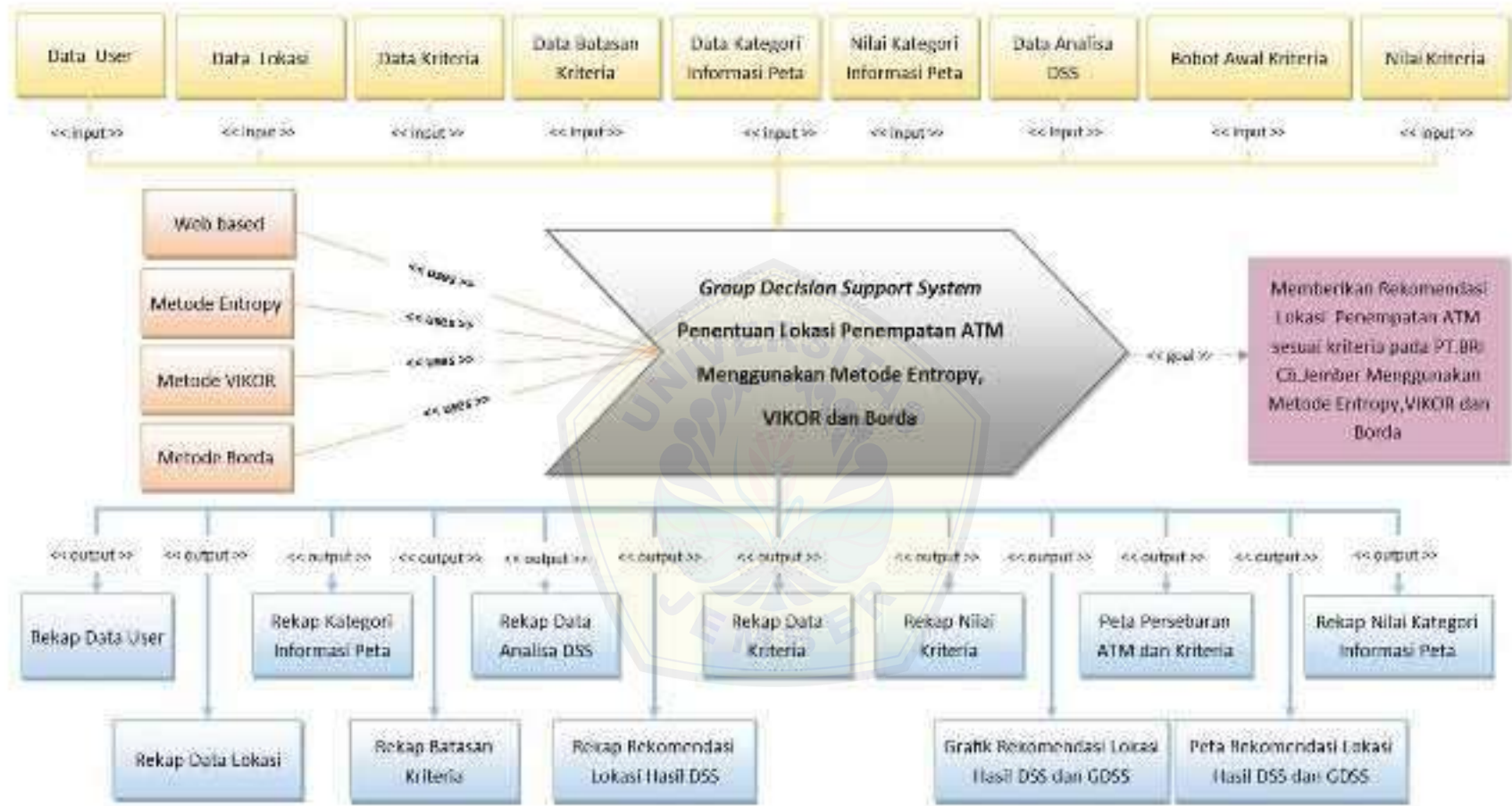
1. Sistem dapat digunakan oleh banyak pengguna pada saat yang bersamaan dan pada komputer yang berbeda.
2. Tampilan dan bahasa komunikasi sistem mudah dimengerti oleh pengguna untuk memberikan kenyamanan pemakaian dan memudahkan pengoperasian.
3. Sistem menggunakan *username* dan *password* untuk autentifikasi akses pengguna terhadap sistem.
4. Sistem mudah dioperasikan pada beberapa sistem operasi.

4.3 Desain Sistem

Desain sistem adalah tahapan untuk memodelkan GDSS Penentuan Lokasi Penempatan ATM. Dokumen desain sistem yang akan dibuat meliputi *Business Process*, *Usecase Diagram*, *Usecase Skenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* dan *Entity Relationship Diagram* (ERD).

4.3.1 *Business Process*

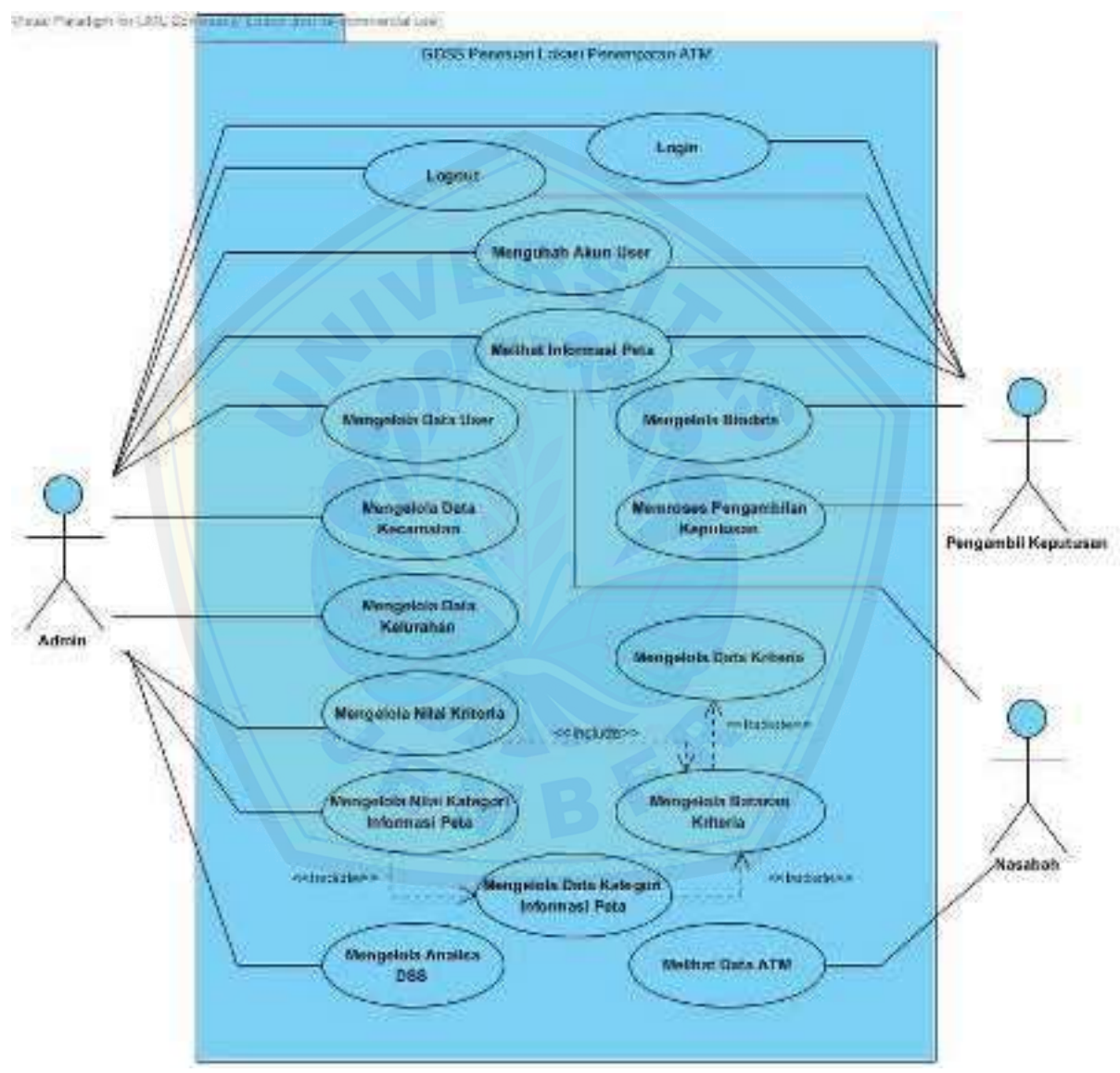
Business Process merupakan sekumpulan proses yang dilakukan untuk mencapai hasil yang diinginkan. Terdapat beberapa komponen untuk mencapai tujuan tersebut, yaitu data yang menjadi masukan (*input*) dan media yang digunakan sehingga akan dihasilkan data keluraan (*output*) untuk melakukan tujuan (*goal*) yang ingin dicapai. *Business Process Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM dapat dilihat pada Gambar 4.1.



Gambar 4.1 Business Process GDSS Penentuan Lokasi Penempatan ATM

4.3.2 Usecase Diagram

Usecase Diagram adalah dokumentasi untuk menggambarkan fitur dan aktor yang terdapat pada sistem yang akan dibuat. Usecase diagram Group Decision Support System (GDSS) penentuan lokasi penempatan ATM dapat dilihat pada Gambar 4.2.



Gambar 4.2 Usecase Diagram GDSS Penentuan Lokasi Penempatan ATM

Penjelasan tentang definisi aktor dan definisi usecase dalam Usecase diagram GDSS penentuan lokasi penempatan ATM dijelaskan pada sub-subbab di bawah ini.

1. Definisi Aktor

Definisi aktor merupakan penjelasan tentang aktor – aktor sebagai pengguna dari *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM yang akan dibangun. Terdapat 3 (tiga) aktor dari hasil analisis seperti yang dijelaskan pada Tabel 4.1.

Tabel 4.1 Definisi Aktor

No	Aktor	Deskripsi
1.	Admin	Aktor admin merupakan aktor yang memiliki hak akses untuk mengatur proses GDSS secara penuh. Aktor ini dapat melakukan <i>login</i> dan <i>logout</i> , mengelola data <i>user</i> , mengubah akun <i>user</i> , mengelola data kecamatan, mengelola data kelurahan, mengelola data dan nilai kategori informasi yang akan ditampilkan ke dalam peta, mengelola kriteria, mengelola batasan kriteria, mengelola nilai kriteria, dan mengelola proses analisa pengambilan keputusan.
2.	Pengambil Keputusan	Aktor pengambil keputusan merupakan aktor yang memiliki hak akses untuk memasukkan kriteria dan penilaiannya ke dalam sistem sehingga sistem menghasilkan rekomendasi lokasi sesuai kriteria yang diberikan. Aktor ini dapat melakukan <i>login</i> dan <i>logout</i> , melihat informasi peta, melihat dan mengubah biodata, mengubah akun <i>user</i> , memproses pengambilan keputusan dan melihat hasil dari analisa pengambilan keputusan individu dan kelompok.
3.	Nasabah	Aktor nasabah merupakan aktor yang dapat melihat informasi peta, dan melihat data ATM yang dikelola oleh admin. Aktor nasabah ini merupakan masyarakat umum baik itu nasabah dari Bank BRI maupun bukan.

2. Definisi *Usecase*

Definisi *Usecase* merupakan penjelasan dari masing – masing *Usecase* atau fitur – fitur pada *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM. Penjelasan *usecase* dapat di lihat pada Tabel 4.2.

Tabel 4.2 Definisi *Usecase*

No.	<i>Usecase</i>	Penjelasan
UC-01	<i>Login</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>login</i> atau autentifikasi untuk masuk sistem.
UC-02	<i>Logout</i>	Merupakan <i>usecase</i> yang menggambarkan proses keluar dari sistem.
UC-03	Melihat Informasi Peta	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan informasi dari peta yaitu persebaran ATM BRI dan penggambaran kriteria pengambilan keputusan.
UC-04	Mengelola Data <i>User</i>	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah dan menghapus data <i>user</i> .
UC-05	Mengelola Data Kecamatan	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah dan menghapus data kecamatan di Kabupaten Jember.
UC-06	Mengelola Data Kelurahan	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah dan menghapus data kelurahan di Kabupaten Jember.
UC-07	Mengelola Data Kriteria	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah dan menghapus data kriteria penentuan lokasi penempatan ATM.
UC-08	Mengelola Batasan Kriteria	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah dan mengubah batasan kriteria berdasarkan data kriteria yang sudah dimasukkan.
UC-09	Mengelola Nilai Kriteria	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah dan menghapus nilai kriteria

		berdasarkan data kriteria dan batasan kriteria yang sudah dimasukkan.
UC-10	Mengelola Data Kategori Informasi Peta	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah dan menghapus kategori informasi yang akan ditampilkan pada peta berdasarkan data kriteria dan batasan kriteria yang sudah dimasukkan.
UC-11	Mengelola Nilai Kategori Informasi Peta	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah dan menghapus nilai kategori informasi peta berdasarkan data kategori informasi peta yang telah dimasukkan.
UC-12	Mengelola Analisa DSS	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah dan menghapus data analisa pengambilan keputusan yang dilakukan.
UC-13	Mengelola Biodata	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan dan mengubah biodata pengambil keputusan yang sedang mengakses sistem.
UC-14	Memroses Pengambilan Keputusan	Merupakan <i>usecase</i> yang menggambarkan proses melihat data pengambilan keputusan, menambah kriteria baru dan melakukan proses pengambilan keputusan menggunakan metode <i>Entropy</i> , <i>VIKOR</i> dan <i>Borda</i> .
UC-15	Melihat Data ATM	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data ATM BRI yang ada di Kabupaten Jember.
UC-16	Mengubah Akun <i>User</i>	Merupakan <i>usecase</i> yang menggambarkan proses mengubah data akun <i>user</i> .

4.3.3 *Usecase* Skenario

Usecase skenario adalah dokumentasi terhadap kebutuhan fungsional sistem. *Usecase* skenario *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM adalah sebagai berikut:

1. *Usecase* Skenario Mengelola Analisa DSS

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola analisa DSS dapat dilihat pada Tabel 4.3.

Tabel 4.3 *Usecase* Skenario Mengelola Analisa DSS

Nomor <i>Usecase</i>	UC-12
Nama	Mengelola Analisa DSS
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu Analisa DSS
<i>Post Condition</i>	Admin berhasil melihat, menambah, mengubah dan menghapus data analisa DSS
SKENARIO NORMAL MELIHAT ANALISA DSS	
Aktor	Sistem
1. Memilih menu Analisa DSS	2. Mengambil data analisa DSS dari <i>database</i>
	3. Menampilkan data analisa DSS pada tabel di halaman data analisa DSS
SKENARIO NORMAL MELIHAT DETAIL ANALISA DSS	
Aktor	Sistem
4. Memilih tombol “Detail Analisa” pada salah satu baris data analisa DSS	5. Mengambil detail analisa DSS dari <i>database</i> berdasarkan analisa yang dipilih
	6. Menampilkan detail analisa DSS pada tabel di halaman detail analisa DSS
7. Klik tombol “Back”	8. Mengambil data analisa DSS dari <i>database</i>
	9. Menampilkan data analisa DSS pada tabel di halaman data analisa DSS
SKENARIO NORMAL MELIHAT HASIL GDSS	
Aktor	Sistem
4. Memilih tombol “Hasil GDSS” pada salah satu baris data analisa dss.	5. Mengambil hasil GDSS dari <i>database</i> berdasarkan analisa yang

- dipilih
6. Menampilkan hasil analisa *decision maker*, hasil analisa kelompok, grafik hasil GDSS, peta hasil GDSS dan rekomendasi lokasi penempatan ATM terbaik
 7. Melihat hasil GDSS
 8. Klik tombol “Back”
 9. Mengambil data analisa DSS dari *database*
 10. Menampilkan data analisa DSS pada tabel di halaman data analisa DSS

SKENARIO NORMAL MENGUBAH STATUS ANALISA DSS

- | Aktor | Sistem |
|--|--|
| <ol style="list-style-type: none"> 4. Menekan tombol “Aktif/Tidak Aktif” pada kolom status di baris data yang akan diubah statusnya | <ol style="list-style-type: none"> 5. Mengubah status analisa DSS dan menyimpan perubahannya ke <i>database</i> 6. Menampilkan <i>alert</i> “Selamat! Analisa Diaktifkan/Tidak Diaktifkan” 7. Mengambil data analisa dss dari <i>database</i> 8. Menampilkan data analisa pada tabel di halaman data analisa dss |

SKENARIO NORMAL MENAMBAH ANALISA DSS

- | Aktor | Sistem |
|---|---|
| <ol style="list-style-type: none"> 4. Menekan tombol “Tambah Analisa” 6. Mengisi seluruh isian <i>form</i> tambah analisa 7. Menekan tombol “Save” | <ol style="list-style-type: none"> 5. Menampilkan <i>form</i> isian tambah analisa DSS 8. Menyimpan masukan data ke <i>database</i> 9. Menampilkan <i>alert</i> “Selamat! Analisa DSS sukses disimpan” 10. Mengambil data analisa DSS dari <i>database</i> 11. Menampilkan data analisa DSS pada tabel di halaman data analisa DSS |

SKENARIO NORMAL TAMBAH DATA KEMBALI KE HALAMAN

ANALISA DSS

Aktor	Sistem
6. Klik tombol "Back"	7. Mengambil data analisa dss dari <i>database</i>
	8. Menampilkan data analisa dss pada tabel di halaman data analisa dss

SKENARIO NORMAL MENGOSONGKAN ISIAN FORM TAMBAH ANALISA DSS

Aktor	Sistem
7. Menekan tombol "Clear"	8. Menampilkan <i>form</i> isian tambah analisa DSS yang sudah dikosongkan

SKENARIO NORMAL MENGUBAH ANALISA DSS

Aktor	Sistem
4. Menekan tombol "Ubah Analisa" pada baris data yang akan diubah	5. Mengambil analisa DSS yang dipilih dari <i>database</i>
	6. Menampilkan analisa DSS ke dalam <i>form</i> ubah analisa
7. Mengubah isian <i>form</i> ubah analisa	
8. Menekan tombol "Save Edit"	9. Menyimpan data yang diubah ke <i>database</i>
	10. Menampilkan <i>alert</i> "Selamat! Data Analisa DSS sukses diubah"
	11. Mengambil data analisa dss dari <i>database</i>
	12. Menampilkan data analisa pada tabel di halaman data analisa dss

SKENARIO NORMAL UBAH DATA KEMBALI KE HALAMAN ANALISA DSS

Aktor	Sistem
7. Klik tombol "Back"	8. Mengambil data analisa dss dari <i>database</i>
	9. Menampilkan data analisa dss pada tabel di halaman data analisa dss

SKENARIO NORMAL MENGEMBALIKAN ANALISA DSS SEPERTI SEMULA

Aktor	Sistem
8. Menekan tombol "Clear"	

9. Menampilkan *form* isian ubah analisa DSS yang belum diubah

SKENARIO NORMAL MENGHAPUS ANALISA DSS

Aktor	Sistem
4. Menekan tombol “Hapus Analisa” pada baris data yang akan dihapus	5. Menampilkan <i>alert</i> “Apakah Anda yakin akan menghapus data ini?” disertai dengan tombol OK dan <i>Cancel</i>
6. Menekan tombol “OK” pada <i>alert</i>	7. Menghapus analisa DSS yang dipilih dari <i>database</i>
	8. Menampilkan <i>alert</i> “Selamat! Analisa DSS sukses dihapus”
	9. Mengambil analisa DSS dari <i>database</i>
	10. Menampilkan data analisa pada tabel di halaman data analisa DSS

SKENARIO NORMAL BATAL MENGHAPUS ANALISA DSS

Aktor	Sistem
6. Menekan tombol “Cancel” pada <i>alert</i>	7. Menampilkan analisa DSS pada halaman data analisa DSS

SKENARIO ALTERNATIF MENAMBAH ANALISA DSS

Jika aktor tidak mengisi *form* tambah analisa secara lengkap

Aktor	Sistem
6. Mengisi isian <i>form</i> tambah analisa tidak lengkap	8. Memeriksa masukan tambah analisa
7. Menekan tombol “Save”	9. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> isian analisa DSS. Tulisan akan muncul dibawah masing-masing <i>field</i> inputan

SKENARIO ALTERNATIF MENGUBAH ANALISA DSS

Jika aktor tidak mengisi *form* ubah data analisa DSS secara lengkap

Aktor	Sistem
7. Mengubah isian <i>form</i> ubah data	

- analisa DSS secara tidak lengkap
8. Menekan tombol "Save Edit"
 9. Memeriksa ubahan analisa DSS
 10. Menampilkan *message* dengan menunjuk isian mana yang tidak diisi pada *form* ubah analisa DSS. Tulisan akan muncul dibawah masing-masing *field* inputan

2. *Usecase* Skenario Memroses Pengambilan Keputusan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario memroses pengambilan keputusan dapat dilihat pada Tabel 4.4.

Tabel 4.4 *Usecase* Skenario Memroses Pengambilan Keputusan

Nomor <i>Usecase</i>	UC-14
Nama	Memroses Pengambilan Keputusan
Aktor	Pengambil Keputusan
<i>Pre Condition</i>	Pengambil Keputusan harus memilih menu Pengambilan Keputusan
<i>Post Condition</i>	Pengambil Keputusan berhasil melihat data dan hasil pengambilan keputusan dalam bentuk grafik dan peta, menambah kriteria baru dan melihat hasil rekomendasi lokasi penempatan ATM berdasarkan perhitungan sistem menggunakan metode <i>Entropy</i> , VIKOR dan Borda

SKENARIO NORMAL MELIHAT DATA PENGAMBILAN KEPUTUSAN

Aktor	Sistem
1. Memilih menu Pengambilan Keputusan	
	2. Mengambil data pengambilan keputusan yang hanya dilakukan oleh Pengambil Keputusan dari <i>database</i>
	3. Menampilkan data pengambilan keputusan pada tabel di halaman data pengambilan keputusan

SKENARIO NORMAL MELIHAT HASIL DSS

Aktor	Sistem
4. Memilih tombol “Hasil DSS” pada salah satu baris data pengambilan keputusan	5. Mengambil hasil pengambilan keputusan Pengambil Keputusan dari <i>database</i> berdasarkan analisa yang dipilih 6. Menampilkan hasil analisa dari <i>decision maker</i> , peta hasil analisa dss dan rekomendasi lokasi penempatan ATM terbaik
7. Klik tombol “Back”	8. Mengambil data pengambilan keputusan yang hanya dilakukan oleh Pengambil Keputusan dari <i>database</i> 9. Menampilkan data pengambilan keputusan pada tabel di halaman data pengambilan keputusan


SKENARIO NORMAL MELIHAT HASIL GDSS

Aktor	Sistem
4. Memilih tombol “Hasil GDSS” pada salah satu baris data pengambilan keputusan	5. Mengambil hasil GDSS dari <i>database</i> berdasarkan data pengambilan keputusan yang dipilih 6. Menampilkan hasil analisa para <i>decision maker</i> , hasil analisa GDSS, grafik hasil GDSS, peta hasil GDSS dan rekomendasi lokasi penempatan ATM terbaik
7. Klik tombol “Back”	8. Mengambil data pengambilan keputusan yang hanya dilakukan oleh DM dari <i>database</i> 9. Menampilkan data pengambilan keputusan pada tabel di halaman data pengambilan keputusan

SKENARIO NORMAL MEMROSES PENGAMBILAN KEPUTUSAN

Aktor	Sistem
4. Menekan tombol "Proses DSS"	5. Mengambil data kriteria dari <i>database</i>
7. Pilih kriteria yang akan dipakai dalam pengambilan keputusan	6. Menampilkan data kriteria, <i>field</i> masukan bobot awal kriteria pada tabel di halaman pilih kriteria dan tombol "Tambah Kriteria"
9. Mengisi bobot awal kriteria	8. Mengaktifkan <i>field</i> masukan bobot awal kriteria yang dipilih agar bisa diisi oleh <i>decision maker</i>
10. Menekan tombol "Pembobotan <i>Entropy</i> "	11. Menyimpan bobot awal kriteria pada <i>database</i>
	12. Menentukan metrik kriteria (kelurahan,kriteria) yang berisi nilai kriteria
	13. Menghitung normalisasi data pada metrik kriteria
	14. Melakukan perhitungan probabilitas kriteria
	15. Melakukan perhitungan natural logaritma probabilitas kriteria
	16. Menghitung bobot <i>entropy</i> untuk setiap kriteria
	17. Menghitung bobot kriteria dengan menggabungkan nilai <i>entropy</i> dengan bobot awal yang sudah disimpan di <i>database</i>
	18. Menyimpan bobot akhir kriteria ke <i>database</i>
	19. Menampilkan metrik kriteria, metrik kriteria ternormalisasi, metrik probabilitas kriteria, metrik natural logaritma probabilitas kriteria, hasil

- pengukuran entropy setiap kriteria, hasil perhitungan bobot entropy setiap kriteria dan hasil pembobotan entropy pada halaman pembobotan entropy.
20. Menekan tombol “Perankingan VIKOR”
 21. Menghitung nilai solusi ideal positif dan negatif setiap kriteria
 22. Menghitung solusi ideal terbobot setiap kriteria
 23. Menghitung nilai solusi tertinggi(S) dan terendah(R) setiap kelurahan
 24. Menghitung nilai rasio VIKOR(Q)
 25. Mengurutkan nilai S,R,Q dengan *decreasing order*
 26. Memeriksa kondisi penentuan ranking kelurahan
 27. Menentukan ranking kelurahan sebagai hasil perankingan VIKOR
 28. Menyimpan hasil perankingan VIKOR ke *database*
 29. Menampilkan metrik kriteria ternormalisasi, metrik solusi ideal positif negatif setiap kriteria, metrik solusi ideal terbobot ,metrik nilai S,R dan Q, hasil *decreasing order* dari nilai S,R dan Q, serta hasil perankingan VIKOR pada halaman perankingan VIKOR
 30. Menekan tombol “Analisa Kelompok”
 31. Mengambil data hasil perhitungan VIKOR dari *database*
 32. Menentukan metrik hasil perankingan VIKOR dari para pengambil keputusan
 33. Memberikan *point* kepada setiap urutan ranking, 1 *point* untuk terendah dan seterusnya

- 
34. Menghitung Borda *count* untuk setiap kelurahan
 35. Memberikan ranking kepada setiap kelurahan dengan Borda *count* yang terbanyak merupakan ranking terbaik
 36. Menyimpan hasil perhitungan Borda ke *database*
 37. Menampilkan metrik ranking kelurahan dari para pengambil keputusan, metrik *point* Borda, hasil Borda *count* setiap kelurahan, dan hasil rekomendasi lokasi penempatan ATM pada halaman analisa kelompok
 38. Menekan tombol “Lihat Hasil Analisa”
 39. Mengambil data pengambilan keputusan yang hanya dilakukan oleh Pengambil Keputusan dari *database*
 40. Menampilkan data pengambilan keputusan pada tabel di halaman data pengambilan keputusan

SKENARIO NORMAL MENAMBAH KRITERIA BARU

- | Aktor | Sistem |
|--|---|
| 7. Menekan tombol “Tambah Kriteria” | |
| 9. Mengisi seluruh isian <i>form</i> tambah kriteria | 8. Menampilkan <i>form</i> tambah kriteria |
| 10. Menekan tombol “Save” | 11. Menyimpan masukan data kriteria ke <i>database</i> |
| | 12. Menampilkan <i>form</i> tambah batasan kriteria |
| 13. Mengisi seluruh isian <i>form</i> batasan kriteria | |
| 14. Menekan tombol “Save All” | 15. Menyimpan masukan batasan kriteria ke <i>database</i> |
| | 16. Mengambil data batasan kriteria dari |

database

17. Menampilkan batasan kriteria dari kriteria yang dimasukkan ke dalam tabel di halaman batasan kriteria

SKENARIO NORMAL MENGOSONGKAN ISIAN *FORM* TAMBAH KRITERIA

Aktor	Sistem
10. Menekan tombol " <i>Clear</i> "	11. Menampilkan <i>form</i> tambah kriteria yang sudah dikosongkan inputannya

SKENARIO NORMAL MENGOSONGKAN ISIAN *FORM* TAMBAH BATASAN KRITERIA

Aktor	Sistem
14. Menekan tombol " <i>Clear</i> "	17. Menampilkan <i>form</i> tambah batasan kriteria yang sudah dikosongkan

SKENARIO NORMAL MENGUBAH BATASAN KRITERIA

Aktor	Sistem
18. Menekan tombol "Ubah Batasan" pada salah satu baris data batasan kriteria	19. Mengambil data batasan kriteria yang dipilih dari <i>database</i>
	20. Menampilkan data batasan kriteria pada <i>form</i> ubah batasan kriteria
21. Mengisi semua masukan <i>form</i> ubah batasan kriteria	
22. Menekan tombol " <i>Save Edit</i> "	23. Mengubah batasan kriteria pada <i>database</i>
	24. Menampilkan <i>alert</i> "Selamat! Batasan kriteria sukses diubah"
	25. Mengambil data batasan kriteria dari <i>database</i>
	26. Menampilkan batasan kriteria dari kriteria yang dimasukkan ke dalam tabel di halaman batasan kriteria

SKENARIO NORMAL MENGEMBALIKAN BATASAN KRITERIA SEPERTI SEMULA

Aktor	Sistem
-------	--------

22. Menekan tombol “*Clear*”

23. Menampilkan *form* ubah batasan kriteria yang belum diubah datanya

SKENARIO NORMAL MENGEMBALIKAN BATASAN KRITERIA SEPERTI SEMULA

Aktor

Sistem

21. Klik tombol “*Back*”

22. Mengambil data batasan kriteria dari *database*

23. Menampilkan batasan kriteria dari kriteria yang dimasukkan ke dalam tabel di halaman batasan kriteria

SKENARIO NORMAL MENAMBAH NILAI KRITERIA

18. Menekan tombol “**Tambah Nilai Kriteria**”

19. Mengambil data kelurahan pada *database*

20. Menampilkan data kelurahan beserta masukan nilai pada tabel di halaman tambah nilai kriteria

21. Mengisi masukan semua nilai kriteria

22. Menekan tombol “*Save All*”

23. Menyimpan nilai kriteria ke *database*

24. Menampilkan *alert* “Selamat! Kriteria sukses disimpan”

25. Mengambil data pengambilan keputusan yang hanya dilakukan oleh DM dari *database*

26. Menampilkan data pengambilan keputusan pada tabel di halaman data pengambilan keputusan

SKENARIO NORMAL MENGOSONGKAN ISIAN *FORM* TAMBAH NILAI KRITERIA

Aktor

Sistem

22. Menekan tombol “*Clear*”

23. Menampilkan halaman tambah nilai kriteria yang sudah dikosongkan nilainya

SKENARIO ALTERNATIF MENAMBAH DATA KRITERIA

Jika aktor tidak mengisi *form* tambah kriteria secara lengkap

Aktor	Sistem
9. Mengisi isian <i>form</i> tambah kriteria secara tidak lengkap	
10. Menekan tombol " <i>Save</i> "	11. Memeriksa masukan kriteria
	12. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> isian tambah kriteria. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan

Jika aktor tidak mengisi *form* tambah batasan kriteria secara lengkap

Aktor	Sistem
13. Mengisi isian <i>form</i> tambah batasan kriteria secara tidak lengkap	
14. Menekan tombol " <i>Save</i> "	15. Memeriksa masukan batasan kriteria
	16. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> isian tambah batasan kriteria. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan

Jika aktor mengisi masukan angka dengan huruf pada *form* tambah batasan kriteria

Aktor	Sistem
13. Mengisi masukan angka pada <i>form</i> tambah batasan kriteria dengan huruf	
14. Menekan tombol " <i>Save</i> "	15. Memeriksa masukan batasan kriteria
	16. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan yang akan muncul adalah "input harus berupa angka" pada <i>form</i> isian tambah batasan kriteria

SKENARIO ALTERNATIF MENGUBAH BATASAN KRITERIA**Jika aktor tidak mengisi *form* mengubah batasan kriteria secara lengkap**

Aktor	Sistem
21. Mengisi isian <i>form</i> ubah batasan kriteria secara tidak lengkap	
22. Menekan tombol “ <i>Save Edit</i> ”	23. Memeriksa ubahan batasan kriteria
	24. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> ubah batasan kriteria. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan

Jika aktor mengisi masukan angka dengan huruf pada *form* ubah batasan kriteria

Aktor	Sistem
21. Mengisi masukan angka pada <i>form</i> ubah batasan kriteria dengan huruf	
22. Menekan tombol “ <i>Save Edit</i> ”	23. Memeriksa ubahan batasan kriteria
	24. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan yang akan muncul adalah “input harus berupa angka” pada <i>form</i> ubah batasan kriteria

SKENARIO ALTERNATIF MENAMBAH NILAI KRITERIA

Jika aktor tidak mengisi masukan pada *form* tambah nilai kriteria secara lengkap

Aktor	Sistem
18. Mengisi isian <i>form</i> tambah nilai kriteria secara tidak lengkap	
19. Menekan tombol “ <i>Save All</i> ”	20. Memeriksa masukan nilai kriteria
	21. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> isian tambah nilai kriteria. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan

Jika aktor mengisi masukan angka dengan huruf pada *form* tambah nilai kriteria

Aktor	Sistem
18. Mengisi masukan angka pada	

- halaman tambah nilai kriteria dengan huruf
19. Menekan tombol “*Save All*”
 20. Memeriksa masukan nilai kriteria
 21. Menampilkan *message* dengan menunjuk isian mana yang salah. Tulisan yang akan muncul adalah “input harus berupa angka” pada *form* tambah nilai kriteria

SKENARIO ALTERNATIF MEMROSES PENGAMBILAN KEPUTUSAN

Jika aktor tidak mengisi bobot awal kriteria

Aktor	Sistem
9. Tidak mengisi bobot awal kriteria	
10. Menekan tombol “Pembobotan <i>Entropy</i> ”	
	11. Memeriksa masukan bobot awal kriteria
	12. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada masukan bobot awal kriteria. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan

Jika aktor mengisi masukan angka dengan huruf

Aktor	Sistem
9. Mengisi isian masukan bobot awal kriteria dengan huruf	
10. Menekan tombol “Pembobotan <i>Entropy</i> ”	
	11. Memeriksa masukan bobot awal kriteria
	12. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan akan muncul “input harus berupa angka”

Jika bobot awal yang dimasukkan jumlahnya tidak sama dengan 1

Aktor	Sistem
9. Mengisi isian masukan bobot awal kriteria	

- 10. Menekan tombol “Pembobotan *Entropy*”
- 11. Memeriksa masukan bobot awal kriteria
- 12. Menampilkan message “Jumlah bobot awal harus sama dengan 1”

Jika aktor mengubah analisa pengambilan keputusan yang pernah dilakukan

Aktor	Sistem
4. Menekan tombol “Ubah DSS”	5. Menampilkan alert “Analisa Anda sebelumnya akan terhapus, apakah Anda yakin akan mengubah analisa?” disertai tombol OK dan Cancel
6. Menekan tombol “OK” pada <i>alert</i>	9. Menghapus analisa pengambilan keputusan yang sudah pernah dilakukan oleh <i>decision maker</i> yang sedang mengakses sistem dari <i>database</i>
	10. Mengambil data kriteria dari <i>database</i>
	11. Menampilkan data kriteria, <i>field</i> masukan bobot awal kriteria pada tabel di halaman pilih kriteria dan tombol “Tambah Kriteria”

Jika aktor batal mengubah analisa pengambilan keputusan yang sudah pernah dilakukan

Aktor	Sistem
6. Menekan tombol “ <i>Cancel</i> ” pada <i>alert</i>	7. Mengambil data pengambilan keputusan yang hanya dilakukan oleh DM dari <i>database</i>
	8. Menampilkan data pengambilan keputusan pada tabel di halaman data pengambilan keputusan

3. *Usecase Skenario Login*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *login* dapat dilihat pada Lampiran A (*Usecase Skenario*).

4. *Usecase Skenario Logout*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *usecase* skenario *logout* dapat dilihat pada Lampiran A (*Usecase Skenario*).

5. *Usecase Skenario Melihat Informasi Peta*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat informasi peta dapat dilihat pada Lampiran A (*Usecase Skenario*).

6. *Usecase Skenario Mengelola Data User*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data *user* dapat dilihat pada Lampiran A (*Usecase Skenario*).

7. *Usecase Skenario Mengelola Data Kecamatan*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data kecamatan dapat dilihat pada Lampiran A (*Usecase Skenario*).

8. *Usecase Skenario Mengelola Data Kelurahan*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data kelurahan dapat dilihat pada Lampiran A (*Usecase Skenario*).

9. *Usecase* Skenario Mengelola Data Kriteria

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data kriteria dapat dilihat pada Lampiran A (*Usecase* Skenario).

10. *Usecase* Skenario Mengelola Batasan Kriteria

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola batasan kriteria dapat dilihat pada Lampiran A (*Usecase* Skenario).

11. *Usecase* Skenario Mengelola Nilai Kriteria

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat nilai kriteria dapat dilihat pada Lampiran A (*Usecase* Skenario).

12. *Usecase* Skenario Mengelola Data Kategori Informasi Peta

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data kategori informasi peta dapat dilihat pada Lampiran A (*Usecase* Skenario).

13. *Usecase* Skenario Mengelola Nilai Kategori Informasi Peta

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola nilai kategori informasi peta dapat dilihat pada Lampiran A (*Usecase* Skenario).

14. *Usecase* Skenario Mengelola Biodata

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola biodata dapat dilihat pada Lampiran A (*Usecase* Skenario).

15. *Usecase* Skenario Melihat Data ATM

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *usecase* skenario melihat data ATM dapat dilihat pada Lampiran A (*Usecase* Skenario).

16. *Usecase* Skenario Mengubah Akun *User*

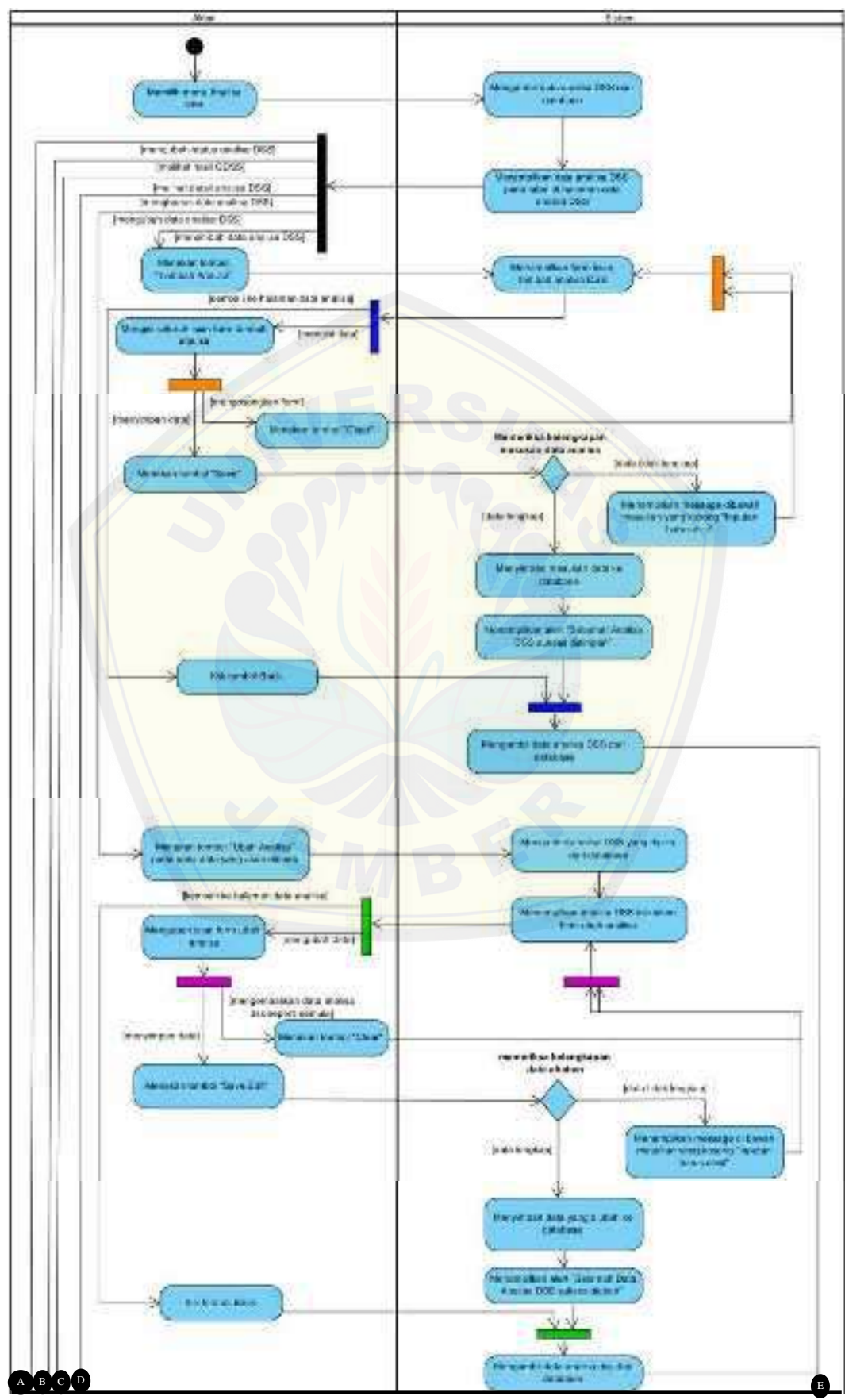
Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah akun *user* dapat dilihat pada Lampiran A (*Usecase* Skenario).

4.3.4 *Activity* Diagram

Activity Diagram menggambarkan aliran aktivitas dalam *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM yang akan dibangun. *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM ini memiliki enam belas *activity* diagram yaitu sebagai berikut:

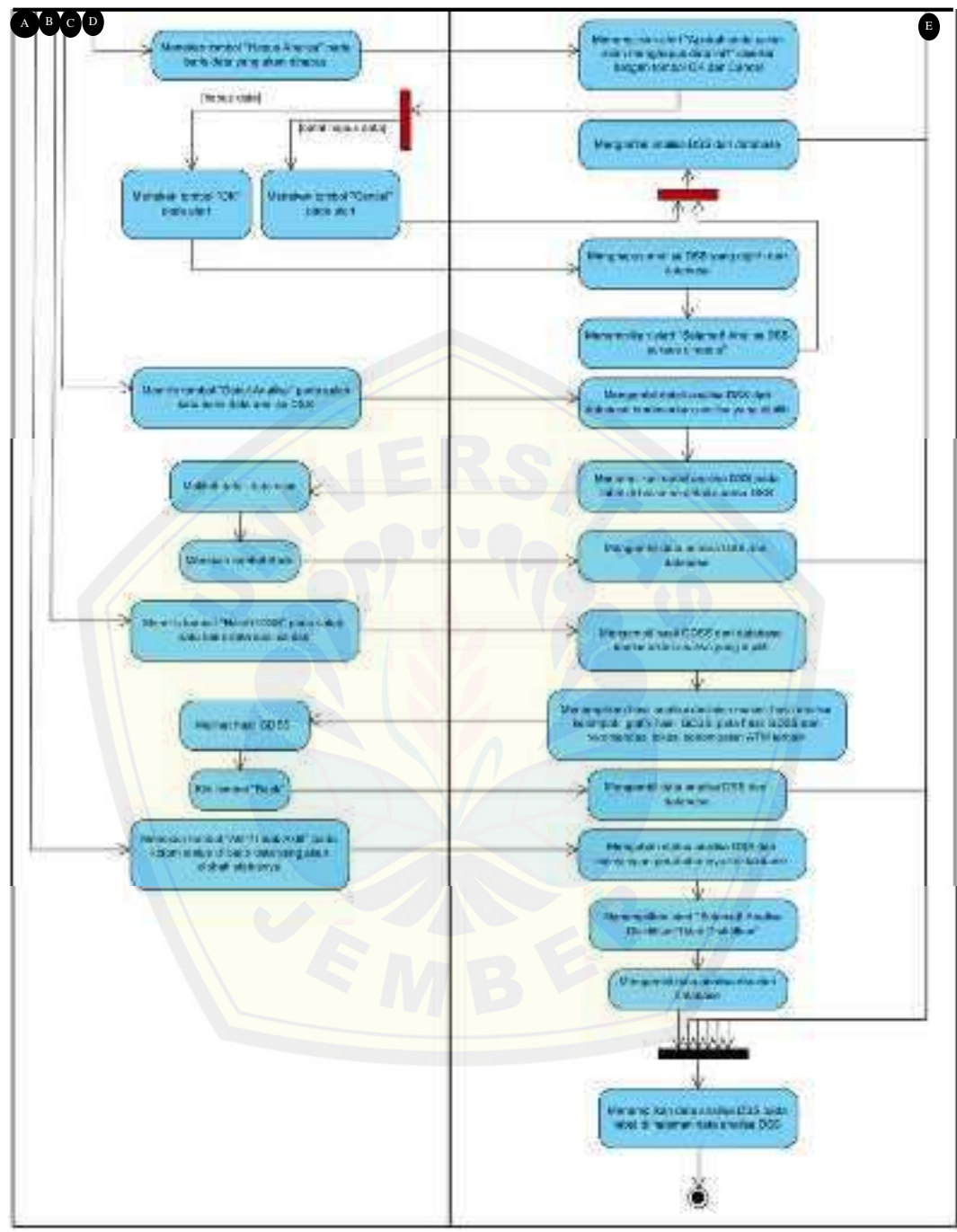
1. *Activity* Diagram Mengelola Analisa DSS

Activity diagram mengelola analisa DSS dapat dilakukan oleh admin. *Activity* diagram mengelola analisa DSS menjelaskan tentang admin menggunakan *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM untuk melihat, menambah, mengubah dan menghapus data analisa pengambilan keputusan yang dilakukan. *Activity* diagram mengelola analisa DSS dapat dilihat pada Gambar 4.3.

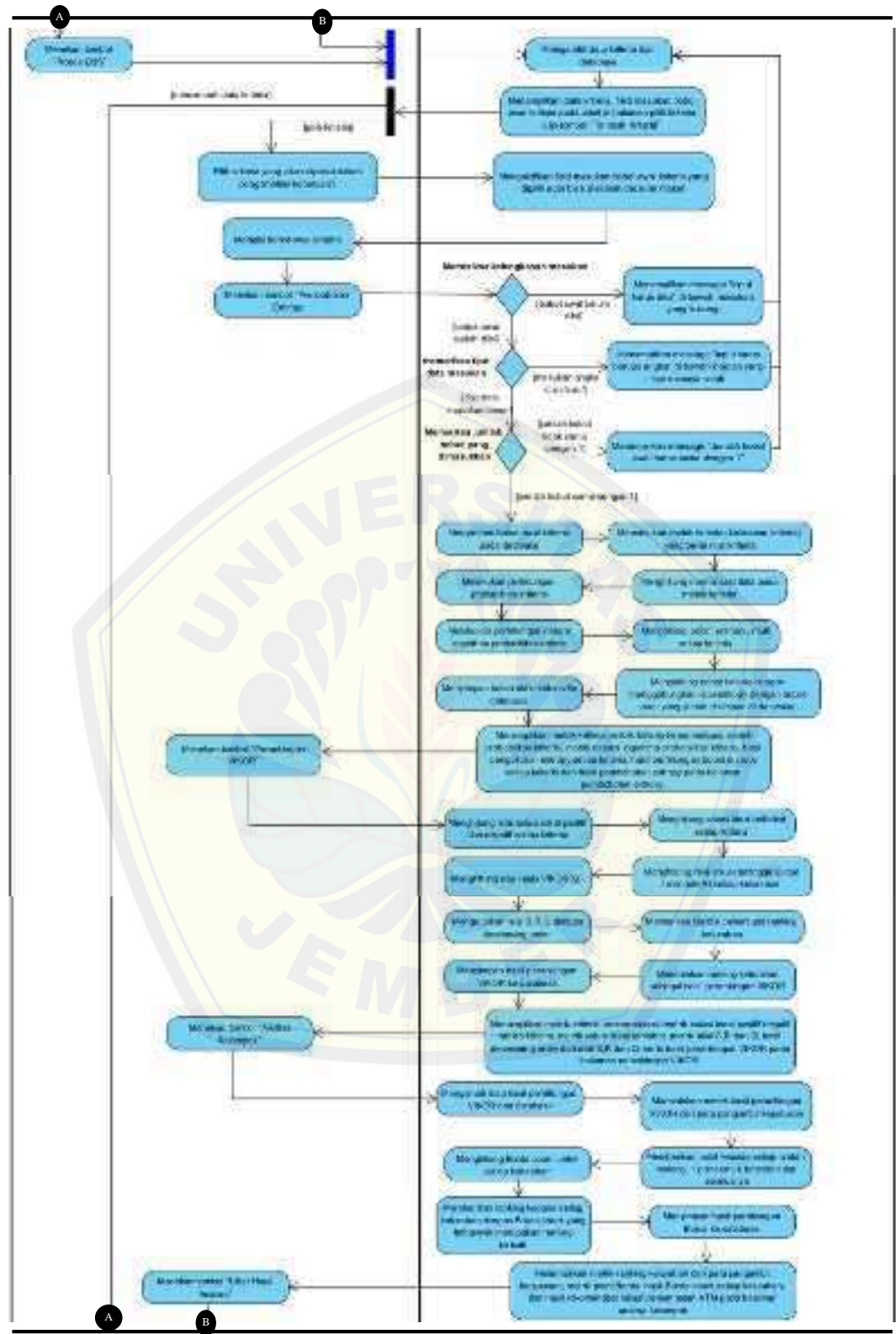


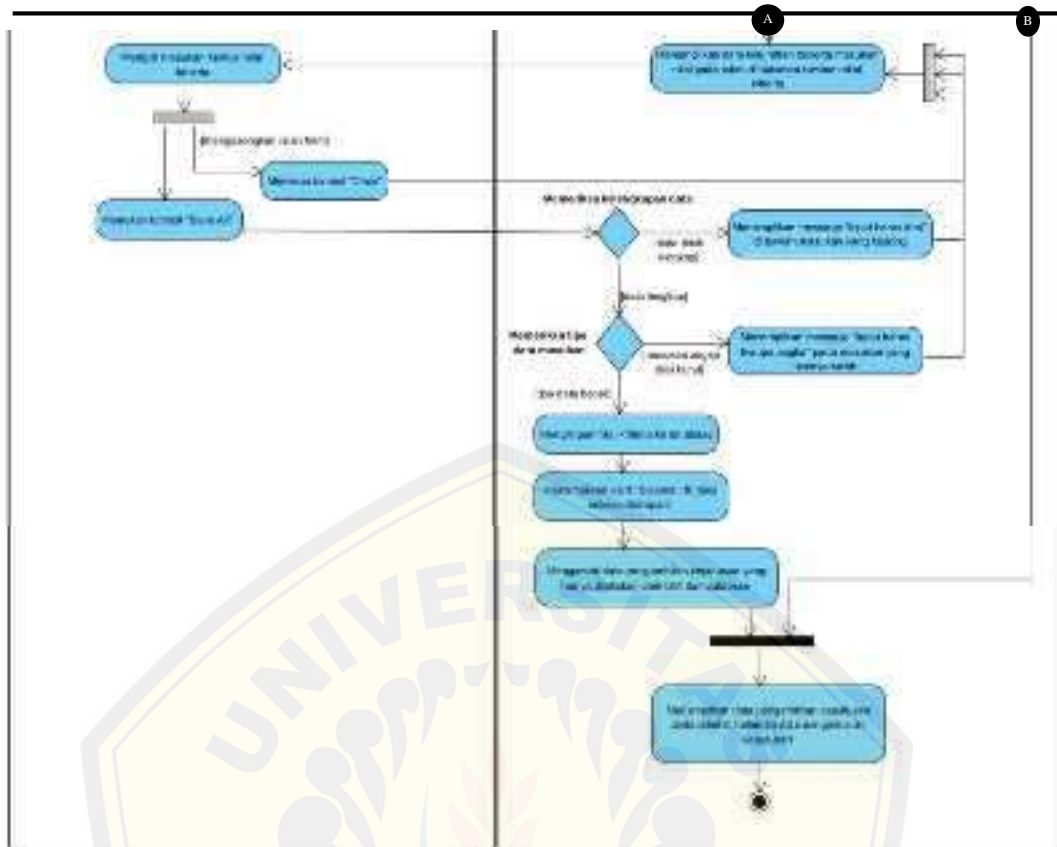
A B C D

E



Gambar 4.3 Activity Diagram Mengelola Analisa DSS





Gambar 4.4 Activity Diagram Memroses Pengambilan Keputusan

3. Activity Diagram Login

Activity diagram login dapat dilakukan oleh admin dan pengambil keputusan. Activity diagram login menjelaskan tentang bagaimana admin dan pengambil keputusan dapat masuk ke dalam sistem Group Decision Support System (GDSS) penentuan lokasi penempatan ATM. Activity diagram login dapat dilihat pada Lampiran B (Activity Diagram).

4. Activity Diagram Logout

Activity diagram logout dapat dilakukan oleh admin dan pengambil keputusan. Activity diagram logout menjelaskan tentang bagaimana admin dan pengambil keputusan dapat keluar dari sistem Group Decision Support System (GDSS) penentuan lokasi penempatan ATM. Activity diagram logout dapat dilihat pada Lampiran B (Activity Diagram).

5. *Activity Diagram Melihat Informasi Peta*

Activity diagram melihat informasi peta dapat dilakukan oleh admin, pengambil keputusan dan nasabah. *Activity diagram melihat informasi peta* menjelaskan tentang bagaimana admin, pengambil keputusan dan nasabah menggunakan *Group Decision Support System (GDSS)* penentuan lokasi penempatan ATM untuk melihat informasi persebaran ATM BRI dan kategori lainnya yang ditampilkan dalam peta. *Activity diagram melihat informasi peta* dapat dilihat pada Lampiran B (*Activity Diagram*).

6. *Activity Diagram Mengelola Data User*

Activity diagram mengelola data user dapat dilakukan oleh admin. *Activity diagram mengelola data user* menjelaskan tentang admin menggunakan *Group Decision Support System (GDSS)* penentuan lokasi penempatan ATM untuk melihat, menambah, mengubah dan menghapus data *user*. *Activity diagram mengelola data user* dapat dilihat pada Lampiran B (*Activity Diagram*).

7. *Activity Diagram Mengelola Data Kecamatan*

Activity diagram mengelola data kecamatan dapat dilakukan oleh admin. *Activity diagram mengelola data kecamatan* menjelaskan tentang admin menggunakan *Group Decision Support System (GDSS)* penentuan lokasi penempatan ATM untuk melihat detail kecamatan dalam bentuk peta, menambah, mengubah dan menghapus data kecamatan. *Activity diagram mengelola data kecamatan* dapat dilihat pada Lampiran B (*Activity Diagram*).

8. *Activity Diagram Mengelola Data Kelurahan*

Activity diagram mengelola data kelurahan dapat dilakukan oleh admin. *Activity diagram mengelola data kelurahan* menjelaskan tentang admin menggunakan *Group Decision Support System (GDSS)* penentuan lokasi penempatan ATM untuk melihat detail kelurahan dalam bentuk peta, menambah, mengubah dan menghapus data kelurahan. *Activity diagram mengelola data kelurahan* dapat dilihat pada Lampiran B (*Activity Diagram*).

9. *Activity Diagram Mengelola Data Kriteria*

Activity diagram mengelola data kriteria dapat dilakukan oleh admin. Activity diagram mengelola data kriteria menjelaskan tentang admin menggunakan Group Decision Support System (GDSS) penentuan lokasi penempatan ATM untuk melihat, menambah, mengubah dan menghapus data kriteria yang digunakan dalam proses pengambilan keputusan. Activity diagram mengelola data kriteria dapat dilihat pada Lampiran B (Activity Diagram).

10. *Activity Diagram Mengelola Batasan Kriteria*

Activity diagram mengelola batasan kriteria dapat dilakukan oleh admin. Activity diagram mengelola batasan kriteria menjelaskan tentang admin menggunakan Group Decision Support System (GDSS) penentuan lokasi penempatan ATM untuk melihat, menambah dan mengubah batasan kriteria berdasarkan data-data kriteria yang sudah dimasukkan. Activity diagram mengelola batasan kriteria dapat dilihat pada Lampiran B (Activity Diagram).

11. *Activity Diagram Mengelola Nilai Kriteria*

Activity diagram mengelola nilai kriteria dapat dilakukan oleh admin. Activity diagram mengelola nilai kriteria menjelaskan tentang administrator menggunakan Group Decision Support System (GDSS) penentuan lokasi penempatan ATM untuk melihat, menambah, mengubah dan menghapus nilai kriteria berdasarkan data kriteria dan batasan kriteria yang sudah dimasukkan. Activity diagram mengelola nilai kriteria dapat dilihat pada Lampiran B (Activity Diagram).

12. *Activity Diagram Mengelola Data Kategori Informasi Peta*

Activity diagram mengelola data kategori informasi peta dapat dilakukan oleh admin. Activity diagram mengelola data kategori informasi peta menjelaskan tentang admin menggunakan Group Decision Support System (GDSS) penentuan lokasi penempatan ATM untuk melihat, menambah, mengubah dan menghapus kategori informasi yang akan ditampilkan pada peta berdasarkan data kriteria dan

batasan kriteria yang sudah dimasukkan. *Activity* diagram mengelola data kategori informasi peta dapat dilihat pada Lampiran B (*Activity* Diagram).

13. *Activity* Diagram Mengelola Nilai Kategori Informasi Peta

Activity diagram mengelola data kategori informasi peta dapat dilakukan oleh admin. *Activity* diagram mengelola nilai kategori informasi peta menjelaskan tentang admin menggunakan *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM untuk melihat, menambah, mengubah dan menghapus nilai kategori informasi peta berdasarkan data kategori informasi peta yang telah dimasukkan. *Activity* diagram mengelola nilai kategori informasi peta dapat dilihat pada Lampiran B (*Activity* Diagram).

14. *Activity* Diagram Mengelola Biodata

Activity diagram mengelola biodata dapat dilakukan oleh pengambil keputusan. *Activity* diagram mengelola biodata menjelaskan tentang pengambil keputusan menggunakan *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM untuk melihat dan mengubah biodata pengambil keputusan yang sedang mengakses sistem. *Activity* diagram mengelola biodata dapat dilihat pada Lampiran B (*Activity* Diagram).

15. *Activity* Diagram Melihat Data ATM

Activity diagram melihat data ATM dapat dilakukan oleh nasabah. *Activity* diagram melihat data ATM menjelaskan tentang nasabah menggunakan *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM untuk melihat data ATM BRI yang ada di Kabupaten Jember. *Activity* diagram melihat data ATM dapat dilihat pada Lampiran B (*Activity* Diagram).

16. *Activity* Diagram Mengubah Akun *User*

Activity diagram mengubah akun user dapat dilakukan oleh admin dan pengambil keputusan. *Activity* diagram mengubah akun *user* menjelaskan tentang admin dan pengambil keputusan menggunakan *Group Decision Support System*

(GDSS) penentuan lokasi penempatan ATM untuk mengubah data akun *user*. *Activity* diagram mengubah akun *user* dapat dilihat pada Lampiran B (*Activity* Diagram).

4.3.5 *Sequence* Diagram

Sequence Diagram adalah dokumentasi suatu diagram terurut yang menampilkan interaksi - interaksi antar objek di dalam sistem. *Sequence* diagram digunakan untuk menggambarkan skenario dan memodelkan aliran logika dalam sistem dengan cara *visual*. *Sequence* diagram dari *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM adalah sebagai berikut:

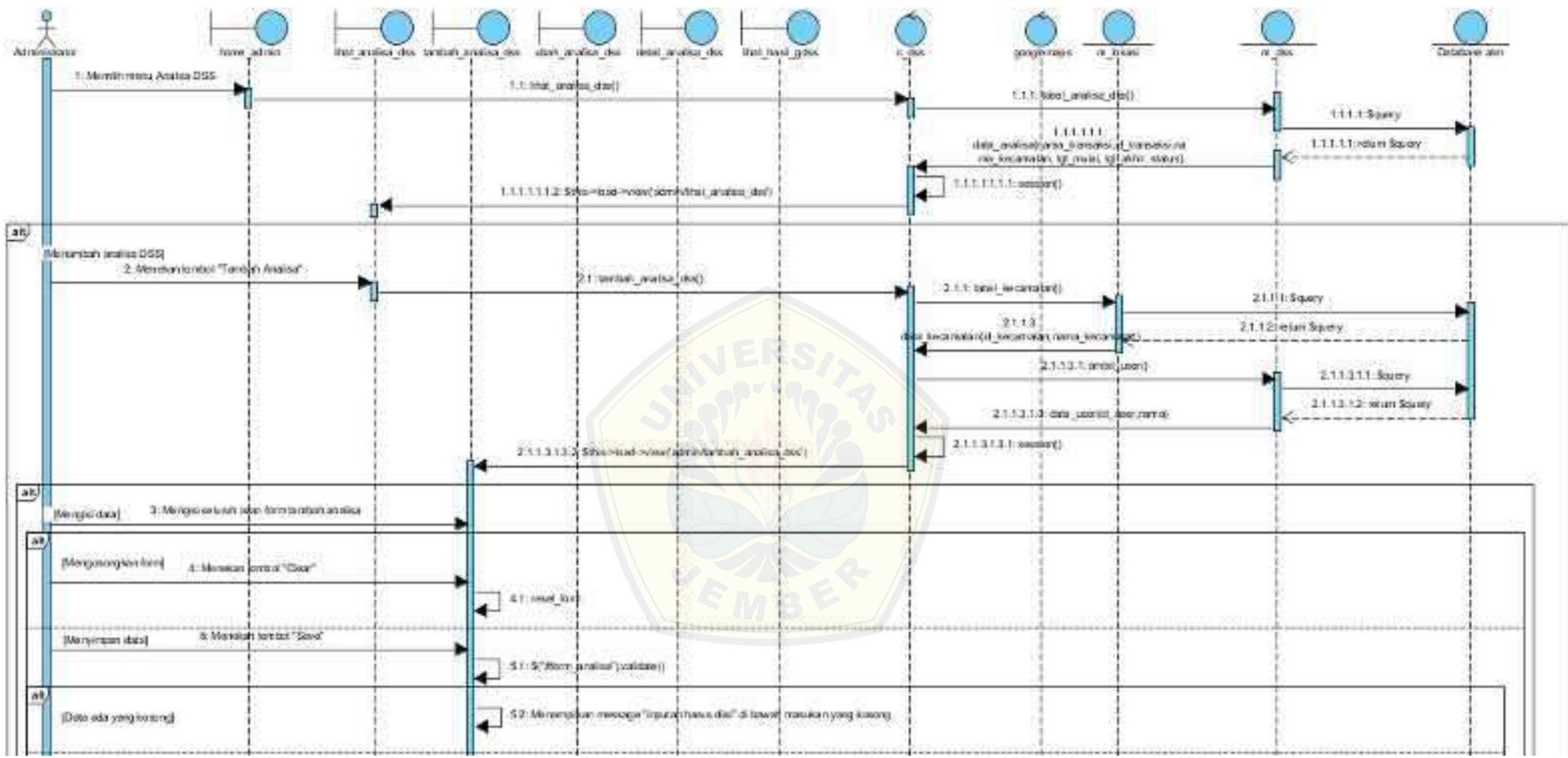
1. *Sequence* Diagram Mengelola Analisa DSS

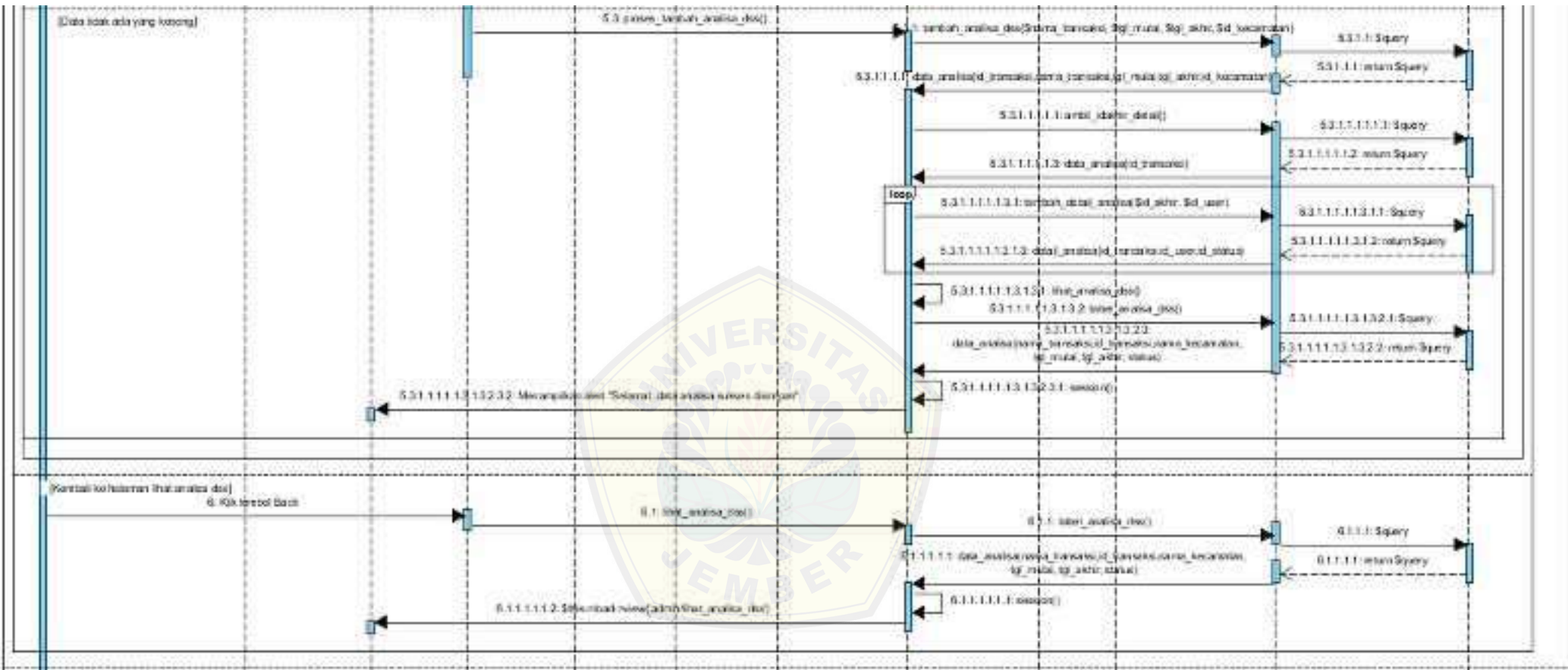
Penggambaran *sequence* diagram mengelola analisa DSS digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence* diagram mengelola analisa DSS adalah *view home_admin*, *view lihat_analisa_dss*, *view tambah_analisa_dss*, *view ubah_analisa_dss*, *view detail_analisa_dss*, *view lihat_hasil_gdss*, *c_dss*, *library googlemaps*, *m_lokasi* dan *m_dss*. Proses dari aliran *sequence* diagram mengelola analisa DSS adalah ketika aktor memilih menu analisa DSS maka sistem menjalankan fungsi *lihat_kriteria()* yang ada di *class c_dss*. Di dalam fungsi *lihat_analisa_dss()*, sistem akan memanggil fungsi *tabel_analisa_dss()* pada *class m_dss*. Fungsi *tabel_analisa_dss()* akan mengakses data analisa dss dari *database* sehingga sistem dapat menampilkan halaman data analisa DSS. Aktor dapat melihat, menambah, mengubah, dan menghapus data analisa DSS serta melihat hasil analisa DSS dalam bentuk grafik dan peta. *Sequence* diagram mengelola analisa DSS oleh admin dijelaskan pada Gambar 4.5.

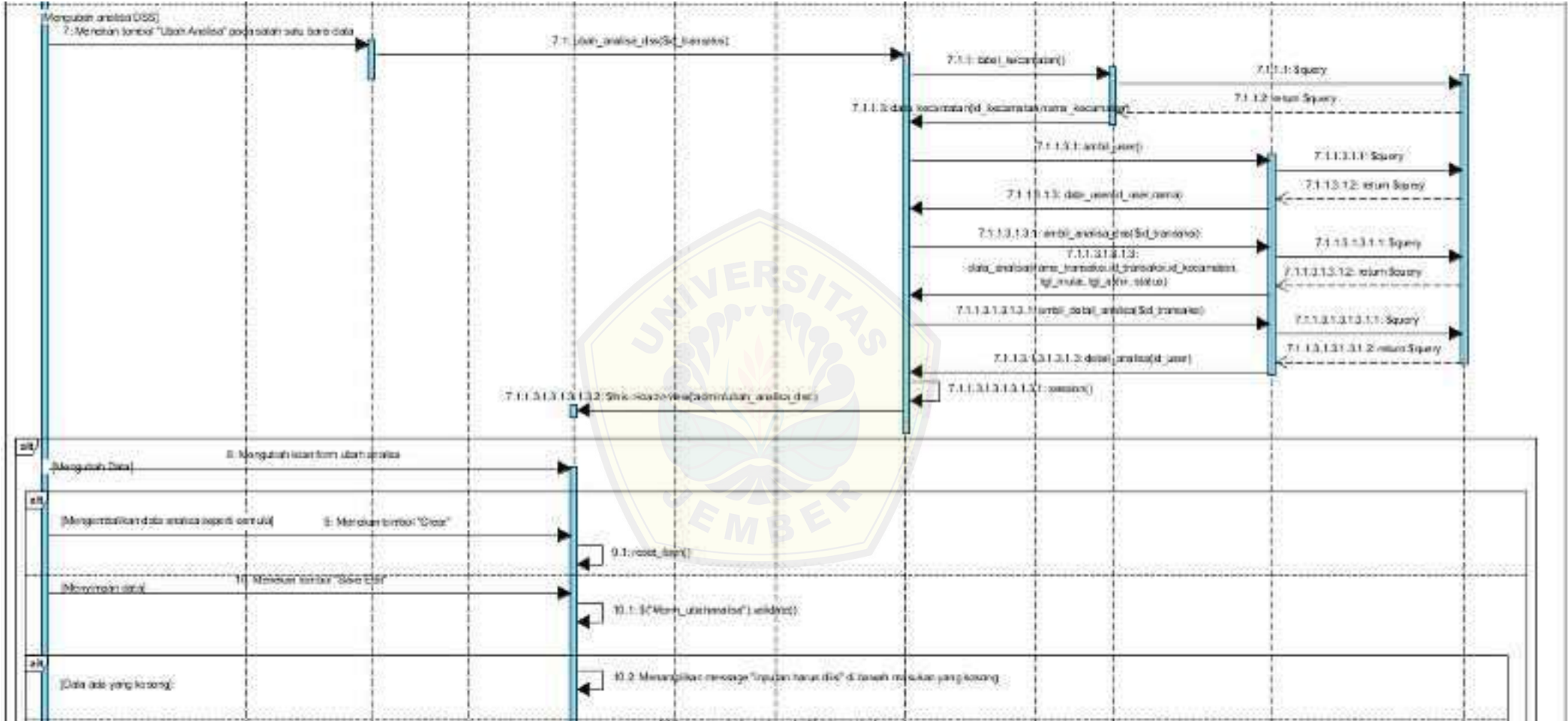
2. *Sequence* Diagram Memroses Pengambilan Keputusan

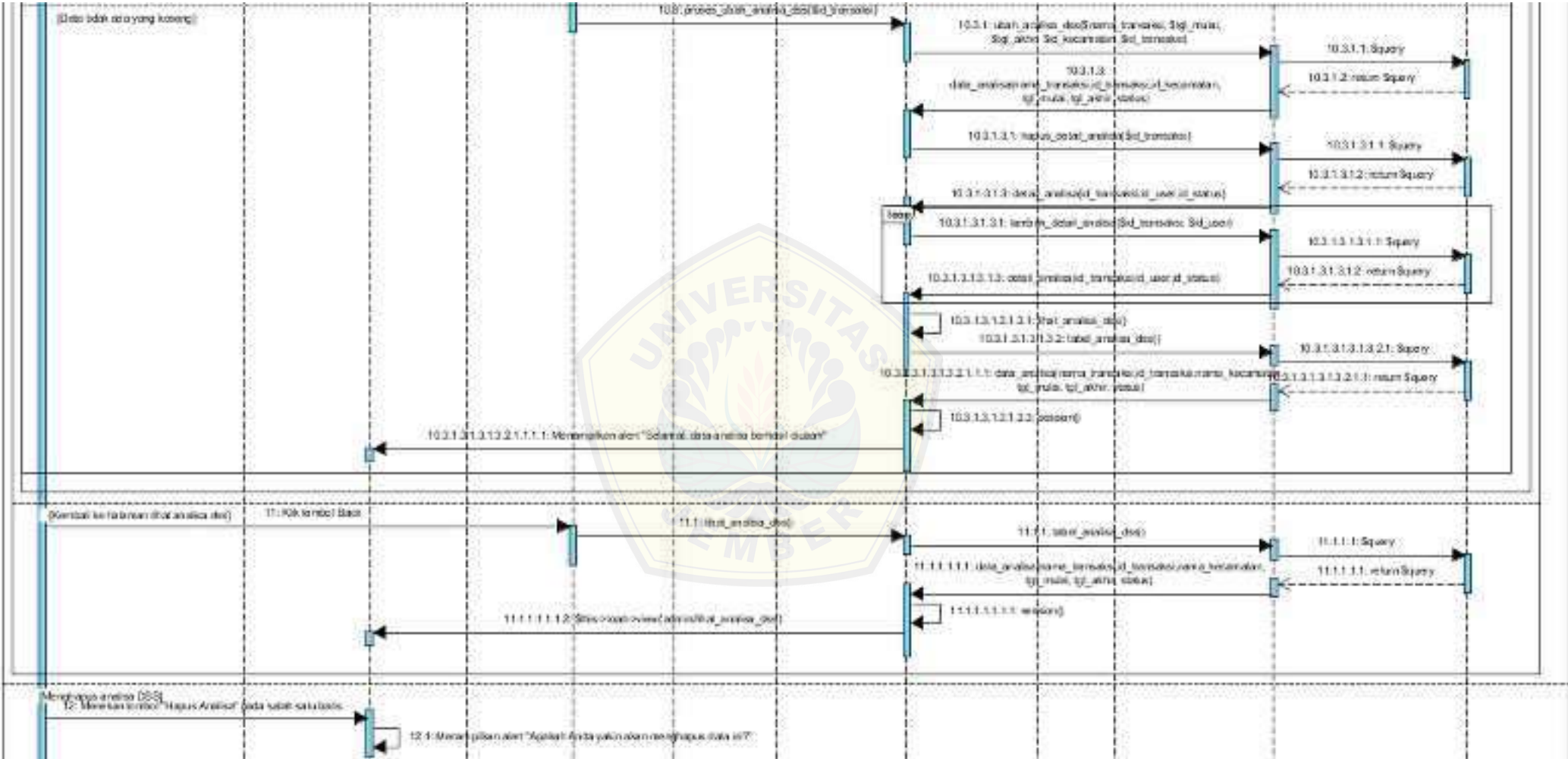
Penggambaran *sequence* diagram memroses pengambilan keputusan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing –

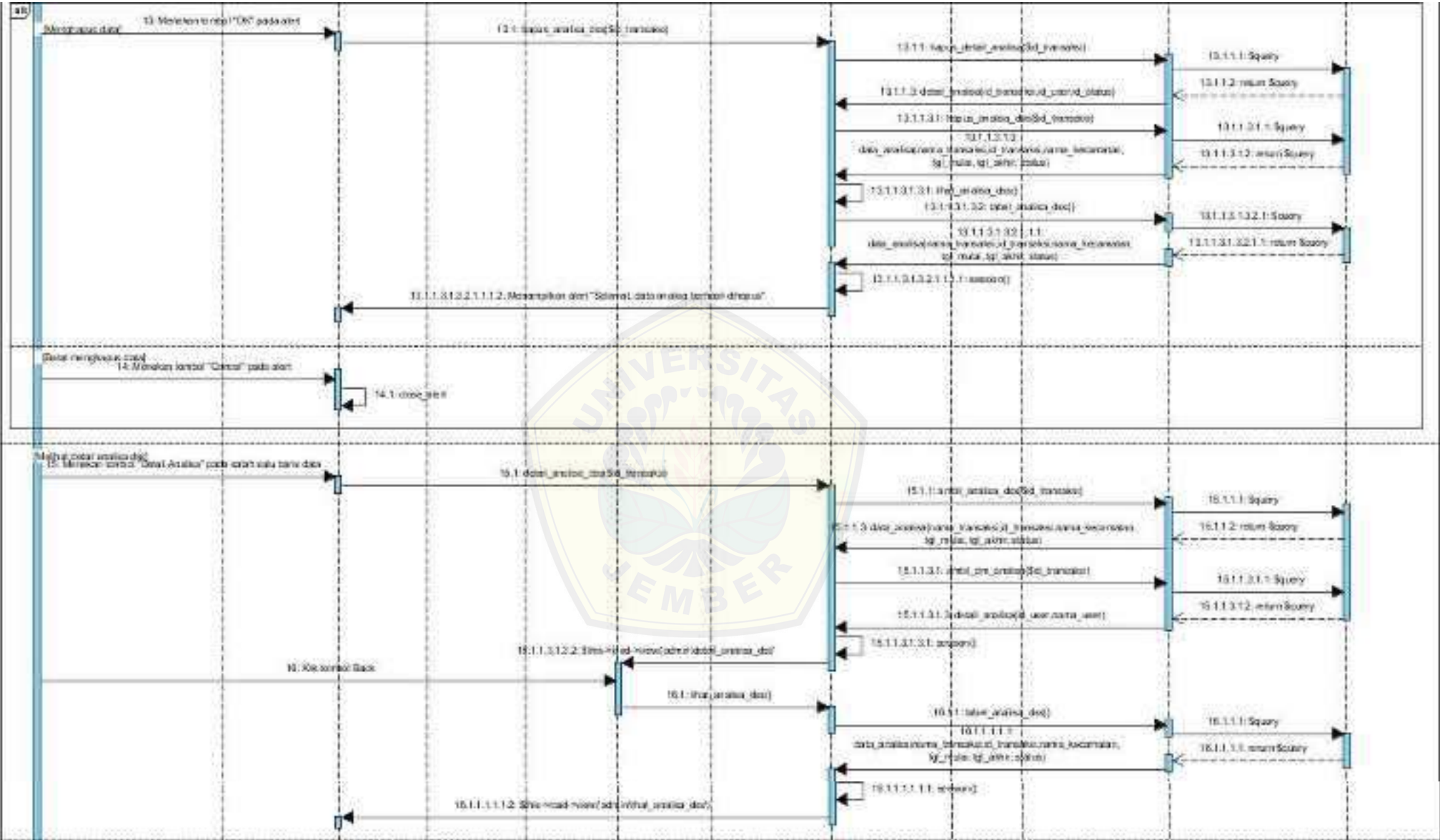
masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence* diagram memroses pengambilan keputusan adalah *view home_dm*, *view lihat_analisa*, *view lihat_hasil_dss*, *view lihat_hasil_gdss*, *view proses1*, *view metode_entropy*, *view metode_vikor*, *view metode_borda*, *view tambah_kriteria*, *view lihat_range*, *view ubah_range*, *view tambah_nilai_kriteria*, *c_dss*, *m_dss*, *m_user*, *m_kriteria*, dan *m_lokasi*. Proses dari aliran *sequence* diagram memroses pengambilan keputusan adalah ketika aktor memilih menu pengambilan keputusan maka sistem menjalankan fungsi *lihat_analisa_dm()* yang berparameter *id* yang ada di *class c_dss*. Di dalam fungsi *lihat_analisa_dm()*, sistem akan memanggil fungsi *tabel_transaksi()* yang berparameter *id* pada *class m_dss*. Fungsi tersebut akan mengakses data pengambilan keputusan dari *database* sehingga sistem dapat menampilkan halaman data pengambilan keputusan. Aktor dapat melihat hasil DSS, melihat hasil GDSS dan memroses pengambilan keputusan. Pada *sequence* diagram memroses pengambilan keputusan inilah dijelaskan tentang proses perhitungan lokasi terbaik menggunakan metode *Entropy*, *VIKOR* dan *Borda*. *Sequence* diagram memroses pengambilan keputusan oleh Pengambil Keputusan dijelaskan pada Gambar 4.6.

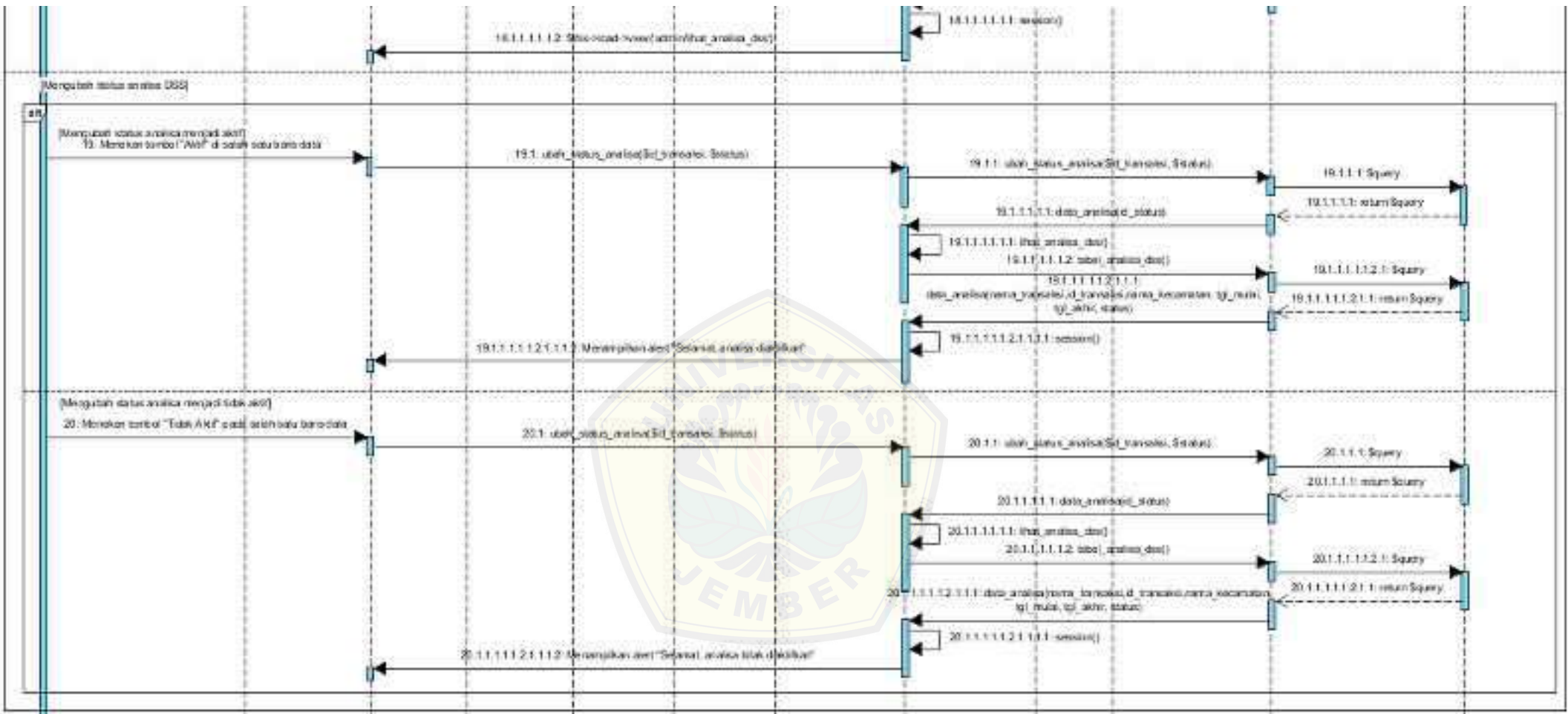




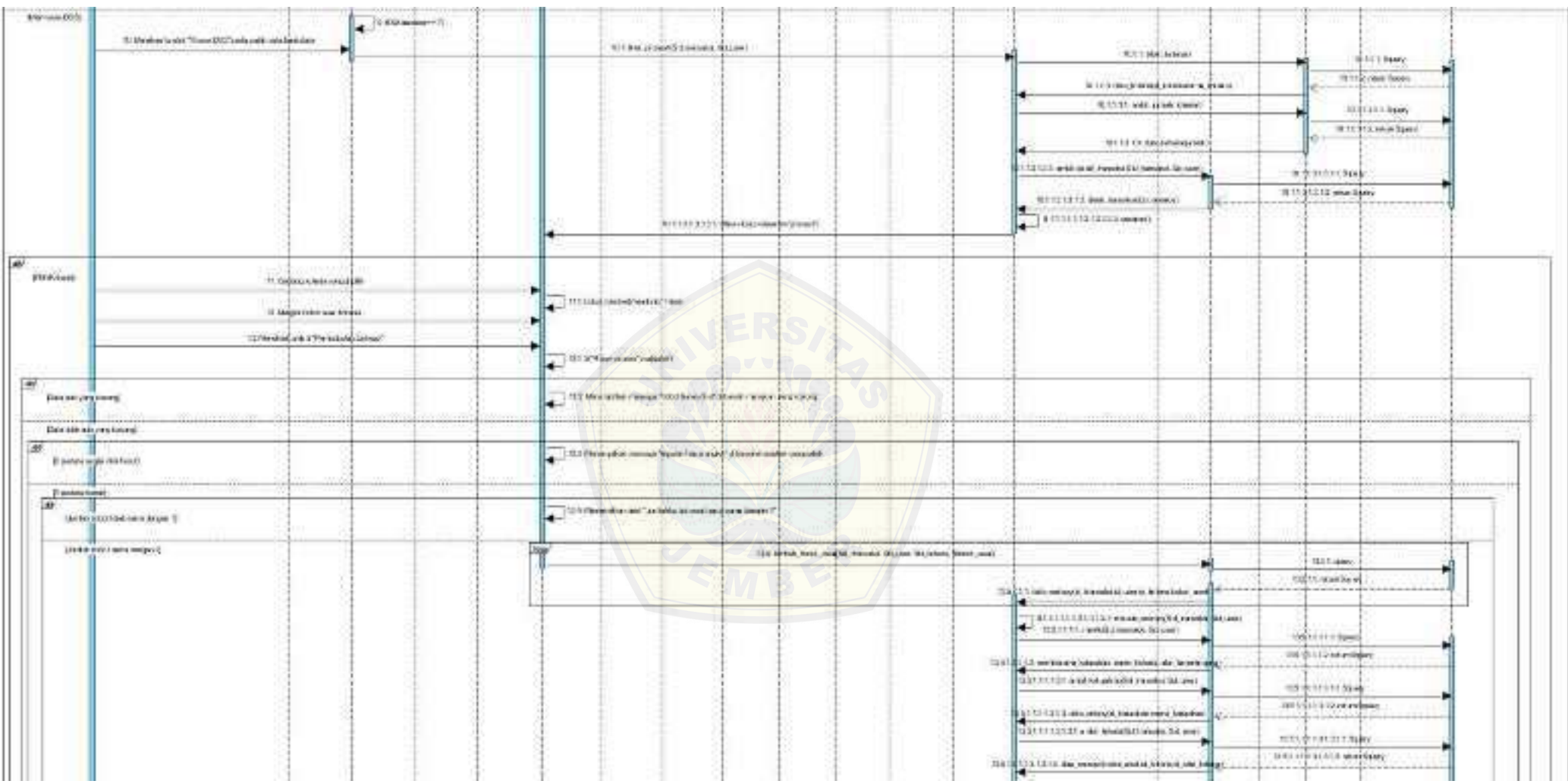


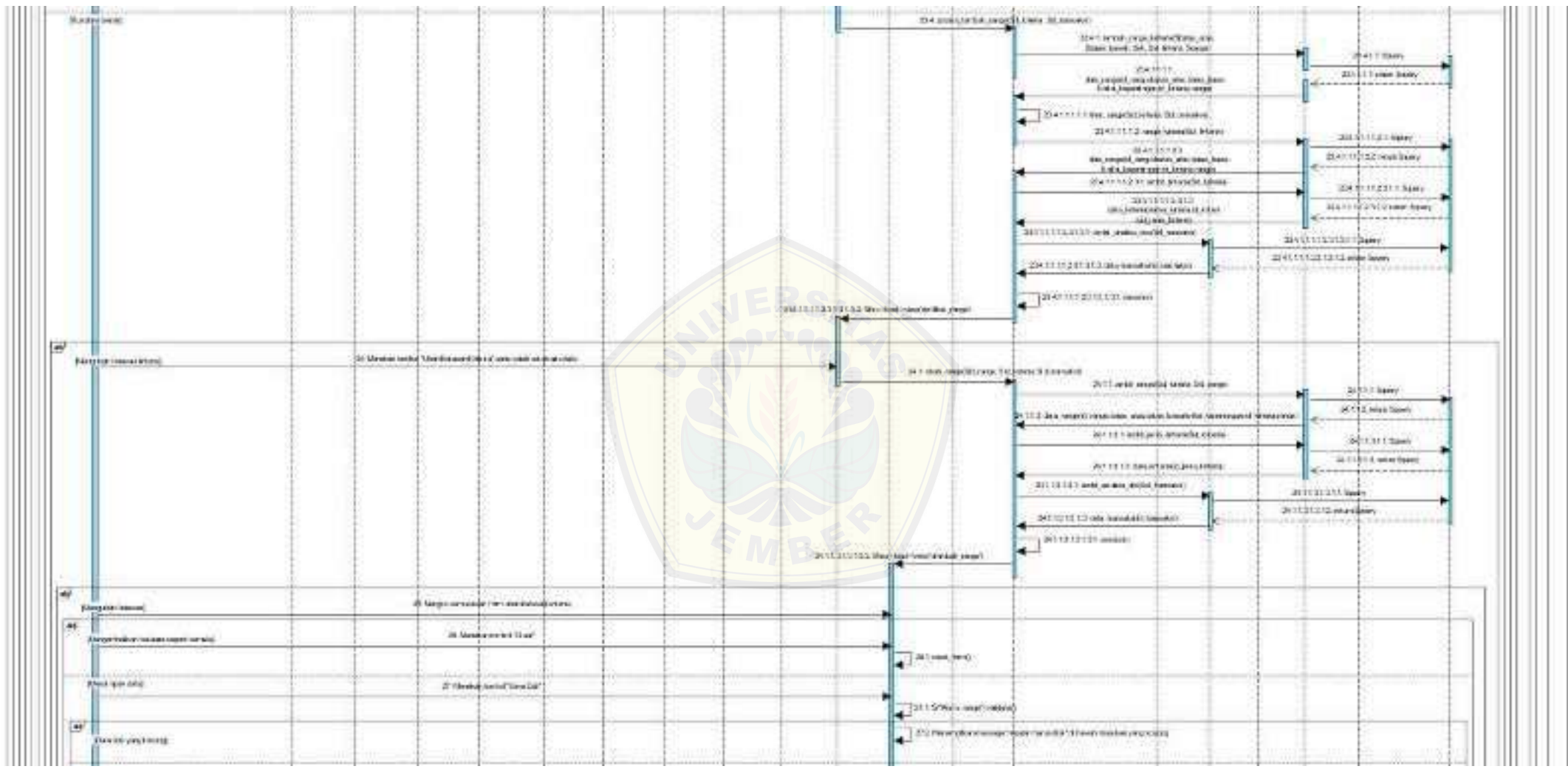


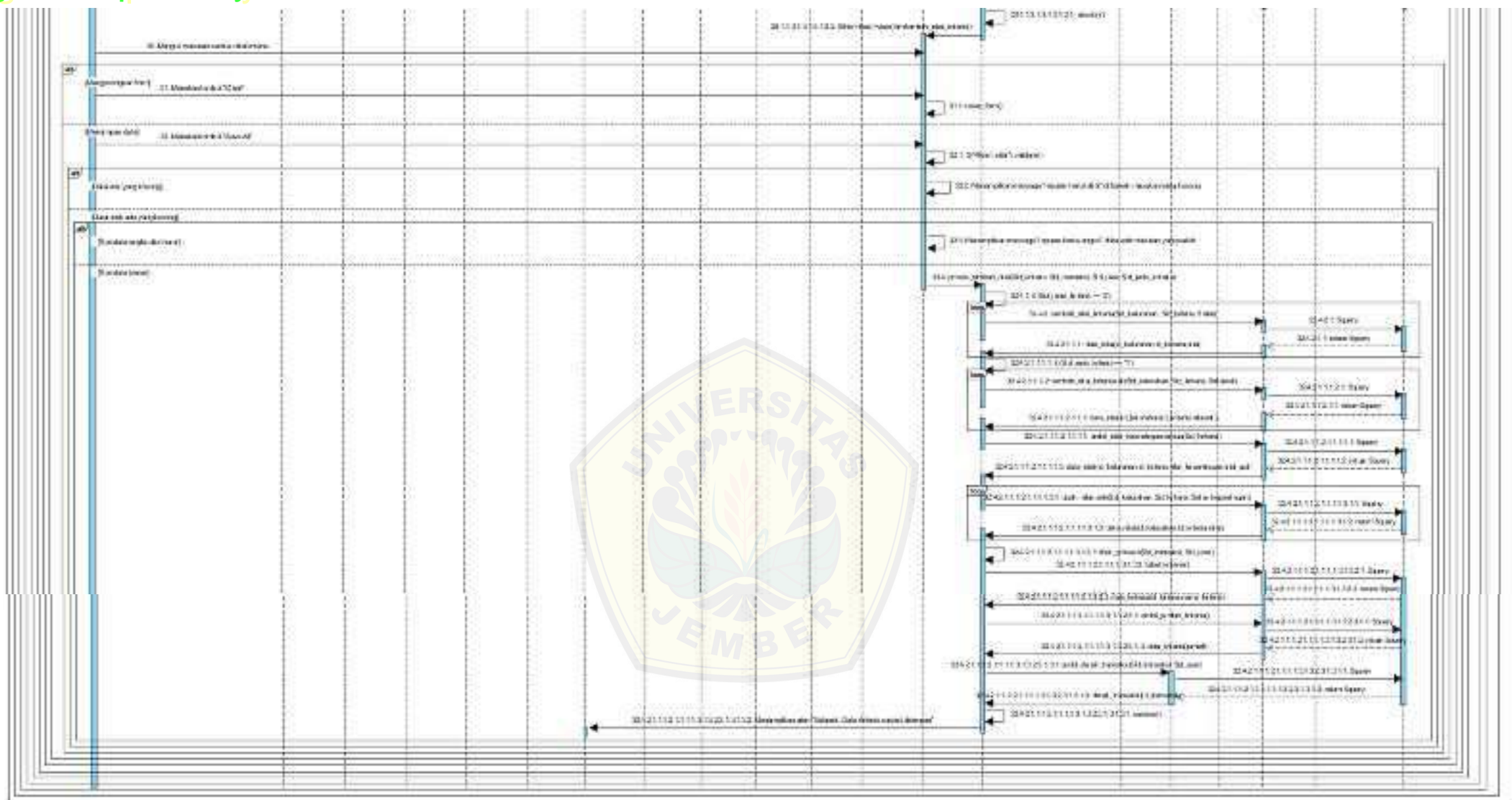




Gambar 4.5 Sequence Diagram Mengelola Analisa DSS







Gambar 4.6 Sequence Diagram Memroses Pengambilan Keputusan

3. *Sequence Diagram Login*

Penggambaran *sequence diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence diagram login* adalah *view login, home_admin, home_dm, c_home, c_login, m_login, library googlemaps, m_lokasi* dan *m_kriteria*. Proses dari aliran *sequence diagram login* yang pertama adalah sistem akan memeriksa kelengkapan data *login* yang dimasukkan oleh aktor. Apabila data *login* sudah lengkap maka ketika aktor menekan tombol *login*, sistem menjalankan fungsi *proses()* yang ada di *class c_login*. Di dalam fungsi *proses()*, sistem akan memanggil fungsi *login()* pada *class m_login* dengan parameter *username* dan *password*. Pada fungsi *login()* di *m_login* inilah sistem akan berhubungan dengan *database* dan akan memeriksa apakah data *login* tersebut benar atau tidak. Jika data *login* benar maka sistem akan menampilkan halaman *home* sesuai level masing-masing aktor. Sebaliknya jika data *login* salah maka akan muncul peringatan pada halaman *login*. *Sequence diagram login* oleh admin dan pengambil keputusan dijelaskan pada Lampiran C (*Sequence Diagram*).

4. *Sequence Diagram Logout*

Penggambaran *sequence diagram logout* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence diagram logout* adalah *view home* dan *c_login*. Proses dari aliran *sequence diagram logout* adalah ketika aktor menekan tombol *logout* maka sistem menghapus ijin akses sistem aktor. *Sequence diagram logout* oleh admin dan pengambil keputusan dijelaskan pada Lampiran C (*Sequence Diagram*).

5. *Sequence Diagram Melihat Informasi Peta*

Penggambaran *sequence diagram melihat informasi peta* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence*

diagram melihat informasi adalah *view home_admin*, *view home_user*, *view home_dm*, *c_home*, *library googlemaps*, *m_lokasi*, dan *m_kriteria*. Proses dari aliran *sequence* diagram melihat informasi peta adalah ketika aktor memilih menu *dashboard* maka sistem menjalankan fungsi *home()* yang ada di *class c_home*. Di dalam fungsi *home()*, sistem akan memanggil beberapa fungsi seperti *data_kelurahan()* sampai *combo_kategori()* pada *class* model dan memanggil fungsi *create_map()* pada *library googlemaps* agar dapat menampilkan peta. Beberapa fungsi yang dipanggil dari model akan mengakses data dari *database* sehingga akan muncul halaman *home* sesuai level aktor masing-masing yang berisi peta persebaran ATM dan kategori lainnya. *Sequence* diagram melihat informasi peta oleh admin, pengambil keputusan dan nasabah dijelaskan pada Lampiran C (*Sequence Diagram*).

6. *Sequence Diagram* Mengelola Data *User*

Penggambaran *sequence* diagram mengelola data *user* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence* diagram mengelola data *user* adalah *view home_admin*, *view lihat_user*, *view tambah_user*, *view ubah_user*, *view detail_user*, *c_user*, dan *m_user*. Proses dari aliran *sequence* diagram mengelola data *user* adalah ketika aktor memilih menu *user* maka sistem menjalankan fungsi *lihat_user()* yang ada di *class c_user*. Di dalam fungsi *lihat_user()*, sistem akan memanggil fungsi *tabel_user()* pada *class m_user*. Fungsi *tabel_user()* akan mengakses data *user* dari *database* sehingga sistem dapat menampilkan halaman data *user*. Aktor dapat melihat detail *user*, menambah, mengubah, dan menghapus data *user*. *Sequence* diagram mengelola data *user* oleh admin dijelaskan pada Lampiran C (*Sequence Diagram*).

7. *Sequence Diagram* Mengelola Data Kecamatan

Penggambaran *sequence* diagram mengelola data kecamatan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam

sequence diagram mengelola data kecamatan adalah *view home_admin*, *view lihat_kecamatan*, *view tambah_kecamatan*, *view ubah_kecamatan*, *view detail_kecamatan*, *c_lokasi*, *library googlemaps*, dan *m_lokasi*. Proses dari aliran *sequence* diagram mengelola data kecamatan adalah ketika aktor memilih menu lokasi dan submenu data kecamatan maka sistem menjalankan fungsi *lihat_kecamatan()* yang ada di *class c_lokasi*. Di dalam fungsi *lihat_kecamatan()*, sistem akan memanggil fungsi *tabel_kecamatan()* pada *class m_lokasi*. Fungsi *tabel_kecamatan()* akan mengakses data kecamatan dari *database* sehingga sistem dapat menampilkan halaman data kecamatan. Aktor dapat melihat detail kecamatan, menambah, mengubah, dan menghapus data kecamatan. *Sequence* diagram mengelola data kecamatan oleh admin dijelaskan pada Lampiran C (*Sequence Diagram*).

8. *Sequence* Diagram Mengelola Data Kelurahan

Penggambaran *sequence* diagram mengelola data kelurahan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence* diagram mengelola data kelurahan adalah *view home_admin*, *view lihat_kelurahan*, *view tambah_kelurahan*, *view ubah_kelurahan*, *view detail_kelurahan*, *c_lokasi*, *library googlemaps*, dan *m_lokasi*. Proses dari aliran *sequence* diagram mengelola data kelurahan adalah ketika aktor memilih menu lokasi dan submenu data kelurahan maka sistem menjalankan fungsi *lihat_kelurahan()* yang ada di *class c_lokasi*. Di dalam fungsi *lihat_kelurahan()*, sistem akan memanggil fungsi *tabel_kelurahan()* pada *class m_lokasi*. Fungsi *tabel_kelurahan()* akan mengakses data kelurahan dari *database* sehingga sistem dapat menampilkan halaman data kelurahan. Aktor dapat melihat detail kelurahan, menambah, mengubah, dan menghapus data kelurahan. *Sequence* diagram mengelola data kelurahan oleh admin dijelaskan pada Lampiran C (*Sequence Diagram*).

9. *Sequence* Diagram Mengelola Data Kriteria

Penggambaran *sequence* diagram mengelola data kriteria digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence* diagram mengelola data kriteria adalah *view home_admin*, *view lihat_kriteria*, *view tambah_kriteria*, *view ubah_kriteria*, *c_kriteria*, dan *m_kriteria*. Proses dari aliran *sequence* diagram mengelola data kriteria adalah ketika aktor memilih menu data kriteria maka sistem menjalankan fungsi *lihat_kriteria()* yang ada di *class c_kriteria*. Di dalam fungsi *lihat_kriteria()*, sistem akan memanggil fungsi *tabel_kriteria()* pada *class m_kriteria*. Fungsi *tabel_kriteria()* akan mengakses data kriteria dari *database* sehingga sistem dapat menampilkan halaman data kriteria. Aktor dapat melihat, menambah, mengubah, dan menghapus data kriteria. *Sequence* diagram mengelola data kriteria oleh admin dijelaskan pada Lampiran C (*Sequence* Diagram).

10. *Sequence* Diagram Mengelola Batasan Kriteria

Penggambaran *sequence* diagram mengelola batasan kriteria digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence* diagram mengelola batasan kriteria adalah *view home_admin*, *view lihat_kriteria*, *view lihat_range_kriteria*, *view ubah_range*, *c_kriteria* dan *m_kriteria*. Proses dari aliran *sequence* diagram mengelola batasan kriteria adalah ketika aktor memilih menu data kriteria dan menekan tombol “Batasan Kriteria” pada salah satu baris data kriteria maka sistem menjalankan fungsi *lihat_range_kriteria()* dengan parameter *id_kriteria* yang ada di *class c_kriteria*. Di dalam fungsi *lihat_range_kriteria()*, sistem akan memanggil fungsi *range_kriteria()* dan *ambil_kriteria()* yang mempunyai parameter yang sama yaitu *id_kriteria* pada *class m_kriteria*. Fungsi *range_kriteria()* dan *ambil_kriteria()* akan mengakses batasan kriteria dari *database* sehingga sistem dapat menampilkan halaman batasan kriteria. Aktor dapat melihat, menambah, dan mengubah batasan kriteria. *Sequence* diagram

mengelola batasan kriteria oleh admin dijelaskan pada Lampiran C (*Sequence Diagram*).

11. *Sequence Diagram* Mengelola Nilai Kriteria

Penggambaran *sequence diagram* mengelola nilai kriteria digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence diagram* mengelola nilai kriteria adalah *view home_admin*, *view lihat_kriteria*, *view lihat_range_kriteria*, *view lihat_nilai_kriteria*, *view tambah_nilai_kriteria*, *view ubah_nilai_kriteria*, *c_kriteria*, *m_kriteria* dan *m_lokasi*. Proses dari aliran *sequence diagram* mengelola nilai kriteria adalah ketika aktor memilih menu data kriteria lalu menekan tombol “Batasan Kriteria” pada salah satu baris data kriteria dan selanjutnya menekan tombol “Nilai Kriteria” maka sistem menjalankan fungsi *lihat_nilai_kriteria()* dengan parameter *id_kriteria* dan *id_bentuk_kriteria* yang ada di *class c_kriteria*. Di dalam fungsi *lihat_range_kriteria()*, sistem akan memanggil fungsi *tabel_kelurahan()* dan *tabel_nilai_kriteria()* pada *class m_kriteria* dan *m_lokasi*. Fungsi-fungsi pada *class m_kriteria* dan *m_lokasi* tersebut akan mengakses nilai kriteria dari *database* sehingga sistem dapat menampilkan form menambah nilai kriteria. Aktor dapat melihat, mengubah dan menghapus nilai kriteria selain menambah nilai kriteria. *Sequence diagram* mengelola nilai kriteria oleh admin dijelaskan pada Lampiran C (*Sequence Diagram*).

12. *Sequence Diagram* Mengelola Data Kategori Informasi Peta

Penggambaran *sequence diagram* mengelola data kategori informasi peta digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence diagram* mengelola data kategori informasi peta adalah *view home_admin*, *view lihat_kriteria*, *view lihat_range_kriteria*, *view lihat_kategori*, *view tambah_kategori*, *view ubah_kategori*, *c_kriteria*, dan *m_kriteria*. Proses dari aliran *sequence diagram* mengelola data kategori informasi peta adalah ketika aktor memilih menu data kriteria lalu menekan tombol “Batasan Kriteria” pada salah satu

baris data kriteria dan selanjutnya menekan tombol “Kategori Peta” maka sistem menjalankan fungsi *lihat_nilai_kriteria()* dengan parameter *id_kriteria* dan *id_bentuk_kriteria* yang ada di *class c_kriteria*. Di dalam *lihat_nilai_kriteria()*, sistem akan memanggil fungsi *ambil_kriteria()* dan *data_kategori_library()* yang mempunyai parameter sama yaitu *id_kriteria* pada *class m_kriteria*. Fungsi-fungsi pada *class m_kriteria* tersebut akan mengakses data kategori informasi peta dari *database* sehingga sistem dapat menampilkan halaman data kategori informasi peta. Aktor dapat melihat, menambah, mengubah dan menghapus data kategori informasi peta. *Sequence* diagram mengelola data kategori informasi peta oleh admin dijelaskan pada Lampiran C (*Sequence Diagram*).

13. *Sequence* Diagram Mengelola Nilai Kategori Informasi Peta

Penggambaran *sequence* diagram mengelola nilai kategori informasi peta digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence* diagram mengelola nilai kategori informasi peta adalah *view_home_admin*, *view_lihat_kriteria*, *view_lihat_range_kriteria*, *view_lihat_kategori*, *view_lihat_nilai_kriteria_peta*, *view_tambah_nilai_kriteria_peta*, *view_ubah_nilai_kriteria_peta*, *view_detail_nilai_kriteria_peta*, *c_kriteria*, *library_googlemaps*, *m_lokasi* dan *m_kriteria*. Proses dari aliran *sequence* diagram mengelola nilai kategori informasi peta adalah ketika aktor memilih menu data kriteria lalu menekan tombol “Batasan Kriteria” pada salah satu baris data kriteria lalu menekan tombol “Kategori Peta” dan selanjutnya menekan tombol “Nilai Kategori” pada salah satu baris data kategori informasi peta maka sistem menjalankan fungsi *lihat_nilai_kriteria_peta()* dengan parameter *id_kriteria* dan *id_kategori* yang ada di *class c_kriteria*. Di dalam fungsi *lihat_nilai_kriteria_peta()*, sistem akan memanggil beberapa fungsi pada *class m_kriteria*. Fungsi-fungsi pada *class m_kriteria* tersebut akan mengakses nilai kategori informasi peta dari *database* sehingga sistem dapat menampilkan halaman nilai kategori informasi peta. Aktor dapat melihat detail, menambah, mengubah dan menghapus nilai kategori informasi peta. *Sequence* diagram mengelola nilai

kategori informasi peta oleh admin dijelaskan pada Lampiran C (*Sequence Diagram*).

14. *Sequence Diagram* Mengelola Biodata

Penggambaran *sequence diagram* mengelola biodata digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence diagram* mengelola biodata adalah *view home_dm*, *view lihat_biodata*, *c_user* dan *m_user*. Proses dari aliran *sequence diagram* mengelola biodata adalah ketika aktor memilih menu biodata maka sistem menjalankan fungsi *biodata()* yang berparameter *id_user* yang ada di *class c_user*. Di dalam fungsi *biodata()*, sistem akan memanggil fungsi *ambil_detail_user()* yang berparameter *id_user* dan fungsi *combo_hak_akses()* pada *class m_user*. Fungsi-fungsi tersebut akan mengakses biodata *user* dari *database* sehingga sistem dapat menampilkan halaman biodata *user*. Aktor dapat melihat dan mengubah biodata miliknya. *Sequence diagram* mengelola biodata oleh pengambil keputusan dijelaskan pada Lampiran C (*Sequence Diagram*).

15. *Sequence Diagram* Melihat Data ATM

Penggambaran *sequence diagram* melihat data ATM digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence diagram* mengelola biodata adalah *view home_user*, *view lihat_atm*, *view detail_atm*, *c_kriteria* dan *m_kriteria*. Proses dari aliran *sequence diagram* melihat data ATM adalah ketika aktor memilih menu ATM BRI maka sistem menjalankan fungsi *lihat_atm()* yang ada di *class c_kriteria*. Di dalam fungsi *lihat_atm()*, sistem akan memanggil fungsi *ambil_kriteria()* dan fungsi *tabel_nilai_kriteria_peta()* pada *class m_kriteria*. Fungsi-fungsi tersebut akan mengakses data ATM BRI dari *database* sehingga sistem dapat menampilkan halaman data ATM BRI. Aktor dapat melihat detail dari ATM yang dipilih pada halaman tersebut. *Sequence diagram* melihat data ATM oleh nasabah dijelaskan pada Lampiran C (*Sequence Diagram*).

16. *Sequence* Diagram Mengubah Akun *User*

Penggambaran *sequence* diagram mengubah akun *user* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Class* yang berperan dalam *sequence* diagram mengelola biodata adalah *view home_admin*, *c_user*, *c_home*, *m_user*, *m_lokasi* dan *m_kriteria* . Proses dari aliran *sequence* diagram mengubah akun *user* adalah ketika aktor klik nama pengguna di bagian pojok kanan atas halaman maka sistem akan menampilkan pop up mengubah akun lalu aktor dapat mulai mengubah data akun miliknya. *Sequence* diagram mengubah akun *user* oleh admin dan pengambil keputusan dijelaskan pada Lampiran C (*Sequence* Diagram)..

4.3.6 *Class* Diagram

Class diagram menggambarkan hubungan antara kelas yang digunakan untuk membangun suatu sistem. Dalam paradigma OOP terdapat 3 jenis kelas yaitu model, view dan controller. Berdasarkan *sequence* diagram yang telah dibangun, *class* diagram *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM terdiri dari 5 kelas model dan 6 kelas controller. *Class* diagram sistem dapat dilihat pada Gambar 4.7.

4.3.7 *Entity Relationship* Diagram

Entity Relationship Diagram (ERD) *Group Decision Support System* (GDSS) penentuan lokasi penempatan ATM merupakan gambaran komponen dan struktur *database* yang digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini terdiri dari 19 entitas sebagaimana dapat dilihat pada Gambar 4.8.

4.4 Penulisan Kode Program

Desain yang telah dibuat akan diimplementasikan ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

1. Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework Code Igniter (CI)*.
2. Manajemen basisdata menggunakan *DBMS PostgreSQL*.

Kode program perhitungan metode *Entropy*, *VIKOR* dan *Borda* terletak di *class c_dss* pada *package controllers*. Penulisan kode program perhitungan metode tersebut dapat dilihat pada Tabel 4.5, Tabel 4.6 dan Tabel 4.7.

Tabel 4.5 Kode Program function `proses1_tambah_bobot_awal()`

```

377. public function proses1_tambah_bobot_awal($id_transaksi, $id_user) {
378.     $no = $this->input->post('no', true);
379.     $simpan = 0;
380.     for ($a = 1; $a < $no; $a++) {
381.         $cek = $this->input->post('check' . $a, true);
382.         if ($cek != null) {
383.             $bobot_awal = $this->input->post('bobot' . $a, true);
384.             $simpan = $simpan + $bobot_awal;
385.         }
386.     }
387.     if ($simpan == 1) {
388.         for ($a = 1; $a < $no; $a++) {
389.             $cek = $this->input->post('check' . $a, true);
390.             if ($cek != null) {
391.                 $id_kriteria = $this->input->post('id_kriteria' . $a, true);
392.                 $bobot_awal = $this->input->post('bobot' . $a, true);
393.                 $query = $this->m_dss->tambah_bobot_awal($id_transaksi, $id_user, $id_kriteria, $bobot_awal);
394.             }
395.         }
396.         if ($query) {
397.             redirect("c_dss/metode_entropy/$id_transaksi/$id_user");
398.         } else {
399.             $this->session->set_userdata('operation', "gagal");
400.             $this->session->set_userdata('message', "Gagal Menambah Bobot Awal");
401.             redirect("c_dss/lihat_proses1/$id_transaksi/$id_user");
402.         }
403.     } else {
404.         $this->session->set_userdata('operation', "gagal");
405.         $this->session->set_userdata('message', "Jumlah bobot awal harus sama dengan 1");
406.         redirect("c_dss/lihat_proses1/$id_transaksi/$id_user");
407.     }
408. }

```

Kode program pada Tabel 4.5 merupakan kode program *function* `proses1_tambah_bobot_awal()` yang digunakan untuk menyimpan kriteria yang dipilih oleh pengambil keputusan dan menyimpan bobot awal dari masing-masing

kriteria tersebut ke dalam *database*. Tahapan dalam menyimpan kriteria dan bobot awal adalah menjumlahkan bobot awal dari kriteria-kriteria yang dipilih oleh pengambil keputusan ke dalam variabel \$simpan seperti pada baris program 378-386. Tahap kedua adalah memeriksa apakah jumlah dari bobot awal adalah 1 atau tidak. Hal tersebut perlu dilakukan karena syarat dari perhitungan metode *Entropy* adalah jumlah bobot awal sama dengan 1. Jika jumlah bobot awal sama dengan 1 maka sistem akan menjalankan *function* tambah_bobot_awal() yang ada di *class* m_dss yang berada pada *package models*. Sistem akan menyimpan data kriteria dan bobot awal kriteria ketika menjalankan *function* tambah_bobot_awal() tersebut. Kode program untuk proses pemeriksaan jumlah bobot awal adalah pada baris 387-395. Tahap selanjutnya adalah memeriksa apakah kriteria dan bobot awal sudah tersimpan di *database* atau belum yaitu pada baris program 396-403. Apabila kriteria dan bobot awal sudah tersimpan, maka sistem menampilkan pesan berhasil dan apabila tidak tersimpan, maka sistem menampilkan pesan gagal. Pada tahap kedua yaitu pemeriksaan jumlah bobot awal, apabila jumlah bobot awal tidak sama dengan 1 maka sistem akan menjalankan baris program 404-408 yaitu menampilkan pesan kalau bobot awal harus sama dengan 1.

Penyimpanan kriteria dan bobot awal ke *database* merupakan langkah awal agar dapat melakukan perhitungan metode *Entropy*. Kode program untuk perhitungan metode *Entropy* terletak pada *function* metode_entropy() yang dapat dilihat pada Tabel 4.6.

Tabel 4.6 Kode Program *function* metode_entropy()

411.	public function metode_entropy(\$id_transaksi, \$id_user) {
412.	\$metrik = array();
413.	\$nilai = array();
414.	\$temp = array();
415.	\$nilaimax = array();
416.	\$nilaimin = array();
417.	\$normalisasi = array();
418.	\$jum = array();
419.	\$jumlenn = array();
420.	\$jumentropy = 0;
421.	\$jumlamda = 0;
422.	\$probabilitas = array();
423.	\$lenprobabilitas = array();
424.	\$metrikenentropy = array();
425.	\$lamda = array();
426.	\$lamdabobot = array();
427.	\$entropy = array();

```

428. $ambildata = $this->m_dss->metrik($id_transaksi, $id_user);
429. $kelurahan = $this->m_dss->ambil_kelurahan($id_transaksi, $id_user);
430. $kriteria = $this->m_dss->ambil_kriteria($id_transaksi, $id_user);
431. $jumkriteria = count($kriteria);
432. $jumdesa = count($kelurahan);
433. $b = 0;
434. foreach ($kriteria as $row) {
435.     $bobotawal[$b] = $row['bobot_awal'];
436.     $sifat[$b] = $row['id_sifat_kriteria'];
437.     $id_krit[$b] = $row['id_kriteria'];
438.     $b++;
439. }
440. //=====menyimpan nilai kepentingan ke metrik 2D=====
441. $skel = "";
442. $des = -1;
443. $skrit = 0;
444. foreach ($ambildata as $row) {
445.     if ($skel != $row['nama_kelurahan']) {
446.         $skrit = 0;
447.         $des++;
448.         $skel = $row['nama_kelurahan'];
449.         $nilai[$des][$skrit] = $row['nilai_kepentingan'];
450.         $skrit++;
451.     } else {
452.         $nilai[$des][$skrit] = $row['nilai_kepentingan'];
453.         $skrit++;
454.     }
455. }
456. //=====1. MEMBUAT METRIK KRITERIA=====
457. for ($j = 0; $j < $jumkriteria; $j++) {
458.     for ($i = 0; $i < $jumdesa; $i++) {
459.         $metrik[$i][$j] = $nilai[$i][$j];
460.     }
461. }
462. //mencari nilai maksimal per kriteria
463. for ($j = 0; $j < $jumkriteria; $j++) {
464.     for ($i = 0; $i < $jumdesa; $i++) {
465.         $temp[$i] = $metrik[$i][$j];
466.     }
467.     $nilaimax [$j] = max($temp);
468.     unset($temp);
469. }
470. //mencari nilai minimal per kriteria
471. for ($j = 0; $j < $jumkriteria; $j++) {
472.     for ($i = 0; $i < $jumdesa; $i++) {
473.         $temp[$i] = $metrik[$i][$j];
474.     }
475.     $nilaimin [$j] = min($temp);
476.     unset($temp);
477. }
478. //=====2. MEMBUAT METRIK TERNORMALISASI=====
479. for ($j = 0; $j < $jumkriteria; $j++) {
480.     for ($i = 0; $i < $jumdesa; $i++) {
481.         if ($sifat[$j] == '1') {
482.             $normalisasi[$i][$j] = $metrik[$i][$j] / $nilaimax [$j]; //benefit
483.         } else {
484.             $normalisasi[$i][$j] = $nilaimin [$j] / $metrik[$i][$j]; //cost
485.         }
486.     }
487. }
488. //=====METODE ENTROPY=====
489. //mencari jumlah nilai ternormalisasi per kriteria

```

```

490.     for ($j = 0; $j < $jumkriteria; $j++) {
491.         $simpan = 0;
492.         for ($i = 0; $i < $jumdesa; $i++) {
493.             $simpan = $simpan + $normalisasi[$i][$j];
494.         }
495.         $jum[$j] = $simpan;
496.     }
497. //=====3. MEMBUAT METRIK PROBABILITAS=====
498.     for ($j = 0; $j < $jumkriteria; $j++) {
499.         for ($i = 0; $i < $jumdesa; $i++) {
500.             $probabilitas[$i][$j] = $normalisasi[$i][$j] / $jum[$j];
501.         }
502.     }
503. //nilai probabilitas * LN(nilai probabilitas)
504.     for ($j = 0; $j < $jumkriteria; $j++) {
505.         for ($i = 0; $i < $jumdesa; $i++) {
506.             $lenprobabilitas[$i][$j] = $probabilitas[$i][$j] * log(($probabilitas[$i][$j]), M_E);
507.         }
508.     }
509. //jumlah len probabilitas per kriteria
510.     for ($j = 0; $j < $jumkriteria; $j++) {
511.         $simpan = 0;
512.         for ($i = 0; $i < $jumdesa; $i++) {
513.             $simpan = $simpan + $lenprobabilitas[$i][$j];
514.         }
515.         $jumlen[$j] = $simpan;
516.     }
517. //=====METRIK ENTROPY PER KRITERIA=====
518.     for ($j = 0; $j < $jumkriteria; $j++) {
519.         $metrikentropy[$j] = ((-1) / log($jumdesa, M_E)) * $jumlen[$j];
520.     }
521. //=====LAMDA ENTROPY PER KRITERIA=====
522. //jumlah nilai entropy
523.     for ($j = 0; $j < $jumkriteria; $j++) {
524.         $jumentropy = $jumentropy + $metrikentropy[$j];
525.     }
526. //hitung lamda
527.     for ($j = 0; $j < $jumkriteria; $j++) {
528.         if(($jumkriteria - $jumentropy)==0){
529.             $lamda[$j]=0;
530.         }
531.         else{
532.             $lamda[$j] = (1 / ($jumkriteria - $jumentropy)) * (1 - $metrikentropy[$j]);
533.         }
534. //=====LAMDA KALI BOBOT AWAL PER KRITERIA=====
535.     for ($j = 0; $j < $jumkriteria; $j++) {
536.         $lamdabobot[$j] = $lamda[$j] * $bobotawal[$j];
537.     }
538. //=====JUMLAH LAMDA KALI BOBOT=====
539.     for ($j = 0; $j < $jumkriteria; $j++) {
540.         $jumlamda = $jumlamda + $lamdabobot[$j];
541.     }
542. //=====HASIL BOBOT ENTROPY PER KRITERIA=====
543.     for ($j = 0; $j < $jumkriteria; $j++) {
544.         if ($lamdabobot[$j] == 0) {
545.             $entropy[$j] = 0;
546.         } else {
547.             $entropy[$j] = $lamdabobot[$j] / $jumlamda;
548.         }
549.     }
550. //SIMPAN ENTROPY
551.     for ($j = 0; $j < $jumkriteria; $j++) {

```

552.	\$bobot_entropy = \$entropy[\$j];
553.	\$id_kriteria = \$id_krit[\$j];
554.	\$query = \$this->m_dss->simpan_bobot_entropy(\$id_transaksi, \$id_user, \$id_kriteria, \$bobot_entropy);
555.	}
556.	if (\$query) {
557.	\$data1['normalisasi'] = \$normalisasi;
558.	\$data1['probabilitas'] = \$probabilitas;
559.	\$data1['lenprobabilitas'] = \$lenprobabilitas;
560.	\$data1['metrikentropy'] = \$metrikentropy;
561.	\$data1['lamda'] = \$lamda;
562.	\$data1['lamdabobot'] = \$lamdabobot;
563.	\$data1['entropy'] = \$entropy;
564.	\$data1['jumlah'] = \$jumlah;
565.	\$data1['kriteria'] = \$this->m_dss->ambil_kriteria(\$id_transaksi, \$id_user);
566.	\$data1['metrik'] = \$this->m_dss->metrik(\$id_transaksi, \$id_user);
567.	\$data1['kelurahan'] = \$this->m_dss->ambil_kelurahan(\$id_transaksi, \$id_user);
568.	\$data1['detail'] = \$this->m_dss->ambil_detail_transaksi(\$id_transaksi, \$id_user);
569.	\$data = \$this->session();
570.	\$data['content'] = \$this->load->view('dm/metode_entropy', \$data1, true);
571.	\$this->load->view('layout/template', \$data);
572.	} else {
573.	redirect("c_dss/proses1_tambah_bobot_awal/\$id_transaksi/\$id_user");
574.	}
575.	}

Kode program pada Tabel 4.6 merupakan kode program *function* *metode_entropy()* yang digunakan untuk melakukan perhitungan bobot kriteria menggunakan metode *entropy*. Tahapan pertama dari perhitungan *entropy* adalah mendeklarasikan variabel-variabel yang digunakan dan mengambil data dari *database* dengan memanggil *function* *metrik()*, *ambil_kriteria()* dan *ambil_kelurahan()* serta menyimpan data dan jumlah data tersebut ke dalam variabel-variabel yang dapat dilihat pada baris program 412-439. Langkah selanjutnya adalah menyimpan nilai kepentingan dari setiap kelurahan berdasarkan kriteria yang terpilih ke dalam metrik 2 dimensi menggunakan kode program baris ke 441 sampai 455. Langkah selanjutnya yaitu membuat metrik kriteria pada baris program 457-461.

Pembuatan metrik kriteria ternormalisasi adalah tahap berikutnya dalam perhitungan *Entropy*. Langkah awal sebelum dapat membuat metrik kriteria ternormalisasi adalah dengan mencari nilai maksimal dan minimal dari nilai kepentingan setiap kriteria yaitu pada baris program 463-477. Pembuatan metrik kriteria ternormalisasi terletak pada baris program 479-487. Langkah selanjutnya pada perhitungan metode *Entropy* adalah pembuatan metrik probabilitas kriteria pada baris program 489-516. Setelah pembuatan metrik probabilitas kriteria selesai

dilakukan, sistem menghitung nilai *entropy* setiap kriteria yaitu pada baris program 518-520. Tahap selanjutnya adalah perhitungan lamda kriteria yang akan digunakan untuk menghitung bobot *entropy* yaitu pada baris program 523-533. Tahap terakhir adalah perhitungan bobot akhir *entropy* yaitu pada baris program 535-549. Hasil perhitungan bobot *entropy* tersebut kemudian disimpan ke dalam *database* untuk digunakan pada perhitungan metode VIKOR. Kode program penyimpanan bobot *entropy* ke dalam *database* dapat dilihat pada baris program 551-555. Hasil dari perhitungan metode *Entropy* mulai dari pembuatan metrik kriteria sampai mendapatkan nilai akhir bobot *entropy* ditampilkan ke dalam tampilan sistem agar dapat dilihat oleh pengambil keputusan menggunakan kode program pada baris ke 556 sampai 574.

Bobot *entropy* yang sudah tersimpan dalam *database* akan digunakan dalam perhitungan metode VIKOR. Kode program untuk perhitungan metode VIKOR terletak pada *function* metode_vikor() yang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Kode Program *function* metode_vikor()

```

577. public function metode_vikor($id_transaksi, $id_user) {
578.     $metrik = array();
579.     $nilai = array();
580.     $temp = array();
581.     $nilaimax = array();
582.     $nilaimin = array();
583.     $normalisasi = array();
584.     $idealpositif = array();
585.     $idealnegasif = array();
586.     $idealtebobot = array();
587.     $solusiteringgi = array();
588.     $solusiterendah = array();
589.     $rasio = array();
590.     $ambildata = $this->m_dss->metrik($id_transaksi, $id_user);
591.     $kelurahan = $this->m_dss->ambil_kelurahan($id_transaksi, $id_user);
592.     $kriteria = $this->m_dss->ambil_kriteria($id_transaksi, $id_user);
593.     $jumkriteria = count($kriteria);
594.     $jumdesa = count($kelurahan);
595.     //=====menyimpan nilai kepentingan ke metrik 2D=====
596.     $kel = "";
597.     $des = -1;
598.     $krit = 0;
599.     foreach ($ambildata as $row) {
600.         if ($kel != $row['nama_kelurahan']) {
601.             $krit = 0;
602.             $des++;
603.             $kel = $row['nama_kelurahan'];
604.             $nilai[$des][$krit] = $row['nilai_kepentingan'];
605.             $krit++;
606.         } else {
607.             $nilai[$des][$krit] = $row['nilai_kepentingan'];

```

```

608.         $krit++;
609.     }
610. }
611. //=====1. MEMBUAT METRIK KRITERIA=====
612. for ($j = 0; $j < $jumkriteria; $j++) {
613.     for ($i = 0; $i < $jumdesa; $i++) {
614.         $metrik[$i][$j] = $nilai[$i][$j];
615.     }
616. }
617. //mencari nilai maksimal per kriteria
618. for ($j = 0; $j < $jumkriteria; $j++) {
619.     for ($i = 0; $i < $jumdesa; $i++) {
620.         $temp[$i] = $metrik[$i][$j];
621.     }
622.     $nilaimax [$j] = max($temp);
623.     unset($temp);
624. }
625. //mencari nilai minimal per kriteria
626. for ($j = 0; $j < $jumkriteria; $j++) {
627.     for ($i = 0; $i < $jumdesa; $i++) {
628.         $temp[$i] = $metrik[$i][$j];
629.     }
630.     $nilaimin [$j] = min($temp);
631.     unset($temp);
632. }
633. //=====2. MEMBUAT METRIK TERNORMALISASI=====
634. $b = 0;
635. foreach ($skriteria as $row) {
636.     $entropy[$b] = $row['bobot_entropy'];
637.     $sifat[$b] = $row['id_sifat_kriteria'];
638.     $b++;
639. }
640. for ($j = 0; $j < $jumkriteria; $j++) {
641.     for ($i = 0; $i < $jumdesa; $i++) {
642.         if ($sifat[$i] == '1') {
643.             $normalisasi[$i][$j] = $metrik[$i][$j] / $nilaimax [$j];
644.         } else {
645.             $normalisasi[$i][$j] = $nilaimin [$j] / $metrik[$i][$j];
646.         }
647.     }
648. }
649. //=====METODE VIKOR=====
650. //1.mencari nilai ideal positif dan negatif setiap kriteria dari metrik normalisasi
651. //ideal positif
652. for ($j = 0; $j < $jumkriteria; $j++) {
653.     for ($i = 0; $i < $jumdesa; $i++) {
654.         $temp[$i] = $normalisasi[$i][$j];
655.     }
656.     $idealpositif [$j] = max($temp);
657.     unset($temp);
658. }
659. //ideal negatif
660. for ($j = 0; $j < $jumkriteria; $j++) {
661.     for ($i = 0; $i < $jumdesa; $i++) {
662.         $temp[$i] = $normalisasi[$i][$j];
663.     }
664.     $idealnegasif [$j] = min($temp);
665.     unset($temp);
666. }
667. //2.mencari nilai solusi tertinggi dan terendah setiap alternatif
668. //nilai solusi ideal terbobot entropy
669. $bagi = array();

```

```

670. for ($j = 0; $j < $jumkriteria; $j++) {
671.     for ($i = 0; $i < $jumdesa; $i++) {
672.         $bagi[$j] = ($idealpositif [$j] - $idealnegasif [$j]);
673.         if ($bagi[$j] == '0') {
674.             $idealterbobot[$i][$j] = 0;
675.         } else {
676.             $idealterbobot[$i][$j] = $entropy[$j] * (($idealpositif [$j] - $normalisasi[$i][$j]) / $bagi[$j]);
677.         }
678.     }
679. }
680. //S = Nilai solusi tertinggi, jumlah ideal terbobot
681. for ($i = 0; $i < $jumdesa; $i++) {
682.     $simpan = 0;
683.     for ($j = 0; $j < $jumkriteria; $j++) {
684.         $simpan = $simpan + $idealterbobot[$i][$j];
685.     }
686.     $solusiteringgi[$i] = $simpan;
687. }
688. //R = Nilai solusi terendah, max ideal terbobot
689. for ($i = 0; $i < $jumdesa; $i++) {
690.     for ($j = 0; $j < $jumkriteria; $j++) {
691.         $temp[$j] = $idealterbobot[$i][$j];
692.     }
693.     $solusiterendah [$i] = max($temp);
694.     unset($temp);
695. }
696. //3. Menghitung Nilai Rasio (Q) setiap alternatif
697. //menghitung nilai Smax
698. for ($i = 0; $i < $jumdesa; $i++) {
699.     $temp[$i] = $solusiteringgi[$i];
700. }
701. $smax = max($temp);
702. unset($temp);
703. //menghitung nilai Smin
704. for ($i = 0; $i < $jumdesa; $i++) {
705.     $temp[$i] = $solusiteringgi[$i];
706. }
707. $smin = min($temp);
708. unset($temp);
709. //menghitung nilai Rmax
710. for ($i = 0; $i < $jumdesa; $i++) {
711.     $temp[$i] = $solusiterendah[$i];
712. }
713. $rmax = max($temp);
714. unset($temp);
715. //menghitung nilai Rmin
716. for ($i = 0; $i < $jumdesa; $i++) {
717.     $temp[$i] = $solusiterendah[$i];
718. }
719. $rmin = min($temp);
720. unset($temp);
721.
722. $kurangs = $smax - $smin;
723. $kurangr = $rmax - $rmin;
724. //menghitung nilai Q per alternatif
725. $v = 0.5;
726. for ($i = 0; $i < $jumdesa; $i++) {
727.     if ($kurangs == '0') {
728.         $hasil1 = 0;
729.     } else {
730.         $hasil1 = ($v * (($solusiteringgi[$i] - $smin) / ($kurangs)));
731.     }

```

```

732.     if ($kurangr == '0') {
733.         $hasil2 = 0;
734.     } else {
735.         $hasil2 = ((1 - $v) * (($solusiterendah[$i] - $rmin) / ($kurangr)));
736.     }
737.     $rasio[$i] = $hasil1 + $hasil2;
738. }
739. //4.Mengurutkan Nilai S,R,Q setiap alternatif decreasing order
740. $a = 0;
741. foreach ($kelurahan as $row) {
742.     $desa[$a] = $row['nama_kelurahan'];
743.     $a++;
744. }
745.
746. //mengurutkan Nilai S
747. $nilaiq = array();
748. $desaq = array();
749. $nilais = array();
750. $desas = array();
751. $nilair = array();
752. $desar = array();
753. $surut = array();
754. for ($i = 0; $i < $jumdesa; $i++) {
755.     $surut [$i] = array(
756.         'alternatif' => $desa[$i],
757.         'ns' => $solusitertinggi[$i]
758.     );
759. }
760. $b = array();
761. $c = array();
762. foreach ($surut as $key => $value) {
763.     $b[$key] = $value['ns'];
764. }
765. natsort($b);
766. foreach ($b as $key => $value) {
767.     $c[] = $surut[$key];
768. }
769. foreach ($c as $row) {
770.     $nilais[] = $row['ns'];
771.     $desas[] = $row['alternatif'];
772. }
773. for ($i = 0; $i < $jumdesa; $i++) {
774.     $ranks [$i] = array(
775.         'nilais' => $nilais[$i],
776.         'desas' => $desas[$i]
777.     );
778. }
779.
780. //mengurutkan Nilai R
781. $surutr = array();
782. for ($i = 0; $i < $jumdesa; $i++) {
783.     $surutr [$i] = array(
784.         'alternatif' => $desa[$i],
785.         'nr' => $solusiterendah[$i]
786.     );
787. }
788. $br = array();
789. $cr = array();
790. foreach ($surutr as $key => $value) {
791.     $br[$key] = $value['nr'];
792. }
793. natsort($br);

```

```

794.     foreach ($br as $key => $value) {
795.         $scr[] = $urutr[$key];
796.     }
797.     foreach ($scr as $row) {
798.         $nilair[] = $row['nr'];
799.         $desar[] = $row['alternatif'];
800.     }
801.     for ($i = 0; $i < $jumdesa; $i++) {
802.         $rankr [$i] = array(
803.             'nilair' => $nilair[$i],
804.             'desar' => $desar[$i]
805.         );
806.     }
807.
808.     //mengurutkan Nilai Q
809.     $urutq = array();
810.     for ($i = 0; $i < $jumdesa; $i++) {
811.         $urutq [$i] = array(
812.             'alternatif' => $desa[$i],
813.             'nq' => $rasio[$i]
814.         );
815.     }
816.     $bq = array();
817.     $cq = array();
818.     $rankq = array();
819.     foreach ($urutq as $key => $value) {
820.         $bq[$key] = $value['nq'];
821.     }
822.     natsort($bq);
823.     foreach ($bq as $key => $value) {
824.         $cq[] = $urutq[$key];
825.     }
826.     foreach ($cq as $row) {
827.         $nilaiq[] = $row['nq'];
828.         $desaq[] = $row['alternatif'];
829.     }
830.     for ($i = 0; $i < $jumdesa; $i++) {
831.         $rankq [$i] = array(
832.             'nilaiq' => $nilaiq[$i],
833.             'desaq' => $desaq[$i]
834.         );
835.     }
836.
837.     //5.CEK KONDISI 1 DAN 2
838.     //- Hitung DQ
839.     $dq = 1 / ($jumdesa - 1);
840.     //KONDISI 1
841.     $hasil = array();
842.     if ($nilaiq[1] - $nilaiq[0] > $dq || $nilaiq[1] - $nilaiq[0] == $dq) {
843.         //KONDISI 2
844.         if ($desaq[0] == $desar[0] && $desaq[0] == $desas[0]) {
845.             //nilainya yg ranking 1 tetep
846.             $d = 0;
847.             $bantu = 0;
848.             for ($i = 0; $i < $jumdesa; $i++) {
849.                 $d++;
850.                 if ($i == 0) {
851.                     $hasil [$i] = array(
852.                         'nilaiq' => $nilaiq[$i],
853.                         'desaq' => $desaq[$i],
854.                         'rankingq' => $d
855.                     );

```

```

856.     }
857.     if ($i > 0) {
858.         if ($nilaiq[$i] == $nilaiq[$i - 1]) {
859.             $d--;
860.             $bantu++;
861.             $hasil [$i] = array(
862.                 'nilaiq' => $nilaiq[$i],
863.                 'desaq' => $desaq[$i],
864.                 'rankingq' => $d
865.             );
866.         } else {
867.             $d = $d + $bantu;
868.             $bantu = 0;
869.             $hasil [$i] = array(
870.                 'nilaiq' => $nilaiq[$i],
871.                 'desaq' => $desaq[$i],
872.                 'rankingq' => $d
873.             );
874.         }
875.     }
876. }
877. } else {
878. //nilai ke 1 dan 2 jadi ranking 1
879. $m = 0;
880. $bantu1 = 0;
881. for ($i = 0; $i < $jumdesa; $i++) {
882.     if ($i == 0 || $i == 1) {
883.         $hasil [$i] = array(
884.             'nilaiq' => $nilaiq[$i],
885.             'desaq' => $desaq[$i],
886.             'rankingq' => 1
887.         );
888.     }
889.     $m++;
890.     if ($i > 1) {
891.         if ($nilaiq[$i] == $nilaiq[$i - 1]) {
892.             $m--;
893.             $bantu1++;
894.             $hasil [$i] = array(
895.                 'nilaiq' => $nilaiq[$i],
896.                 'desaq' => $desaq[$i],
897.                 'rankingq' => $m
898.             );
899.         } else {
900.             $m = $m + $bantu1;
901.             $bantu1 = 0;
902.             $hasil [$i] = array(
903.                 'nilaiq' => $nilaiq[$i],
904.                 'desaq' => $desaq[$i],
905.                 'rankingq' => $m
906.             );
907.         }
908.     }
909. }
910. }
911. } else {
912. $o = 0;
913. $g = 1;
914. $ban = 0;
915. for ($i = 0; $i < $jumdesa; $i++) {
916.     if ($nilaiq[$i] - $nilaiq[0] < $dq || $nilaiq[$i] - $nilaiq[0] == $dq) {
917.         $hasil [$i] = array(

```

```

918.         'nilaiq' => $nilaiq[$i],
919.         'desaq' => $desaq[$i],
920.         'rankingq' => $g
921.     );
922.     $ban++;
923. } else {
924.     $o++;
925.     if ($i > 0) {
926.         if ($nilaiq[$i] == $nilaiq[$i - 1]) {
927.             $o--;
928.             $ban++;
929.             $hasil [$i] = array(
930.                 'nilaiq' => $nilaiq[$i],
931.                 'desaq' => $desaq[$i],
932.                 'rankingq' => $o
933.             );
934.         } else {
935.             $o = $o + $ban;
936.             $ban = 0;
937.             $hasil [$i] = array(
938.                 'nilaiq' => $nilaiq[$i],
939.                 'desaq' => $desaq[$i],
940.                 'rankingq' => $o
941.             );
942.         }
943.     }
944. }
945.
946. }
947. }
948.
949. $k = 0;
950. foreach ($hasil as $row) {
951.     $rankingfix[$k] = $row['rankingq'];
952.     $desafix[$k] = $row['desaq'];
953.     $lurah = $this->m_dss->cek_kelurahan($desafix[$k]);
954.     if ($lurah) {
955.         $id[$k] = $lurah->id_kelurahan;
956.     }
957.     $k++;
958. }
959. for ($i = 0; $i < $jumdesa; $i++) {
960.     $idkel = $id[$i];
961.     $ranking = $rankingfix[$i];
962.     $query = $this->m_dss->simpan_vikor($id_transaksi, $id_user, $idkel, $ranking);
963. }
964. $query1 = $this->m_dss->ubah_status_dm($id_transaksi, $id_user);
965. if ($query && $query1) {
966.     $data1['normalisasi'] = $normalisasi;
967.     $data1['idealpositif'] = $idealpositif;
968.     $data1['idealnegasif'] = $idealnegasif;
969.     $data1['idealtebobot'] = $idealtebobot;
970.     $data1['solusiteringgi'] = $solusiteringgi;
971.     $data1['solusiterendah'] = $solusiterendah;
972.     $data1['rasio'] = $rasio;
973.     $data1['rankq'] = $rankq;
974.     $data1['ranks'] = $ranks;
975.     $data1['rankr'] = $rankr;
976.     $data1['hasil'] = $hasil;
977.     $data1['kriteria'] = $this->m_dss->ambil_kriteria($id_transaksi, $id_user);
978.     $data1['metrik'] = $this->m_dss->metrik($id_transaksi, $id_user);
979.     $data1['kelurahan'] = $this->m_dss->ambil_kelurahan($id_transaksi, $id_user);

```

980.	\$data1['detail'] = \$this->m_dss->ambil_detail_transaksi(\$id_transaksi, \$id_user);
981.	\$data = \$this->session();
982.	\$data['content'] = \$this->load->view('dm/metode_vikor', \$data1, true);
983.	\$this->load->view('layout/template', \$data);
984.	} else {
985.	redirect("c_dss/metode_entropy/\$id_transaksi/\$id_user");
986.	}
987.	}

Kode program pada Tabel 4.7 merupakan kode program *function* *metode_vikor()* yang digunakan untuk melakukan perhitungan perankingan lokasi menggunakan metode VIKOR. Tahapan pertama dari perhitungan VIKOR adalah mendeklarasikan variabel-variabel yang digunakan dan mengambil data dari *database* dengan memanggil *function* *metrik()*, *ambil_kriteria()* dan *ambil_kelurahan()* serta menyimpan data dan jumlah data tersebut ke dalam variabel-variabel yang dapat dilihat pada baris program 578-594. Langkah selanjutnya adalah menyimpan nilai kepentingan dari setiap kelurahan menggunakan kode program baris ke 596 sampai 610. Langkah selanjutnya yaitu membuat metrik kriteria pada baris program 612-616.

Pembuatan metrik kriteria ternormalisasi adalah tahap berikutnya dalam perhitungan VIKOR yaitu pada baris program 618-648. Tahap selanjutnya adalah menghitung nilai solusi ideal positif dan ideal negatif setiap kriteria dari metrik ternormalisasi yaitu pada baris program 652-666. Sistem kemudian menghitung nilai solusi tertinggi (S) dan solusi terendah (R) setiap alternatif/kelurahan dengan menggunakan baris program 669-695. Langkah selanjutnya adalah menghitung nilai rasio/indeks VIKOR (Q) untuk setiap alternatif yaitu pada baris program 698-738. Setelah mendapatkan nilai S,R,Q maka tahap selanjutnya adalah mengurutkan nilai S,R dan Q dengan *decreasing order*. Nilai S,R dan Q dengan nilai paling kecil merupakan nilai dengan ranking pertama. Kode program untuk mengurutkan nilai S,R dan Q terdapat pada baris program 740-835. Langkah selanjutnya adalah memeriksa beberapa kondisi dari metode VIKOR untuk menentukan ranking yang tepat pada perhitungan S,R dan Q. Ada 2 kondisi yang harus diperiksa dengan menggunakan baris program 839-947. Hasil dari pemeriksaan kondisi tersebut adalah hasil perankingan kelurahan yang merupakan rekomendasi lokasi penempatan ATM.

Hasil perankingan VIKOR tdisimpan ke dalam *database* yaitu pada baris ke 949 sampai 963 agar dapat dipakai dalam perhitungan keputusan kelompok menggunakan metode Borda. Setelah hasil metode VIKOR disimpan ke dalam *database*, maka sistem mengubah status pengambil keputusan menjadi sudah melakukan analisa dengan menggunakan kode program baris ke 964. Langkah selanjutnya adalah menampilkan hasil dari perhitungan VIKOR ke dalam tampilan pengambil keputusan mulai dari metrik kriteria ternormalisasi sampai hasil perankingan lokasi menggunakan kode program pada baris ke 965 sampai 987.

Hasil perankingan lokasi yang sudah tersimpan dalam *database* akan digunakan dalam perhitungan keputusan kelompok menggunakan metode Borda. Kode program untuk perhitungan metode Borda terletak pada *function* `metode_borda()` yang dapat dilihat pada Tabel 4.8.

Tabel 4.8 Kode Program *function* `metode_borda()`

989.	<code>public function metode_borda(\$id_transaksi) {</code>
990.	<code> \$metrik = \$this->m_dss->ambil_vikor(\$id_transaksi);</code>
991.	<code> \$kelurahan = \$this->m_dss->ambil_vikor_kelurahan(\$id_transaksi);</code>
992.	<code> \$dm = \$this->m_dss->ambil_vikor_user(\$id_transaksi);</code>
993.	<code> \$jumuser = count(\$dm);</code>
994.	<code> \$jumdesa = count(\$kelurahan);</code>
995.	<code> \$awal = array();</code>
996.	<code> \$nilai = array();</code>
997.	<code> //matrik ranking sebelum borda</code>
998.	<code> \$kel = "";</code>
999.	<code> \$des = -1;</code>
1000.	<code> \$user = 0;</code>
1001.	<code> foreach (\$metrik as \$row) {</code>
1002.	<code> if (\$kel != \$row['nama_kelurahan']) {</code>
1003.	<code> \$user = 0;</code>
1004.	<code> \$des++;</code>
1005.	<code> \$kel = \$row['nama_kelurahan'];</code>
1006.	<code> \$nilai[\$des][\$user] = \$row['ranking'];</code>
1007.	<code> \$user++;</code>
1008.	<code> } else {</code>
1009.	<code> \$nilai[\$des][\$user] = \$row['ranking'];</code>
1010.	<code> \$user++;</code>
1011.	<code> }</code>
1012.	<code> }</code>
1013.	<code> //1.METRIK RANKING 2 D</code>
1014.	<code> for (\$j = 0; \$j < \$jumuser; \$j++) {</code>
1015.	<code> for (\$i = 0; \$i < \$jumdesa; \$i++) {</code>
1016.	<code> \$awal[\$i][\$j] = \$nilai[\$i][\$j];</code>
1017.	<code> }</code>
1018.	<code> }</code>
1019.	<code> //2. MASUKKAN POINT KE RANKING</code>
1020.	<code> \$x = 1;</code>
1021.	<code> \$y = \$jumdesa;</code>
1022.	<code> \$z = 0;</code>
1023.	<code> \$hasilpoint = array();</code>

```

1024. for ($j = 0; $j < $jumuser; $j++) {
1025.     for ($i = 0; $i < $jumdesa; $i++) {
1026.         if ($awal[$i][$j] == $y) {
1027.             $hasilpoint[$i][$j] = $x;
1028.         } else {
1029.             for ($z = 1; $z < $jumdesa; $z++) {
1030.                 if ($awal[$i][$j] == ($y - $z)) {
1031.                     $hasilpoint[$i][$j] = $x + $z;
1032.                 }
1033.             }
1034.         }
1035.     }
1036. }
1037. //3. JUMLAHKAN POINT PER KELURAHAN
1038. //jumlah point
1039. $jumpoint = array();
1040. for ($i = 0; $i < $jumdesa; $i++) {
1041.     $simpan = 0;
1042.     for ($j = 0; $j < $jumuser; $j++) {
1043.         $simpan = $simpan + $hasilpoint[$i][$j];
1044.     }
1045.     $jumpoint[$i] = $simpan;
1046. }
1047. //4. URUTKAN DARI BESAR KE KECIL
1048. $a = 0;
1049. foreach ($kelurahan as $row) {
1050.     $desa[$a] = $row['nama_kelurahan'];
1051.     $a++;
1052. }
1053. $nilaiborda = array();
1054. $desaborda = array();
1055. $surut = array();
1056. for ($i = 0; $i < $jumdesa; $i++) {
1057.     $surut[$i] = array(
1058.         'alternatif' => $desa[$i],
1059.         'point' => $jumpoint[$i]
1060.     );
1061. }
1062. $b = array();
1063. $c = array();
1064. foreach ($surut as $key => $value) {
1065.     $b[$key] = $value['point'];
1066. }
1067. arsort($b);
1068. foreach ($b as $key => $value) {
1069.     $c[] = $surut[$key];
1070. }
1071. foreach ($c as $row) {
1072.     $nilaiborda[] = $row['point'];
1073.     $desaborda[] = $row['alternatif'];
1074. }
1075. //BERIKAN RANKING, NILAI SAMA DAPAT RANKING SAMA
1076. $d = 0;
1077. $bantu = 0;
1078. for ($i = 0; $i < $jumdesa; $i++) {
1079.     $d++;
1080.     if ($i == 0) {
1081.         $hasilborda[$i] = array(
1082.             'ranking' => $nilaiborda[$i],
1083.             'desa' => $desaborda[$i],
1084.             'borda' => $d
1085.         );

```

```

1086.     }
1087.     if ($i > 0) {
1088.         if ($nilaiborda[$i] == $nilaiborda[$i - 1]) {
1089.             $d--;
1090.             $bantu++;
1091.             $hasilborda [$i] = array(
1092.                 'ranking' => $nilaiborda[$i],
1093.                 'desa' => $desaborda[$i],
1094.                 'borda' => $d
1095.             );
1096.         } else {
1097.             $d = $d + $bantu;
1098.             $bantu = 0;
1099.             $hasilborda [$i] = array(
1100.                 'ranking' => $nilaiborda[$i],
1101.                 'desa' => $desaborda[$i],
1102.                 'borda' => $d
1103.             );
1104.         }
1105.     }
1106. }
1107. //CEK UDAH ADA NILAI BORDA DI DATABASE
1108. $cekborda = $this->m_dss->cek_borda($id_transaksi);
1109. $k = 0;
1110. foreach ($hasilborda as $row) {
1111.     $rankingborda[$k] = $row['borda'];
1112.     $desaborda[$k] = $row['desa'];
1113.     $nborda[$k] = $row['ranking'];
1114.     $lurah = $this->m_dss->cek_kelurahan($desaborda[$k]);
1115.     if ($lurah) {
1116.         $sid[$k] = $lurah->id_kelurahan;
1117.     }
1118.     $k++;
1119. }
1120. if ($cekborda == null) {
1121.     for ($i = 0; $i < $jumdesa; $i++) {
1122.         $sidkel = $id[$i];
1123.         $rank = $rankingborda[$i];
1124.         $nilai_borda = $nborda[$i];
1125.         $query = $this->m_dss->simpan_borda($id_transaksi, $sidkel, $rank, $nilai_borda);
1126.     }
1127. } else {
1128.     for ($i = 0; $i < $jumdesa; $i++) {
1129.         $sidkel = $id[$i];
1130.         $rank = $rankingborda[$i];
1131.         $nilai_borda = $nborda[$i];
1132.         $query = $this->m_dss->ubah_borda($id_transaksi, $sidkel, $rank, $nilai_borda);
1133.     }
1134. }
1135. if ($query) {
1136.     $data1['borda'] = $hasilborda;
1137.     $data1['point'] = $hasilpoint;
1138.     $data1['jumpoint'] = $jumpoint;
1139.     $data1['user'] = $this->m_dss->ambil_vikor_user($id_transaksi);
1140.     $data1['metrik'] = $this->m_dss->ambil_vikor($id_transaksi);
1141.     $data1['kelurahan'] = $this->m_dss->ambil_vikor_kelurahan($id_transaksi);
1142.     $data = $this->session();
1143.     $data['content'] = $this->load->view('dm/metode_borda', $data1, true);
1144.     $this->load->view('layout/template', $data);
1145. }
1146. }

```

Kode program pada Tabel 4.8 merupakan kode program *function* `metode_borda()` yang digunakan untuk melakukan perhitungan perankingan lokasi hasil keputusan kelompok menggunakan metode Borda. Tahapan pertama dari perhitungan Borda adalah mendeklarasikan variabel-variabel yang digunakan dan mengambil data dari *database* dengan memanggil *function* `ambil_vikor()`, `ambil_vikor_kelurahan()`, `ambil_vikor_user()` serta menyimpan data dan jumlah data tersebut ke dalam variabel-variabel yang dapat dilihat pada baris program 990-996. Langkah selanjutnya adalah membuat metrik ranking lokasi hasil keputusan oleh setiap pengambil keputusan pada baris program 998-1018. Setiap ranking lokasi akan diberikan sebuah *point* Borda. Ranking lokasi yang pertama akan mendapatkan *point* sebanyak jumlah lokasi yang ada. Pemberian *point* Borda dapat dilihat pada baris program 1020-1036. Langkah selanjutnya adalah menghitung Borda *count* untuk setiap kelurahan dengan cara menjumlahkan *point* Borda pada kelurahan masing-masing. Perhitungan Borda *count* terdapat pada baris program 1038-1046. Hasil Borda *count* kemudian diurutkan dari nilai yang paling besar sampai nilai yang paling kecil. Borda *count* yang paling banyak akan direkomendasikan menjadi lokasi terbaik untuk dibangun sebuah ATM baru. Pengurutan Borda *count* terdapat pada baris 1048-1074. Borda *count* yang sama akan mendapatkan ranking yang sama menggunakan baris program 1076-1106. Langkah selanjutnya adalah memeriksa apakah nilai Borda dan ranking lokasi sudah pernah tersimpan di dalam *database* atau belum. Apabila sudah pernah tersimpan, maka sistem menjalankan *function* `ubah_borda()` untuk mengubah data di *database* dan apabila belum pernah tersimpan, maka sistem menjalankan *function* `simpan_borda()` untuk menyimpan data di *database* yang dapat dilihat pada baris program 1108-1134. Langkah selanjutnya adalah menampilkan hasil dari perhitungan Borda ke dalam tampilan pengambil keputusan dan admin mulai dari metrik ranking lokasi setiap pengambil keputusan sampai hasil perankingan lokasi sebagai keputusan kelompok menggunakan kode program pada baris ke 1135 sampai 1146. Penulisan kode program *function* lainnya pada *package controllers* dan *models* dapat dilihat pada Lampiran D (Kode Program).

2) Listing Program Mengubah Analisa DSS

```

1220 public function proses_ubah_analisa_dss($id_transaksi) } 1
1221     $id_user = $this->login->post('id_user', true);
1222     $from_transaksi = $this->login->post('from_transaksi', true);
1223     $tgl_mulai = $this->login->post('tgl_mulai', true);
1224     $tgl_akhir = $this->login->post('tgl_akhir', true);
1225     $id_kontraktor = $this->login->post('id_kontraktor', true);
1226     $query = $this->db->ubah_analisa_dss($from_transaksi, $tgl_mulai, $tgl_akhir, $id_kontraktor, $id_transaksi);
1227     $id } 3
1228     $query2 = $this->db->get_detail_analisa($id_transaksi); } 4
1229     $id } 6
1230     $query3 = $this->db->get_detail_analisa($id_transaksi, $from_transaksi); } 7
1231     $query4 = $this->db->get_detail_analisa($id_transaksi, $from_transaksi); } 8
1232     }
1233     } 9
1234     $this->session->set_userdata('operation', "sukses");
1235     $this->session->set_userdata('message', "Analisa DSS berhasil di Ubah");
1236     $redirect = "c_dss/lihat_analisa_dss"; } 10
1237     } else {
1238     $this->session->set_userdata('operation', "gagal");
1239     $this->session->set_userdata('message', "Analisa DSS gagal di Ubah");
1240     $redirect = "c_dss/lihat_analisa_dss"; } 11
1241     } else {
1242     $this->session->set_userdata('operation', "gagal");
1243     $this->session->set_userdata('message', "Analisa DSS gagal di Ubah");
1244     $redirect = "c_dss/lihat_analisa_dss"; } 5
1245     }
1246     } 12

```

Gambar 4.10 Listing Program Mengubah Analisa DSS

3) Listing Program Menghapus Analisa DSS

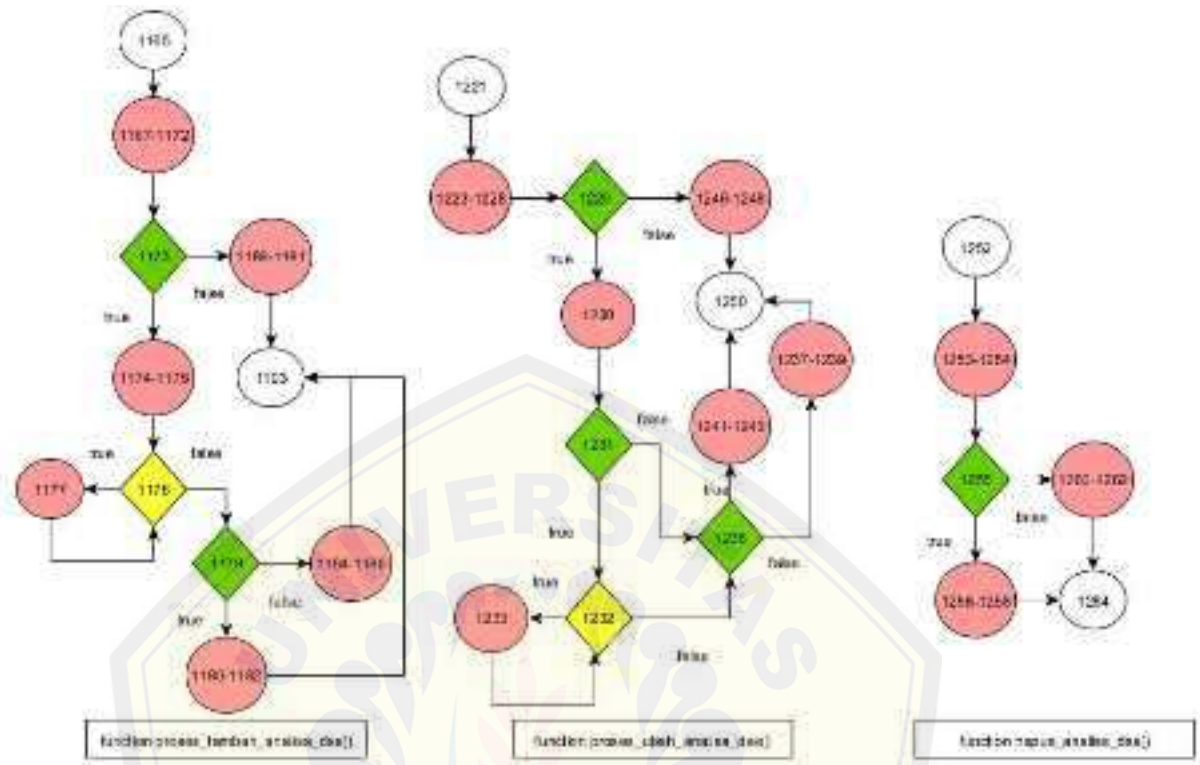
```

1253 public function hapus_analisa_dss($id_transaksi) } 1
1254     $query = $this->db->hapus_detail_analisa($id_transaksi);
1255     $query2 = $this->db->hapus_analisa_dss($id_transaksi); } 2
1256     if ($query && $query2) } 3
1257     $this->session->set_userdata('operation', "sukses");
1258     $this->session->set_userdata('message', "Analisa DSS Sukses di Hapus");
1259     $redirect = "c_dss/lihat_analisa_dss"; } 4
1260     } else {
1261     $this->session->set_userdata('operation', "gagal");
1262     $this->session->set_userdata('message', "Analisa DSS gagal di Hapus");
1263     $redirect = "c_dss/lihat_analisa_dss"; } 5
1264     }
1265     } 6

```

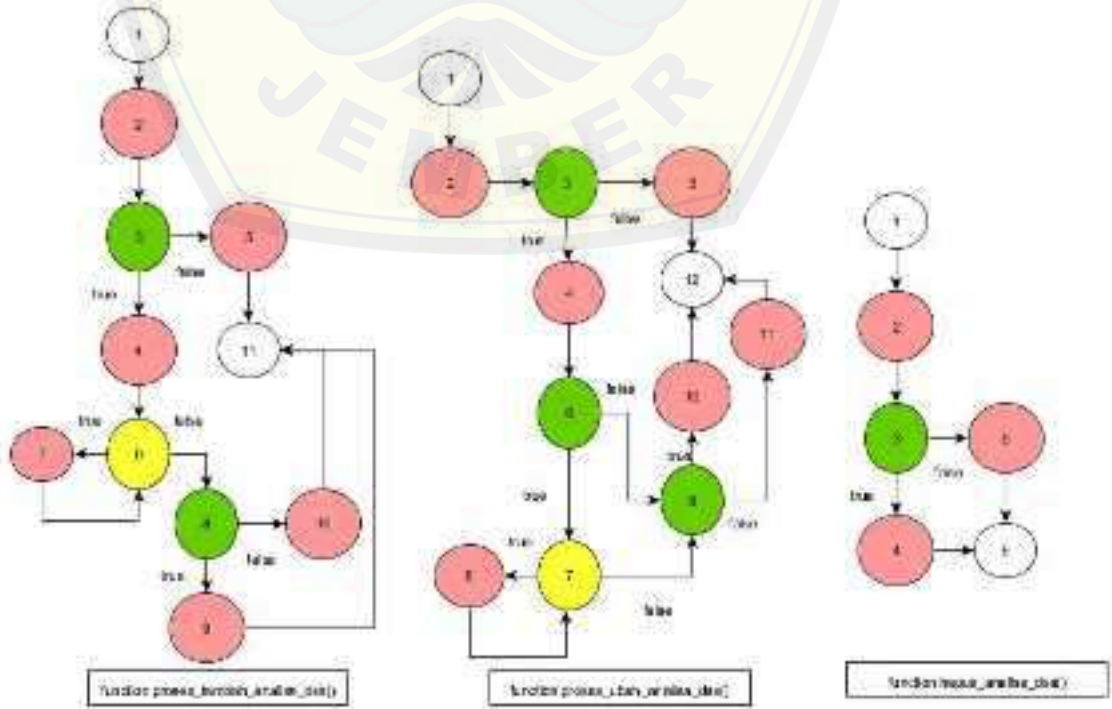
Gambar 4.11 Listing Program Menghapus Analisa DSS

(b). Diagram Alir Fitur Mengelola Analisa DSS



Gambar 4.12 Diagram Alir Mengelola Analisa DSS

(c). Grafik Alir Fitur Mengelola Analisa DSS



Gambar 4.13 Grafik Alir Fitur Mengelola Analisa DSS

(d). Kompleksitas Siklometik

Penghitungan kompleksitas siklomatik $V(G)$ untuk grafik alir pada Gambar 4.13 menggunakan rumus $V(G) = E - N + 2$. Dimana E adalah jumlah *edge* grafik alir dan N adalah jumlah *node* grafik alir.

$$\text{Fuction proses_tambah_analisa_dss() } V(G) = E - N + 2 = 13 - 11 + 2 = 4$$

$$\text{Fuction proses_ubah_analisa_dss() } V(G) = E - N + 2 = 15 - 12 + 2 = 5$$

$$\text{Fuction hapus_analisa_dss() } V(G) = E - N + 2 = 6 - 6 + 2 = 2$$

(e). Pengujian Jalur Program Fitur Mengelola Analisa DSS

Pengujian jalur program fitur mengelola analisa DSS berdasarkan pada Gambar 4.13 adalah sebagai berikut :

1) *Fuction* proses_tambah_analisa_dss() :

Jalur 1 : 1 - 2 - 3 - 4 - 6 - 7 - 8 - 9 - 11

Jalur 2 : 1 - 2 - 3 - 4 - 6 - 7 - 8 - 10 - 11

Jalur 3 : 1 - 2 - 3 - 4 - 6 - 8 - 10 - 11

Jalur 4 : 1 - 2 - 3 - 4 - 5 - 11

2) *Fuction* proses_ubah_analisa_dss() :

Jalur 1 : 1 - 2 - 3 - 4 - 6 - 7 - 8 - 7 - 9 - 10 - 12

Jalur 2 : 1 - 2 - 3 - 4 - 6 - 7 - 8 - 7 - 9 - 11 - 12

Jalur 3 : 1 - 2 - 3 - 4 - 6 - 7 - 9 - 11 - 12

Jalur 4 : 1 - 2 - 3 - 4 - 6 - 9 - 11 - 12

Jalur 5 : 1 - 2 - 3 - 5 - 12

3) *Fuction* hapus_analisa_dss() :

Jalur 1 : 1 - 2 - 3 - 4 - 6

Jalur 2 : 1 - 2 - 3 - 5 - 6

(f). *Test Case* Fitur Mengelola Analisa DSS

Tabel 4.9 *Test Case* Fitur Mengelola Analisa DSS

Test Case function proses_tambah_analisa_dss()

Jalur 1

<i>Test Case</i>	Jika berhasil menyimpan analisa DSS dan detail analisa
Target yang diharapkan	Menyimpan analisa dan detail DSS ke <i>database</i> dan menampilkan alert “Selamat, Analisa DSS Sukses di Simpan”
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 11
Jalur 2	
<i>Test Case</i>	Jika gagal menyimpan detail analisa
Target yang diharapkan	Menampilkan alert “Maaf, Analisa DSS Gagal di Simpan”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 6 – 7 – 8 – 10 – 11
Jalur 3	
<i>Test Case</i>	Jika <i>foreach(\$id_user as \$row)</i> tidak dilakukan
Target yang diharapkan	Menampilkan alert “Maaf, Analisa DSS Gagal di Simpan”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 6 – 8 – 10 – 11
Jalur 4	
<i>Test Case</i>	Jika gagal menyimpan data analisa DSS
Target yang diharapkan	Menampilkan alert “Maaf, Analisa DSS Gagal di Simpan”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 5 – 11
Test Case function proses_ubah_analisa_dss()	
Jalur 1	
<i>Test Case</i>	Jika berhasil mengubah analisa DSS, menghapus detail analisa dan menambah detail analisa
Target yang diharapkan	Mengubah analisa DSS, menghapus detail analisa dan menambah detail analisa ke <i>database</i> dan menampilkan alert “Selamat, Analisa DSS Sukses di Ubah”
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6 – 7 – 8 – 7 – 9 – 10 - 12
Jalur 2	
<i>Test Case</i>	Jika gagal menyimpan detail analisa
Target yang diharapkan	Menampilkan alert “Maaf, Analisa DSS Gagal di Ubah”
Hasil pengujian	Gagal

Path/Jalur	1 – 2 – 3 – 4 – 6 – 7 – 8 – 7 – 9 – 11 – 12
Jalur 3	
<i>Test Case</i>	Jika <i>foreach(\$id_user as \$row)</i> tidak dilakukan
Target yang diharapkan	Menampilkan alert “Maaf, Analisa DSS Gagal di Ubah”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 6 – 7 – 9 – 11 – 12
Jalur 4	
<i>Test Case</i>	Jika gagal menghapus detail analisa
Target yang diharapkan	Menampilkan alert “Maaf, Analisa DSS Gagal di Ubah”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 6 – 9 – 11 – 12
Jalur 5	
<i>Test Case</i>	Jika gagal mengubah analisa DSS
Target yang diharapkan	Menampilkan alert “Maaf, Analisa DSS Gagal di Simpan”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 5 – 12
Test Case function hapus_analisa_dss()	
Jalur 1	
<i>Test Case</i>	Jika berhasil menghapus detail dan data analisa DSS
Target yang diharapkan	Menghapus detail dan data analisa DSS dari <i>database</i> dan menampilkan alert “Selamat, Analisa DSS Sukses di hapus”
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6
Jalur 2	
<i>Test Case</i>	Jika gagal menghapus detail dan data analisa DSS
Target yang diharapkan	Menampilkan alert “Maaf, Analisa DSS Gagal di Hapus”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 5 – 6

2. Pengujian *White Box* Fitur Memilih Kriteria dan Menentukan Bobot Awal
 Pengujian *white box* memilih kriteria dan menentukan bobot awal oleh pengambil keputusan meliputi *listing* program, grafik alir, kompleksitas siklomatis,

pengujian jalur program dan *test case*. Pengujian *white box* fitur memilih kriteria dan menentukan bobot awal oleh pengambil keputusan adalah sebagai berikut:

(a). Listing Program Fitur Memilih Kriteria dan Menentukan Bobot Awal

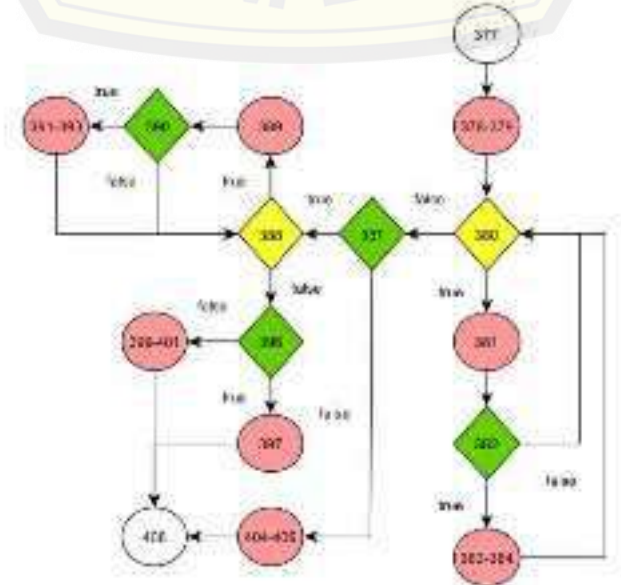
```

public function proses_tambah_bobot_awal($id_transaksi, $id_user) { } 1
370 $cek = $this->input->post('id', true); } 2
371 $skorpen = 0;
380 for ($i = 1; $i <= $cek; $i++) { } 3
381 $cek = $this->input->post('check', $i, true); } 4
382 if ($cek != null) { } 5
383 $bobot_awal = $this->input->post('bobot', $i, true); } 6
384 $skorpen = $skorpen + $bobot_awal;
385 }
386 }
387 if ($skorpen == 1) { } 7
388 for ($a = 1; $a <= $cek; $a++) { } 8
389 $cek = $this->input->post('iduser', $a, true); } 10
390 if ($cek != null) { } 11
391 $id_transaksi = $this->input->post('id_transaksi', $a, true);
392 $bobot_awal = $this->input->post('bobot', $a, true);
393 $source = $this->input->post('source', $id_transaksi, $id_user, $bobot_awal);
394 }
395 }
396 }
397 }
398 }
399 $this->session->user->userdata('password', 'gagal');
400 $this->session->user->userdata('password', 'Coba Memasukkan Bobot Awal');
401 redirect('user/lihat_proses/$id_transaksi/$id_user');
402 }
403 }
404 $this->session->user->userdata('password', 'gagal');
405 $this->session->user->userdata('password', 'Masukan bobot awal harus sama dengan 1');
406 redirect('user/lihat_proses/$id_transaksi/$id_user');
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }

```

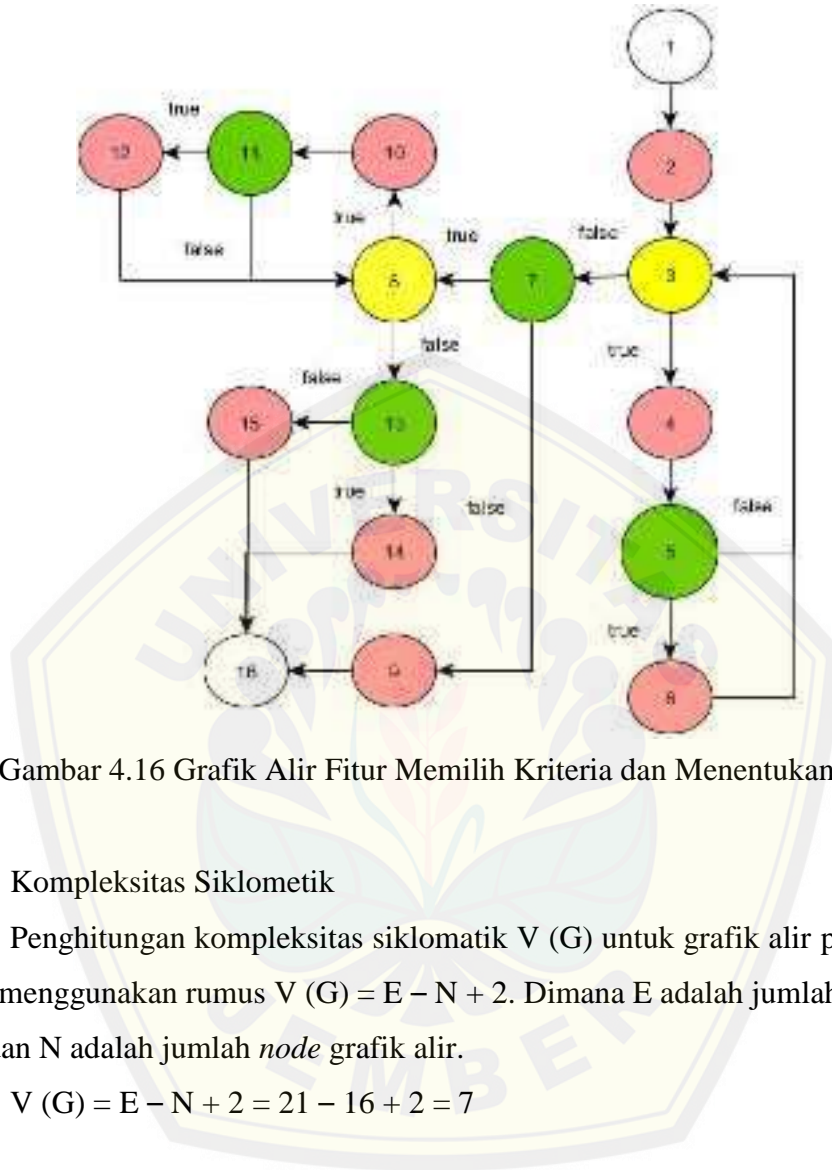
Gambar 4.14 Listing Program Fitur Memilih Kriteria dan Menentukan Bobot Awal Kriteria

(b). Diagram Alir Fitur Memilih Kriteria dan Menentukan Bobot Awal



Gambar 4.15 Diagram Alir Fitur Memilih Kriteria dan Menentukan Bobot Awal

(c). Grafik Alir Fitur Memilih Kriteria dan Menentukan Bobot Awal



Gambar 4.16 Grafik Alir Fitur Memilih Kriteria dan Menentukan Bobot Awal

(d). Kompleksitas Siklometik

Penghitungan kompleksitas siklometik $V(G)$ untuk grafik alir pada Gambar 4.16 menggunakan rumus $V(G) = E - N + 2$. Dimana E adalah jumlah *edge* grafik alir dan N adalah jumlah *node* grafik alir.

$$V(G) = E - N + 2 = 21 - 16 + 2 = 7$$

(e). Pengujian Jalur Program Fitur Memilih Kriteria dan Menentukan Bobot Awal

Pengujian jalur program fitur memilih kriteria dan menentukan bobot awal berdasarkan pada Gambar 4.16 adalah sebagai berikut :

Jalur 1 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 16

Jalur 2 : 1 - 2 - 3 - 7 - 9 - 16

Jalur 3 : 1 - 2 - 3 - 4 - 5 - 7 - 9 - 16

Jalur 4 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 9 - 16

Jalur 5 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 13 - 15 - 16

Jalur 6 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 13 – 15 – 16

Jalur 7 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 15 – 16

(f). *Test Case* Fitur Memilih Kriteria dan Menentukan Bobot Awal

Tabel 4.10 *Test Case* Fitur Memilih Kriteria dan Menentukan Bobot Awal

Test Case function proses1_tambah_bobot_awal()	
Jalur 1	
<i>Test Case</i>	Jika berhasil menyimpan kriteria dan bobot awal ke <i>database</i>
Target yang diharapkan	Menyimpan kriteria dan bobot awal ke <i>database</i> dan menjalankan <i>function metode_entropy()</i>
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 16
Jalur 2	
<i>Test Case</i>	Jika <i>for(\$a=1;\$a<\$no;\$a++)</i> pada <i>list 3</i> tidak dilakukan
Target yang diharapkan	Menampilkan <i>alert</i> “Maaf, Jumlah bobot awal harus sama dengan 1”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 7 – 9 – 16
Jalur 3	
<i>Test Case</i>	Jika <i>if(\$cek!=null)</i> pada <i>list 5</i>
Target yang diharapkan	Menampilkan <i>alert</i> “Maaf, Jumlah bobot awal harus sama dengan 1”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 5 – 7 – 9 – 16
Jalur 4	
<i>Test Case</i>	Jika jumlah bobot awal tidak sama dengan 1
Target yang diharapkan	Menampilkan <i>alert</i> “Maaf, Jumlah bobot awal harus sama dengan 1”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 16
Jalur 5	
<i>Test Case</i>	Jika <i>for(\$a=1;\$a<\$no;\$a++)</i> pada <i>list 3</i> tidak dilakukan

Target yang diharapkan	Menampilkan <i>alert</i> “Maaf, Gagal menambah bobot awal”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 13 – 15 – 16
Jalur 6	
<i>Test Case</i>	Jika <i>if(\$cek!=null)</i> pada <i>list 11</i>
Target yang diharapkan	Menampilkan <i>alert</i> “Maaf, Gagal menambah bobot awal”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 13 – 15 – 16
Jalur 7	
<i>Test Case</i>	Jika gagal menyimpan kriteria dan bobot awal
Target yang diharapkan	Menampilkan <i>alert</i> “Maaf, Gagal menambah bobot awal”
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 15 – 16

3. Pengujian *White Box* Perhitungan Metode *Entropy*

Pengujian *white box* perhitungan metode *entropy* meliputi *listing* program, grafik alir, kompleksitas siklomatis, pengujian jalur program dan *test case*. Pengujian *white box* perhitungan metode *entropy* adalah sebagai berikut:

(a). Listing Program Perhitungan Metode Entropy

```

public function metode_entropy($id_transaksi, $id_user) { } 1
412     $sentrik = array();
413     $nilai = array();
414     $group = array();
415     $jumlah = array();
416     $jumlahn = array();
417     $normalisasi = array();
418     $jum = array();
419     $jumlahn = array();
420     $jumlahentropy = 0;
421     $jumlah = 0;
422     $probabilitas = array();
423     $logprobabilitas = array();
424     $entropi = array();
425     $sigma = array();
426     $sigmaobot = array();
427     $entropi = array();
428     $jumlahdata = count($id_transaksi);
429     $keluaran = array();
430     $probabilitas = array();
431     $skor = array();
432     $jumlah = count($keluaran);
433     $i = 0;
434     foreach ($skor as $row) { } 3
435         $keluaran[$i] = $row['keluaran'];
436         $skor[$i] = $row['skor'];
437         $i++;
438     }
439     //----- Menentukan nilai probabilitas ke sentrik 2D
440     $sentrik = array();
441     $sigma = array();
442     $i = 0;
443     foreach ($keluaran as $row) { } 6
444         $i++;
445         $sigma[$i] = $row['sigma'];
446         $sentrik[$i] = $row['sentrik'];
447         $i++;
448     }
449     //----- Menentukan nilai probabilitas ke sentrik 2D
450     $sentrik = array();
451     $sigma = array();
452     $i = 0;
453     foreach ($sentrik as $row) { } 10
454         $i++;
455         $sentrik[$i] = $row['sentrik'];
456         $sigma[$i] = $row['sigma'];
457         $i++;
458     }
459     //----- Menentukan nilai probabilitas ke sentrik 2D
460     $sentrik = array();
461     $sigma = array();
462     $i = 0;
463     foreach ($sentrik as $row) { } 13
464         $i++;
465         $sentrik[$i] = $row['sentrik'];
466         $sigma[$i] = $row['sigma'];
467         $i++;
468     }
469     $sentrik[0] = max($sentrik);
470     $sigma[0] = max($sigma);
471     }

```

```

470 //menarik nilai minimal per kriteria
471 for (jy = 0; jy < Sjumlahkriteria; jy++) } 17
472   for (ji = 0; ji < Sjumlah; ji++) } 18
473     Stemp[ji] = Satrik[Sa][jy]; } 19
474   }
475   Ssolusimin[jy] = min(Stemp); } 20
476   unset(Stemp);
477 }
478 //----- MENENTUKAN METODE NORMALISASI -----
479 for (jy = 0; jy < Sjumlahkriteria; jy++) } 21
480   for (ji = 0; ji < Sjumlah; ji++) } 22
481     if (Satribut[jy] == "1") } 23
482       Snormalisasi[ji][jy] = Satrik[ji][jy] / Ssolusimax[jy]; //normalisasi } 24
483     } else {
484       Snormalisasi[ji][jy] = Ssolusimin[jy] / Satrik[ji][jy]; //normalisasi } 25
485     }
486   }
487 }
488 //----- MENENTUKAN METODE ENTROPY -----
489 //menentukan jumlah nilai Snormalisasi per kriteria
490 for (jy = 0; jy < Sjumlahkriteria; jy++) } 26
491   Stemp = 0; } 27
492   for (ji = 0; ji < Sjumlah; ji++) } 28
493     Stemp = Stemp + Snormalisasi[ji][jy]; } 29
494   }
495   Sjum[jy] = Stemp; } 30
496 }
497 //----- MENENTUKAN METODE PROBABILITAS -----
498 for (jy = 0; jy < Sjumlahkriteria; jy++) } 31
499   for (ji = 0; ji < Sjumlah; ji++) } 32
500     Sprobabilitas[ji][jy] = Snormalisasi[ji][jy] / Sjum[jy]; } 33
501   }
502 }
503 //nilai probabilitas = nilai Sprobabilitas
504 for (jy = 0; jy < Sjumlahkriteria; jy++) } 34
505   for (ji = 0; ji < Sjumlah; ji++) } 35
506     Slogprobabilitas[ji][jy] = Sprobabilitas[ji][jy] * log(Sprobabilitas[ji][jy]); //M_KI } 36
507   }
508 }
509 //jumlah log probabilitas per kriteria
510 for (jy = 0; jy < Sjumlahkriteria; jy++) } 37
511   Stemp = 0; } 38
512   for (ji = 0; ji < Sjumlah; ji++) } 39
513     Stemp = Stemp + Slogprobabilitas[ji][jy]; } 40
514   }
515   Sjumlog[jy] = Stemp; } 41
516 }
517 //----- MENENTUKAN METODE ENTROPY -----
518 for (jy = 0; jy < Sjumlahkriteria; jy++) } 42
519   Satrikentropy[jy] = (1 - Sjumlog[jy]) / log(Sjumlah, M_KI) * Sjumlog[jy]; } 43
520 }
521 //----- MENENTUKAN METODE ENTROPY -----
522 //jumlah nilai entropy
523 for (jy = 0; jy < Sjumlahkriteria; jy++) } 44
524   Sjumentropy = Sjumentropy + Satrikentropy[jy]; } 45
525 }
526 //hitung Sinda
527 for (k = 0; k < Sjumlahkriteria; k++) } 46
528   Sinda[k] = (1 / Sjumlahkriteria - Sjumentropy) * (1 - Satrikentropy[k]); } 47
529 }
530 //----- MENENTUKAN METODE ENTROPY -----
531 for (k = 0; k < Sjumlahkriteria; k++) } 48
532   Sindeks[k] = Sinda[k] * Satribut[k]; } 49
533 }

```



```

334 //MELAKUKAN BILAI BANYAK SAMA PER ENTROPY
335 for (i = 0; i < $jumlahbaris; i++) { } 50
336     $jumlah = $jumlah + $databaris[i]; } 51
337 }
338 //MELAKUKAN ENTROPY PER KATEGORI
339 for (i = 0; i < $jumlahbaris; i++) { } 52
340     if ($databaris[i] == 0) } 53
341         $entropy[i] = 0; } 54
342     } else {
343         $entropy[i] = $databaris[i] / $jumlahbaris; } 55
344     }
345 }
346 //MENCARI ENTROPY
347 for (i = 0; i < $jumlahbaris; i++) { } 56
348     $bobot_entropy = $entropy[i];
349     $id_kategori = $id_kategori[i];
350     $query = "SELECT SUM(das_jumlah_baris_bobot_entropy * $id_kategori, $id_user, $id_kategori, $bobot_entropy)";
351 } 57
352 if ($query) } 58
353     $data["normalisasi"] = $normalisasi;
354     $data["probabilitas"] = $probabilitas;
355     $data["logprobabilitas"] = $logprobabilitas;
356     $data["burgibentropy"] = $burgibentropy;
357     $data["jumlah"] = $jumlah;
358     $data["jumlahbaris"] = $jumlahbaris;
359     $data["entropy"] = $entropy;
360     $data["jumlah"] = $jumlah;
361     $data["kategori"] = "Walaupun ini adalah kategori yang digunakan untuk";
362     $data["user"] = "Nama dan data statistik ($id_kategori, $id_user)";
363     $data["kategori"] = "Nama dan data statistik ($id_kategori, $id_user)";
364     $data["user"] = "Nama dan data statistik ($id_kategori, $id_user)";
365     $data = "Nama kategori";
366     $data["user"] = "Nama dan data statistik ($id_kategori, $id_user)";
367     $data["user"] = "Nama dan data statistik ($id_kategori, $id_user)";
368     $data["user"] = "Nama dan data statistik ($id_kategori, $id_user)";
369     $data["user"] = "Nama dan data statistik ($id_kategori, $id_user)";
370 } 60
371 } } 61

```

Gambar 4.17 Listing Program Perhitungan Metode Entropy

(d). Kompleksitas Siklometik

Penghitungan kompleksitas siklomatik $V(G)$ untuk grafik alir pada Gambar 4.19 menggunakan rumus $V(G) = E - N + 2$. Dimana E adalah jumlah *edge* grafik alir dan N adalah jumlah *node* grafik alir.

$$V(G) = E - N + 2 = 89 - 61 + 2 = 30$$

(e). Pengujian Jalur Program Perhitungan Metode *Entropy*

Pengujian jalur program perhitungan metode *Entropy* berdasarkan pada Gambar 4.19 adalah sebagai berikut :

Jalur 1 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 55 - 56 - 57 - 58 - 59 - 61

Jalur 2 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 55 - 56 - 57 - 58 - 60 - 61; 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 54 - 56 - 57 - 58 - 60 - 61 (terdapat 16 kombinasi)

Jalur 3 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 54 - 56 - 57 - 58 - 60 - 61; 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 55 - 56 - 57 - 58 - 60 - 61 (terdapat 16 kombinasi)

Jalur 4 : 1 - 2 - 3 - 4 - 5 - 6 - 7 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33

– 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 54 – 56 – 57 – 58 – 60 – 61; 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 55 – 56 – 57 – 58 – 60 – 61 (terdapat 16 kombinasi)

Jalur 5 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 54 – 56 – 57 – 58 – 60 – 61; 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 55 – 56 – 57 – 58 – 60 – 61 (terdapat 16 kombinasi).

Jalur 6 : 1 – 2 – 3 – 5 – 6 – 10 – 13 – 17 – 21 – 25 – 31 – 34 – 37 – 42 – 43 – 44 – 46 – 48 – 50 – 52 – 56 – 58 – 60 – 61(terdapat 16 kombinasi).

Jalur 7 : 1 – 2 – 3 – 4 – 5 – 6 – 10 – 13 – 17 – 21 – 26 – 31 – 34 – 37 – 42 – 43 – 44 – 46 – 48 – 50 – 52 – 56 – 58 – 60 – 61(terdapat 16 kombinasi).

Jalur 8 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 13 – 17 – 21 – 26 – 31 – 34 – 37 – 42 – 43 – 44 – 46 – 48 – 50 – 52 – 56 – 58 – 60 – 61(terdapat 8 kombinasi).

Jalur 9 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 13 – 17 – 21 – 26 – 31 – 34 – 37 – 42 – 43 – 44 – 46 – 48 – 50 – 52 – 56 – 58 – 60 – 61(terdapat 8 kombinasi).

Jalur 10 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 17 – 21 – 26 – 31 – 34 – 37 – 42 – 43 – 44 – 46 – 48 – 50 – 52 – 56 – 58 – 60 – 61(terdapat 8 kombinasi).

Jalur 11 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 17 – 21 – 26 – 31 – 34 – 37 – 42 – 43 – 44 – 46 – 48 – 50 – 52 – 56 – 58 – 60 – 61(terdapat 8 kombinasi).

Jalur 29 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 56 – 58 – 60 – 61(terdapat 4 kombinasi).

Jalur 30 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 56 – 58 – 60 – 61(terdapat 2 kombinasi).

(f). *Test Case* Perhitungan Metode *Entropy*

Tabel 4.11 *Test Case* Perhitungan Metode *Entropy*

Test Case function metode <i>entropy</i>()	
Jalur 1	
<i>Test Case</i>	Jika berhasil menghitung bobot <i>entropy</i> dan menyimpan bobot <i>entropy</i> ke <i>database</i>
Target yang diharapkan	Menyimpan bobot <i>entropy</i> ke <i>database</i> dan menjalankan <i>function</i> metode <i>vikor</i> ()
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 55 – 56 – 57 – 58 – 59 – 61
Jalur 2	
<i>Test Case</i>	Jika <i>if(\$kel!=\$row), if(\$sifat[\$j]==1)</i>
Target yang diharapkan	Menyimpan nilai kepentingan ke metrik 2 dimensi yang kriterianya bersifat <i>benefit</i>
Hasil pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 55 – 56 – 57 – 58 – 60 – 61
Jalur 3	
<i>Test Case</i>	Jika <i>if(\$kel!=\$row), if(\$sifat[\$j]==2)</i>
Target yang diharapkan	Menyimpan nilai kepentingan ke metrik 2 dimensi yang kriterianya bersifat <i>cost</i>
Hasil pengujian	Benar

Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 54 - 56 - 57 - 58 - 60 - 61
Jalur 4	
<i>Test Case</i>	Jika <i>if(\$kel==\$row), if(\$sifat[\$j]==1)</i>
Target yang diharapkan	Menyimpan nilai kepentingan ke metrik 2 dimensi yang kriterianya bersifat <i>benefit</i>
Hasil pengujian	Benar
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 54 - 56 - 57 - 58 - 60 - 61
Jalur 5	
<i>Test Case</i>	Jika <i>if(\$kel==\$row), if(\$sifat[\$j]==2)</i>
Target yang diharapkan	Menyimpan nilai kepentingan ke metrik 2 dimensi yang kriterianya bersifat <i>cost</i>
Hasil pengujian	Benar
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 54 - 56 - 57 - 58 - 60 - 61
Jalur 6	
<i>Test Case</i>	Jika <i>foreach (\$kriteria as \$row)</i> tidak dilakukan
Target yang diharapkan	Tidak dapat mengambil kriteria dari <i>database</i>
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 5 - 6 - 10 - 13 - 17 - 21 - 25 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 7	
<i>Test Case</i>	Jika <i>foreach (\$ambildata as \$row)</i> tidak dilakukan
Target yang diharapkan	Tidak dapat mengambil data kelurahan dan nilai kepentingan dari <i>database</i>
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 10 - 13 - 17 - 21 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 8	

<i>Test Case</i>	Jika <i>for</i> ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 10 tidak dilakukan
Target yang diharapkan	Tidak dapat membuat metrik kriteria dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 13 - 17 - 21 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 9	
<i>Test Case</i>	Jika <i>for</i> ($\$i=0; \$i < \$jumdesa; \$i++$) pada list 11 tidak dilakukan
Target yang diharapkan	Tidak dapat membuat metrik kriteria dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 13 - 17 - 21 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 10	
<i>Test Case</i>	Jika <i>for</i> ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 13 tidak dilakukan
Target yang diharapkan	Tidak dapat menghitung nilai maksimal per kriteria dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 17 - 21 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 11	
<i>Test Case</i>	Jika <i>for</i> ($\$i=0; \$i < \$jumdesa; \$i++$) pada list 14 tidak dilakukan
Target yang diharapkan	Tidak dapat menghitung nilai maksimal per kriteria dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 17 - 21 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 12	
<i>Test Case</i>	Jika <i>for</i> ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 17 tidak dilakukan
Target yang diharapkan	Tidak dapat menghitung nilai minimal per kriteria dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal

Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 21 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 13	
Test Case	Jika <i>for</i> ($\$i=0; \$i < \$jumdesa; \$i++$) pada list 18 tidak dilakukan
Target yang diharapkan	Tidak dapat menghitung nilai minimal per kriteria dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 21 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 14	
Test Case	Jika <i>for</i> ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 21 tidak dilakukan
Target yang diharapkan	Tidak dapat membuat metrik ternormalisasi dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 15	
Test Case	Jika <i>for</i> ($\$i=0; \$i < \$jumdesa; \$i++$) pada list 22 tidak dilakukan
Target yang diharapkan	Tidak dapat membuat metrik ternormalisasi dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 16	
Test Case	Jika <i>for</i> ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 26 tidak dilakukan
Target yang diharapkan	Tidak dapat menghitung jumlah nilai normalisasi dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 17	

<i>Test Case</i>	Jika for ($\$i=0; \$i < \$jumdesa; \$i++$) pada list 28 tidak dilakukan
Target yang diharapkan	Tidak dapat menghitung jumlah nilai normalisasi dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 18	
<i>Test Case</i>	Jika for ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 31 tidak dilakukan
Target yang diharapkan	Tidak dapat membuat metrik probabilitas dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 19	
<i>Test Case</i>	Jika for ($\$i=0; \$i < \$jumdesa; \$i++$) pada list 32 tidak dilakukan
Target yang diharapkan	Tidak dapat membuat metrik probabilitas dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 20	
<i>Test Case</i>	Jika for ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 34 tidak dilakukan
Target yang diharapkan	Tidak dapat menghitung nilai probabilitas dikalikan ln probabilitas dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 21	
<i>Test Case</i>	Jika for ($\$i=0; \$i < \$jumdesa; \$i++$) pada list 35 tidak dilakukan

Target yang diharapkan	Tidak dapat menghitung nilai probabilitas dikalikan ln probabilitas dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 22	
Test Case	Jika for ($j=0; j < \text{Jumkriteria}; j++$) pada list 39 tidak dilakukan
Target yang diharapkan	Tidak dapat menghitung jumlah ln probabilitas dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 23	
Test Case	Jika for ($i=0; i < \text{Jumdesa}; i++$) pada list 39 tidak dilakukan
Target yang diharapkan	Tidak dapat menghitung nilai probabilitas * ln probabilitas dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 24	
Test Case	Jika for ($j=0; j < \text{Jumkriteria}; j++$) pada list 42 tidak dilakukan
Target yang diharapkan	Tidak dapat dapat membuat metrik nilai entropy dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 25	
Test Case	Jika for ($j=0; j < \text{Jumkriteria}; j++$) pada list 44 tidak dilakukan

Target yang diharapkan	Tidak dapat menghitung jumlah nilai <i>entropy</i> dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 26	
Test Case	Jika <i>for</i> ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 46 tidak dilakukan
Target yang diharapkan	Tidak dapat dapat menghitung lamda dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 27	
Test Case	Jika <i>for</i> ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 48 tidak dilakukan
Target yang diharapkan	Tidak dapat dapat menghitung lamda terbobot dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 28	
Test Case	Jika <i>for</i> ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 50 tidak dilakukan
Target yang diharapkan	Tidak dapat dapat menghitung jumlah lamda terbobot dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 52 - 56 - 58 - 60 - 61
Jalur 29	
Test Case	Jika <i>for</i> ($\$j=0; \$j < \$jumkriteria; \$j++$) pada list 52 tidak dilakukan

Target yang diharapkan	Tidak dapat dapat menghitung bobot <i>Entropy</i> dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 56 – 58 – 60 – 61
Jalur 30	
<i>Test Case</i>	Jika <i>for (\$j=0;\$j<\$jumkriteria;\$j++)</i> pada <i>list</i> 56 tidak dilakukan
Target yang diharapkan	Tidak dapat dapat menyimpan bobot <i>Entropy</i> ke dalam <i>database</i> dan menampilkan halaman proses pilih kriteria
Hasil pengujian	Gagal
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 56 – 58 – 60 – 61

4.5.2 Black Box Testing

Pengujian *black box* dilakukan untuk mengetahui apakah masukan dan keluaran dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada form untuk setiap *usecase*. Pengujian ini dilakukan oleh *developer* dan pengambil keputusan GDSS Penentuan Lokasi Penempatan ATM dengan menggunakan metode *Entropy*, VIKOR dan Borda. Hasil pengujian *black box usecase* mengelola analisa DSS dapat dilihat pada Tabel 4.12.

Tabel 4.12 Hasil Pengujian *Black Box Usecase* Mengelola Analisa DSS

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat Analisa DSS	Memilih menu Analisa DSS	a) Mengambil data analisa DSS dari <i>database</i> b) Menampilkan data analisa DSS pada tabel di halaman data analisa DSS	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2.	Melihat Detail Analisa DSS	Memilih tombol “Detail Analisa” pada salah satu baris data analisa DSS	a) Mengambil detail analisa DSS dari <i>database</i> berdasarkan analisa yang dipilih b) Menampilkan detail analisa DSS pada tabel di halaman detail analisa DSS	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol “Back”	a) Mengambil data analisa DSS dari <i>database</i> b) Menampilkan data analisa DSS pada tabel di halaman data analisa DSS	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
3.	Melihat Hasil GDSS	Memilih tombol “Hasil GDSS” pada salah satu baris data analisa dss.	a) Mengambil hasil GDSS dari <i>database</i> berdasarkan analisa yang dipilih b) Menampilkan hasil analisa pengambil keputusan, hasil analisa kelompok, grafik hasil GDSS, peta hasil GDSS dan rekomendasi lokasi penempatan ATM terbaik	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

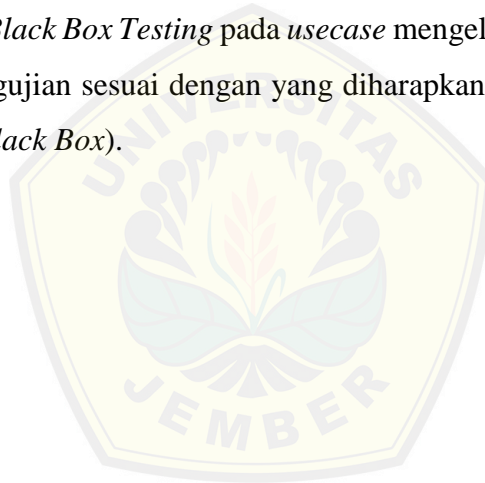
		Klik tombol “Back”	a) Mengambil data analisa DSS dari <i>database</i> b) Menampilkan data analisa DSS pada tabel di halaman data analisa DSS	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
4.	Mengubah Status Analisa DSS	Menekan tombol “Aktif” pada kolom status di baris data yang akan diubah statusnya	a) Mengubah status analisa DSS dan menyimpan perubahannya ke <i>database</i> b) Menampilkan <i>alert</i> “Selamat! Analisa Diaktifkan” c) Mengambil data analisa dss dari <i>database</i> d) Menampilkan data analisa pada tabel di halaman data analisa dss	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Menekan tombol “Tidak Aktif” pada kolom status di baris data yang akan diubah statusnya	a) Mengubah status analisa DSS dan menyimpan perubahannya ke <i>database</i> b) Menampilkan <i>alert</i> “Selamat! Analisa Tidak Diaktifkan” c) Mengambil data analisa dss dari <i>database</i> d) Menampilkan data analisa pada tabel di halaman data analisa dss	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
5.	Menambah Analisa DSS	Menekan tombol “Tambah Analisa”	Menampilkan <i>form</i> isian tambah analisa DSS	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi seluruh isian <i>form</i> tambah analisa secara lengkap lalu menekan tombol “Save”	a) Menyimpan masukan data ke <i>database</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		b) Menampilkan <i>alert</i> “Selamat! Analisa DSS sukses disimpan”	
		c) Mengambil data analisa DSS dari <i>database</i>	
		d) Menampilkan data analisa DSS pada tabel di halaman data analisa DSS	
	Mengisi isian <i>form</i> tambah analisa secara tidak lengkap lalu menekan tombol “ <i>Save</i> ”	a) Memeriksa masukan tambah analisa	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		b) Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> isian analisa DSS. Tulisan akan muncul dibawah masing-masing <i>field</i> inputan	
	Klik tombol “ <i>Back</i> ”	a) Mengambil data analisa dss dari <i>database</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		b) Menampilkan data analisa dss pada tabel di halaman data analisa dss	
	Menekan tombol “ <i>Clear</i> ”	Menampilkan <i>form</i> isian tambah analisa DSS yang sudah dikosongkan	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
6.	Mengubah Analisa DSS	Menekan tombol “Ubah Analisa” pada baris data yang akan diubah	a) Mengambil analisa DSS yang dipilih dari <i>database</i> <input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal b) Menampilkan analisa DSS ke dalam <i>form</i> ubah analisa
		Mengubah isian <i>form</i> ubah analisa secara lengkap lalu menekan tombol “ <i>Save Edit</i> ”	a) Menyimpan data yang diubah ke <i>database</i> <input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		b) Menampilkan <i>alert</i> “Selamat! Data Analisa DSS sukses diubah”	
		c) Mengambil data analisa dss dari <i>database</i>	
		d) Menampilkan data analisa pada tabel di halaman data analisa dss	
	Mengubah isian <i>form</i> ubah analisa secara tidak lengkap lalu menekan tombol “ <i>Save Edit</i> ”	a) Memeriksa ubahan analisa DSS b) Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> ubah analisa DSS. Tulisan akan muncul dibawah masing-masing <i>field</i> inputan	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Klik tombol “ <i>Back</i> ”	a) Mengambil data analisa dss dari <i>database</i> b) Menampilkan data analisa dss pada tabel di halaman data analisa dss	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “ <i>Clear</i> ”	Menampilkan <i>form</i> isian ubah analisa DSS yang belum diubah	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
7.	Menghapus Analisa DSS	Menekan tombol “Hapus Analisa” pada baris data yang akan dihapus	Menampilkan <i>alert</i> “Apakah Anda yakin akan menghapus data ini?” disertai dengan tombol OK dan <i>Cancel</i>
		a) Menghapus analisa DSS yang dipilih dari <i>database</i> b) Menampilkan <i>alert</i> “Selamat! Analisa DSS sukses dihapus”	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

	c) Mengambil analisa DSS dari <i>database</i>	
	d) Menampilkan data analisa pada tabel di halaman data analisa DSS	
Menekan tombol “ <i>Cancel</i> ” pada <i>alert</i>	Menampilkan analisa DSS pada halaman data analisa DSS	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Tabel 4.12 merupakan hasil pengujian *Black Box Testing* pada *usecase* mengelola analisa DSS. Hasil pengujian *usecase* mengelola analisa DSS menunjukkan bahwa hasil pengujian sesuai dengan yang diharapkan oleh pengguna. Pengujian *black box* selengkapnya dapat dilihat pada Lampiran E (Pengujian *Black Box*).



BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil pembangunan sistem dan juga pembahasannya. Penelitian ini menghasilkan sebuah *Group Decision Support System* (GDSS) penentuan lokasi penempatan Anjungan Tunai Mandiri menggunakan metode *Entropy*, VIKOR dan Borda yang digunakan untuk menghasilkan rekomendasi lokasi penempatan ATM berdasarkan keputusan kelompok pada PT. BRI (Persero) Tbk Cabang Jember. Pembahasan bertujuan untuk menjelaskan bagaimana penelitian ini menjawab perumusan masalah serta tujuan dan manfaat dari penelitian.

5.1 Hasil Penerapan Metode *Entropy*, VIKOR dan Borda Pada Proses Pengambilan Keputusan

Proses pengambilan keputusan untuk menentukan lokasi penempatan Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember dilakukan oleh 3 pihak yaitu Bidang IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan Bank. Penerapan metode *Entropy*, VIKOR dan Borda pada proses pengambilan keputusan tersebut bertujuan untuk memberikan rekomendasi lokasi penempatan ATM sesuai dengan kriteria sehingga hasil keputusan kelompok lebih objektif dan dapat menampung semua penilaian dari para pengambil keputusan.

Penerapan metode *Entropy*, VIKOR dan Borda pada proses pengambilan keputusan dilakukan untuk menganalisa salah satu kecamatan yang akan dibangun sebuah ATM pada salah satu kelurahan di kecamatan tersebut. Peneliti menerapkan metode *Entropy*, VIKOR dan Borda pada analisa salah satu kecamatan yaitu Kecamatan Kaliwates untuk mengetahui apakah metode-metode tersebut cocok digunakan pada proses pengambilan keputusan pada PT. BRI (Persero) Tbk Cabang Jember sebelum menerapkan metode tersebut ke dalam sistem yang dibangun. Langkah-langkah penerapan metode *Entropy*, VIKOR dan Borda untuk Kecamatan Kaliwates dapat dilihat pada sub-bab Tahap Analisis yang terletak pada BAB 3.

Hasil dari penerapan metode *Entropy*, VIKOR dan Borda pada proses pengambilan keputusan untuk Kecamatan Kaliwates sesuai dengan Tabel 3.63 menunjukkan bahwa Kelurahan Tegal Besar adalah lokasi yang paling direkomendasikan menjadi lokasi penempatan ATM baru.

Kesimpulan yang dapat diperoleh dari penerapan metode *Entropy* pada proses pengambilan keputusan untuk Kecamatan Kaliwates antara lain:

1. Metode *Entropy* dapat menghitung data numerik maupun deskriptif dan setiap data sudah mempunyai batasan dan nilai kepentingan sendiri-sendiri sehingga menghasilkan bobot yang objektif dengan perhitungan yang lebih rinci dan detail.
2. Bobot awal kriteria ditentukan oleh pengambil keputusan secara subjektif. Bobot awal diberikan oleh pengambil keputusan berdasarkan seberapa penting kriteria tersebut mempengaruhi proses pengambilan keputusan
3. Bobot kriteria akhir berasal dari gabungan bobot awal pengambil keputusan dan perhitungan karakteristik data menggunakan metode *Entropy* agar hasil pembobotan lebih objektif tanpa harus mengabaikan penilaian dari pengambil keputusan.
4. Data yang mempunyai batasan nilai yang besar dan mempunyai variasi nilai yang tinggi untuk tiap alternatif, akan memperoleh bobot yang tinggi. Artinya, kriteria tersebut dianggap mampu untuk membedakan performansi tiap alternatif.
5. Hasil dari pembobotan kriteria menunjukkan bahwa bobot awal yang diberikan oleh pengambil keputusan tidak jauh berbeda hasilnya dengan bobot kriteria hasil perhitungan metode *entropy* akan tetapi tergantung dengan batasan nilai dan variasi nilai yang ada.
6. Bobot *entropy* kriteria yang tinggi menandakan bahwa kriteria tersebut merupakan kriteria yang paling berpengaruh dalam proses pengambilan keputusan

Berdasarkan kesimpulan yang didapatkan dari perhitungan metode *Entropy* di atas, peneliti dan pengambil keputusan memutuskan bahwa metode *Entropy* cocok digunakan dalam proses pengambilan keputusan untuk menentukan bobot

kriteria yang sesuai dengan penilaian pengambil keputusan dan karakteristik data yang ada sehingga menghasilkan bobot yang objektif. Sedangkan kesimpulan yang dapat diperoleh dari penerapan metode VIKOR pada proses pengambilan keputusan untuk Kecamatan Kaliwates adalah :

1. Metode VIKOR menentukan ranking dengan mempertimbangkan kedekatan antar alternatif yang ideal maupun tidak ideal dari setiap alternatif sehingga rekomendasi lokasi bisa diketahui, mana lokasi yang sangat direkomendasikan, mana yang kurang direkomendasikan.
2. Alternatif yang mempunyai nilai S, R dan nilai rasio kedekatan(Q) yang paling kecil merupakan alternatif lokasi yang paling direkomendasikan apabila sudah memenuhi beberapa kondisi yang ditentukan pada perhitungan VIKOR.

Berdasarkan kesimpulan yang didapatkan dari perhitungan metode VIKOR di atas, peneliti dan pengambil keputusan memutuskan bahwa metode VIKOR cocok digunakan dalam proses penentuan ranking lokasi karena pengambil keputusan tidak dapat menentukan preferensinya pada saat awal desain sistem sehingga sangat dibutuhkan metode yang dapat menunjukkan mana lokasi yang sangat direkomendasikan dan yang kurang direkomendasikan. Sedangkan kesimpulan yang dapat diperoleh dari penerapan metode Borda pada proses pengambilan keputusan untuk Kecamatan Kaliwates adalah :

1. Metode Borda mengakomodasi semua penilaian pengambil keputusan, tidak ada ranking rendah yang dibuang atau hanya pengambilan mayoritasnya saja sehingga seperti prinsip demokrasi metode Borda dapat menampung semua aspirasi pengambil keputusan.
2. Alternatif yang mendapatkan nilai Borda paling tinggi merupakan alternatif yang paling direkomendasikan untuk dibangun ATM baru di lokasi tersebut.
3. Alternatif yang sering mendapat ranking terbaik akan mendapatkan nilai Borda yang lebih banyak daripada alternatif yang sering mendapat ranking rendah.

Berdasarkan kesimpulan yang didapatkan dari perhitungan metode Borda di atas, peneliti dan pengambil keputusan memutuskan bahwa metode Borda cocok

digunakan dalam proses pengambilan keputusan secara kelompok karena dapat mengakomodasi semua penilaian pengambil keputusan dan hasil dari metode Borda sangat akurat karena hasil nilai yang pasti dengan menjumlahkan penilaian semua pengambil keputusan. Berdasarkan hasil penerapan metode sesuai Tabel 3.63, maka metode *Entropy*, VIKOR dan Borda terbukti dapat diterapkan pada proses pengambilan keputusan untuk menentukan lokasi penempatan ATM yang sesuai dengan kriteria pada PT. BRI (Persero) Tbk Cabang Jember.

5.2 Hasil Pembuatan GDSS Penentuan Lokasi Penempatan ATM

Group Decision Support System Penentuan Lokasi Penempatan ATM memiliki tiga hak akses yaitu admin, pengambil keputusan dan nasabah. Admin dapat mengakses beberapa fitur utama yaitu melihat informasi peta, mengelola data kriteria, batasan kriteria, nilai kriteria, kategori informasi, nilai kategori informasi, lokasi, *user*, dan analisa DSS serta memiliki fitur pendukung yaitu *login*, *logout*, dan mengubah akun. Pengambil keputusan dapat mengakses fitur melihat informasi peta, biodata dan memroses pengambilan keputusan serta memiliki fitur pendukung yaitu *login*, *logout*, dan mengubah akun. Sedangkan nasabah dapat mengakses fitur melihat informasi peta dan data ATM BRI. Fitur-fitur yang terdapat pada GDSS Penentuan Lokasi Penempatan ATM dijelaskan pada sub-bab di bawah ini.

5.2.1 Fitur *Login*

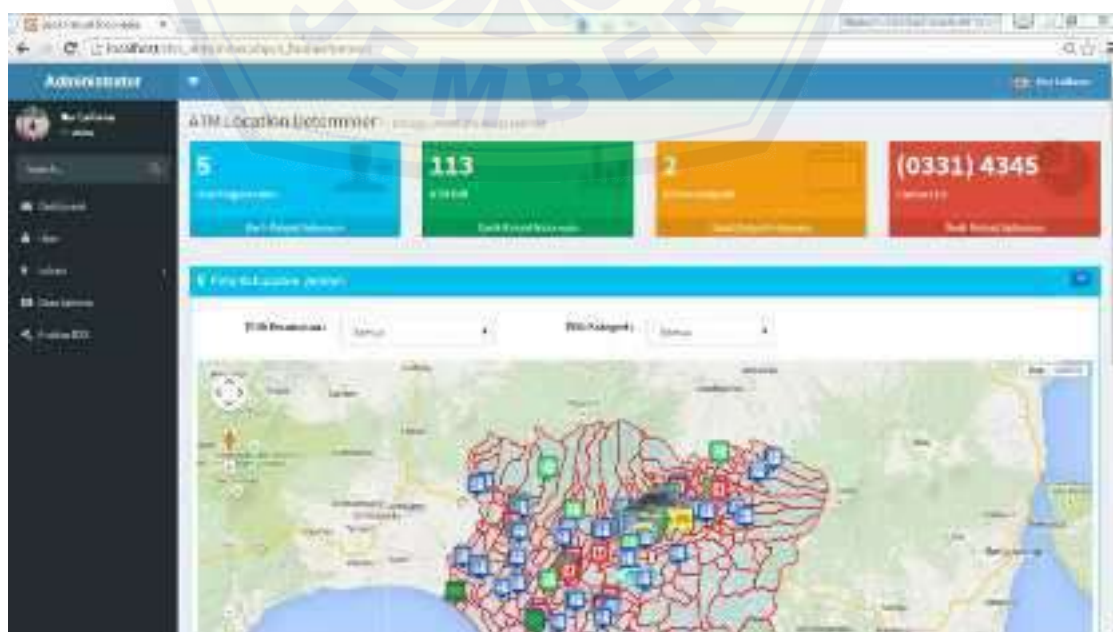
GDSS Penentuan Lokasi Penempatan ATM ini terdiri dari halaman admin, pengambil keputusan dan nasabah. Terdapat fitur keamanan yang harus dilewati oleh admin dan pengambil keputusan untuk memasuki halaman utamanya masing-masing yaitu harus melewati portal *login*. Aktor yang akan dijelaskan adalah admin yang melakukan *login* dengan memasukkan *username* leli dan *password* 12345 pada halaman *login* yang dapat dilihat pada Gambar 5.1.



Gambar 5.1 Tampilan Halaman *Login*

5.2.2 Halaman Depan Admin

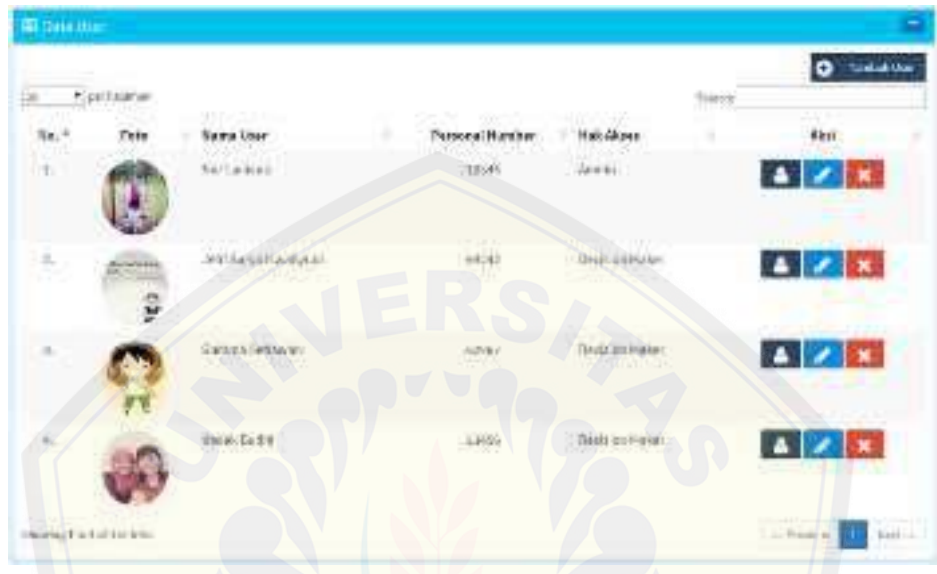
Admin yang telah melakukan proses *login* dengan sukses akan dapat melihat tampilan halaman depan admin. Terdapat beberapa menu pada halaman depan admin yaitu menu *dashboard*, *user*, lokasi, data kriteria dan analisa DSS dan juga terdapat peta persebaran ATM BRI dan kategori lainnya di Kabupaten Jember. Tampilan halaman depan admin dapat dilihat pada Gambar 5.2.



Gambar 5.2 Tampilan Halaman Depan Admin

5.2.3 Fitur Mengelola Data User

Admin dapat mengelola data *user* dengan cara memilih menu *user*. Admin dapat menambah, mengubah, menghapus dan melihat detail data *user* pada fitur mengelola data *user* ini. Tampilan dari halaman fitur mengelola data *user* dapat dilihat pada Gambar 5.3.



Gambar 5.3 Tampilan Fitur Mengelola Data User

5.2.4 Fitur Mengelola Data Kecamatan

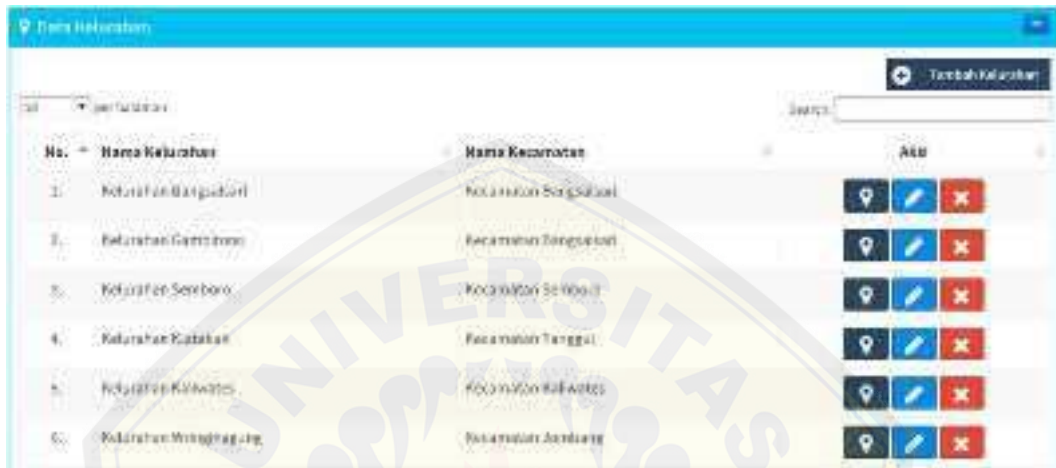
Admin dapat mengelola data kecamatan dengan cara memilih menu lokasi dan sub menu data kecamatan. Admin dapat menambah, mengubah, menghapus dan melihat detail data kecamatan pada fitur mengelola data kecamatan ini. Tampilan dari halaman fitur mengelola data kecamatan dapat dilihat pada Gambar 5.4.



Gambar 5.4 Tampilan Fitur Mengelola Data Kecamatan

5.2.5 Fitur Mengelola Data Kelurahan

Admin dapat mengelola data kecamatan dengan cara memilih menu lokasi dan sub menu data kelurahan. Admin dapat menambah, mengubah, menghapus dan melihat detail data kelurahan pada fitur mengelola data kelurahan ini. Tampilan dari halaman fitur mengelola data kelurahan dapat dilihat pada Gambar 5.5.



Gambar 5.5 Tampilan Fitur Mengelola Data Kelurahan

5.2.6 Fitur Mengelola Data Kriteria

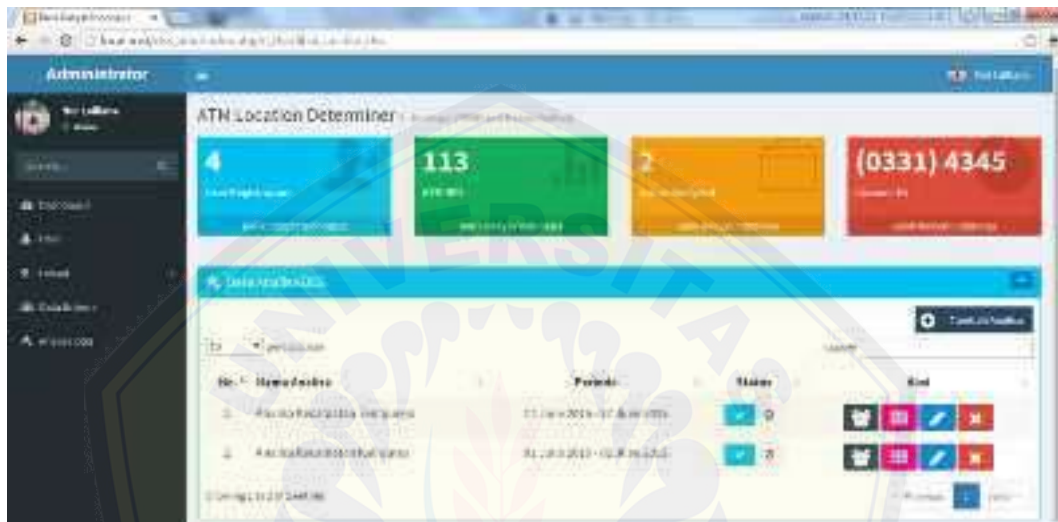
Admin dapat mengelola data kriteria dengan cara memilih menu data kriteria. Admin dapat menambah, mengubah dan menghapus data kriteria pada fitur mengelola data kriteria ini. Tampilan dari halaman fitur mengelola data kriteria dapat dilihat pada Gambar 5.6.



Gambar 5.6 Tampilan Fitur Mengelola Data Kriteria

5.2.7 Fitur Mengelola Analisa DSS

Admin dapat mengelola analisa DSS dengan cara memilih menu analisa DSS. Admin dapat menambah, mengubah, menghapus dan melihat detail analisa DSS pada fitur mengelola analisa DSS ini. Admin juga dapat melihat hasil keputusan kelompok dan mengubah status analisa DSS. Tampilan dari halaman fitur mengelola analisa DSS dapat dilihat pada Gambar 5.7.



Gambar 5.7 Tampilan Fitur Mengelola Analisa DSS

Fitur yang dapat dilakukan oleh admin seperti fitur mengelola batasan kriteria, nilai kriteria, kategori informasi, nilai kategori informasi, melihat informasi peta, *logout* dan mengubah akun dapat dilihat pada Lampiran F (Hasil Pembuatan Sistem). Pada Lampiran F (Hasil Pembuatan Sistem) juga dijelaskan tampilan fitur yang dapat diakses oleh pengambil keputusan dan nasabah.

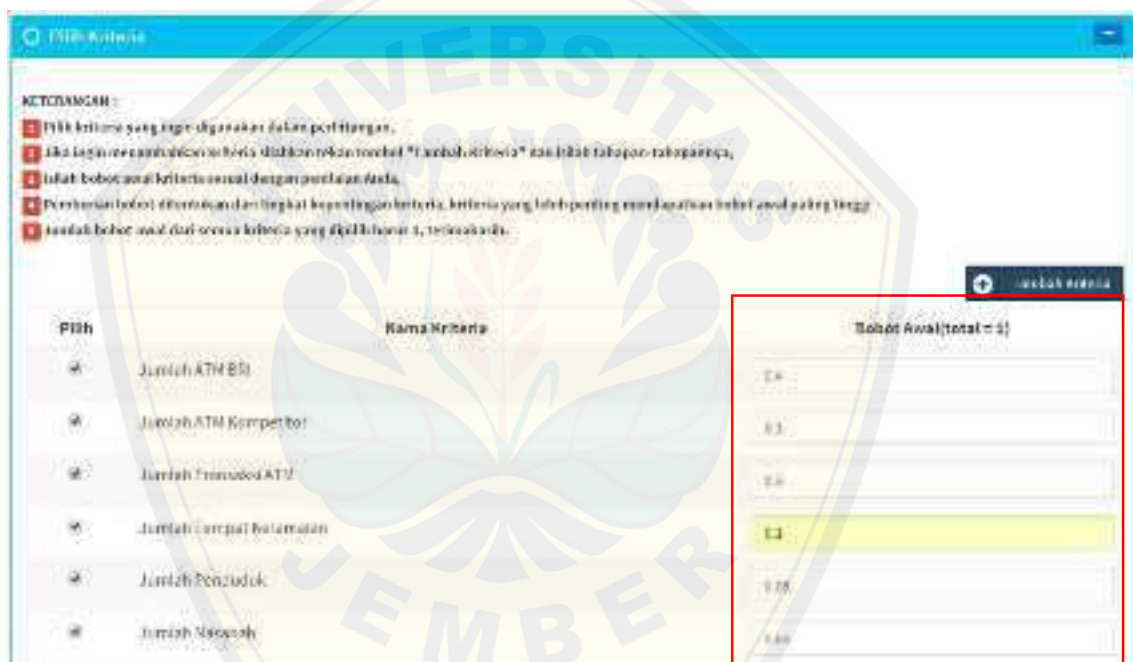
5.3 Hasil Penerapan Metode *Entropy*, VIKOR dan Borda Pada Sistem

Penerapan Metode *Entropy*, VIKOR dan Borda pada GDSS Penentuan Lokasi Penempatan ATM terletak pada fitur memroses pengambilan keputusan yang dapat diakses oleh pengambil keputusan. Tahap pertama penerapan metode-metode tersebut adalah pemilihan kriteria dan pemberian bobot awal kriteria oleh pengambil keputusan. Langkah selanjutnya yaitu melakukan perhitungan bobot kriteria menggunakan metode *Entropy*, melakukan perankingan lokasi menggunakan metode VIKOR dan menentukan keputusan kelompok menggunakan

metode Borda. Penjelasan dari hasil implementasi metode-metode tersebut pada sistem dijelaskan pada sub-bab berikut ini:

5.3.1 Pemilihan Kriteria dan Pemberian Bobot Awal Kriteria

Tahap awal untuk melakukan perhitungan metode *Entropy* adalah pemilihan kriteria dan pemberian bobot awal oleh pengambil keputusan. Total dari bobot awal yang dimasukkan harus sama dengan 1. Tampilan halaman proses pemilihan kriteria dan pemberian bobot awal kriteria oleh Pelaksana IT dan *E-Channel* untuk analisa Kecamatan Kaliwates dapat dilihat pada Gambar 5.8.



The screenshot shows a web application window titled "Pilih Kriteria". It contains a list of criteria on the left and a table for assigning initial weights on the right. The table has a header "Bobot Awal (total = 1)" and a "Total" row at the bottom. The criteria listed are:

Pilih	Nama Kriteria
<input type="checkbox"/>	Jumlah ATM BRI
<input type="checkbox"/>	Jumlah ATM Kompetitor
<input type="checkbox"/>	Jumlah Promosi ATM
<input type="checkbox"/>	Jumlah Lempai Batasaman
<input type="checkbox"/>	Jumlah Penduduk
<input type="checkbox"/>	Jumlah Mesekeh

The weight assignment table on the right is as follows:

Bobot Awal (total = 1)	
0.1	
0.2	
0.3	
0.4	
0.5	
0.6	
0.7	
0.8	
0.9	
1.0	
Total	

Gambar 5.8 Tampilan Pemilihan Kriteria dan dan Pemberian Bobot Awal Kriteria

Total bobot awal yang diberikan oleh pengambil keputusan pada Gambar 5.8 adalah 1. Kriteria yang mempunyai bobot awal tinggi berarti kriteria tersebut dianggap sangat penting dalam proses pengambilan keputusan. Bobot awal dan kriteria tersebut akan disimpan ke dalam *database* ketika pengambil keputusan menekan tombol "Pembobotan *Entropy*" pada Gambar 5.8 menggunakan kode program seperti pada Tabel 4.5 yaitu pada *function* proses1_tambah_bobot_awal(). Bobot awal dan kriteria yang sudah tersimpan akan menjadi masukan untuk perhitungan bobot kriteria menggunakan metode *Entropy*.

5.3.2 Perhitungan Bobot Kriteria Menggunakan Metode *Entropy*

Masukan yang digunakan untuk perhitungan bobot kriteria adalah kriteria dan bobot awal yang sudah tersimpan di *database*. Tahap awal perhitungan metode *Entropy* adalah pembuatan metrik kriteria. Sebelum membuat metrik kriteria, nilai kepentingan dari setiap kelurahan sesuai dengan kriteria yang dipilih disimpan ke dalam metrik 2 dimensi agar dapat ditampilkan dalam bentuk metrik kriteria. Penyimpanan nilai kepentingan ke dalam metrik 2 dimensi menggunakan kode program pada baris 441-455 yang terletak pada *function* *metode_entropy()* pada *class c_dss* yang dapat dilihat pada Tabel 5.1.

Tabel 5.1 Kode Program Penyimpanan Nilai Kepentingan ke Dalam Metrik 2 Dimensi

441.	<code>\$kel = "";</code>
442.	<code>\$des = -1;</code>
443.	<code>\$krit = 0;</code>
444.	<code>foreach (\$ambildata as \$row) {</code>
445.	<code> if (\$kel != \$row['nama_kelurahan']) {</code>
446.	<code> \$krit = 0;</code>
447.	<code> \$des++;</code>
448.	<code> \$kel = \$row['nama_kelurahan'];</code>
449.	<code> \$nilai[\$des][\$krit] = \$row['nilai_kepentingan'];</code>
450.	<code> \$krit++;</code>
451.	<code> } else {</code>
452.	<code> \$nilai[\$des][\$krit] = \$row['nilai_kepentingan'];</code>
453.	<code> \$krit++;</code>
454.	<code> }</code>
455.	<code>}</code>

Langkah selanjutnya adalah membuat metrik kriteria. Metrik kriteria berisi nilai kepentingan pada suatu kelurahan berdasarkan suatu kriteria. Pembuatan metrik kriteria menggunakan kode program baris 457-461 yang terletak pada *function* *metode_entropy()* di *class c_dss* dapat dilihat pada Tabel 5.2.

Tabel 5.2 Kode Program Pembuatan Metrik Kriteria

457.	<code>for (\$j = 0; \$j < \$jumkriteria; \$j++) {</code>
458.	<code> for (\$i = 0; \$i < \$jumdesa; \$i++) {</code>
459.	<code> \$metrik[\$i][\$j] = \$nilai[\$i][\$j];</code>
460.	<code> }</code>
461.	<code>}</code>

Nilai kepentingan diulang sebanyak jumlah kelurahan dan jumlah kriteria yang ada sehingga nilai tersebut dapat masuk ke dalam metrik (0,0), (0,1) dan seterusnya pada Tabel 5.2. Tampilan hasil metrik kriteria pada sistem sesuai dengan kriteria yang dipilih oleh pengambil keputusan pada Gambar 5.8 dapat dilihat pada Gambar 5.9.

Keurahan	Jumlah GTM BRI	Jumlah GTM Koperasi	Jumlah Barabab	Jumlah Perdukahan	Jumlah Tempat Koneksi	Jumlah Transaksi ATM
Kecamatan (Kecamatan) Desa	4	4	1	2	1	1
Kecamatan (Kecamatan) Desa	4	4	3	2	1	1
Kecamatan (Kecamatan) Desa	3	2	2	2	1	1
Kecamatan (Kecamatan) Desa	4	4	1	1	1	1
Kecamatan (Kecamatan) Desa	3	3	2	4	1	1
Kecamatan (Kecamatan) Desa	2	2	4	2	1	1
Kecamatan (Kecamatan) Desa	3	3	2	4	1	1

Gambar 5.9 Tampilan Metrik Kriteria Pada Sistem

Metrik kriteria yang sudah terbentuk akan dinormalisasi agar diskriminasi data akibat sifat kriteria yang berbeda dan kesenjangan nilai kepentingan dapat teratasi. Normalisasi metrik kriteria menggunakan rumus [3.1] dan [3.2]. Kode program untuk membuat metrik kriteria ternormalisasi yang terletak pada *function* metode *entropy()* di *class c_dss* dapat dilihat pada Tabel 5.3.

Tabel 5.3 Kode Program Pembuatan Metrik Kriteria Ternormalisasi

```

462. //mencari nilai maksimal per kriteria
463. for ($j = 0; $j < $jumkriteria; $j++) {
464.     for ($i = 0; $i < $jumdesa; $i++) {
465.         $temp[$i] = $metrik[$i][$j];
466.     }
467.     $nilaimax [$j] = max($temp);
468.     unset($temp);
469. }
470. //mencari nilai minimal per kriteria
471. for ($j = 0; $j < $jumkriteria; $j++) {
472.     for ($i = 0; $i < $jumdesa; $i++) {
473.         $temp[$i] = $metrik[$i][$j];

```

```

474.     }
475.     $nilaimin [$j] = min($temp);
476.     unset($temp);
477.   }
478.   //2. MEMBUAT METRIK TERNORMALISASI
479.   for ($j = 0; $j < $jumkriteria; $j++) {
480.     for ($i = 0; $i < $jumdesa; $i++) {
481.       if ($sifat[$j] == '1') {
482.         $normalisasi[$i][$j] = $metrik[$i][$j] / $nilaimax [$j]; //benefit
483.       } else {
484.         $normalisasi[$i][$j] = $nilaimin [$j] / $metrik[$i][$j]; //cost
485.       }
486.     }
487.   }

```

Pencarian nilai maksimal dan nilai minimal tiap kriteria dilakukan pada Tabel 5.3 , kemudian jika sifat kriteria sama dengan 1 (*benefit*) akan dimasukkan ke dalam baris program ke 482 dan sebaliknya jika *cost* akan dimasukkan ke dalam baris program 484. Nilai normalisasi tersebut diulang sebanyak jumlah kelurahan dan jumlah kriteria sehingga menghasilkan metrik kriteria ternormalisasi. Tampilan hasil metrik kriteria ternormalisasi pada sistem dapat dilihat pada Gambar 5.10.

Kelurahan	Jumlah ATM BNI	Jumlah ATM Kompetitor	Jumlah Minibank	Jumlah Persekiok	Jumlah Tempat Karyawan	Jumlah Transaksi ATM
Kecamatan Jember Barat	0,25	1	0,25	0,5	0,333333333333333	0,5
Kecamatan Kalawati	0,25	1	0,75	0,75	0,333333333333333	1
Kecamatan Kebon Agung	1	0,25	0,5	0,5	0,333333333333333	0,5
Kecamatan Kepethan	0,25	1	0,75	0,5	1	1
Kecamatan Mangli	0,5	0,25	0,75	1	0,333333333333333	0,5
Kecamatan Semarati	0,5	0,75	1	0,5	1	1
Kecamatan Pagelaran	0,000000000000000	0,75	0,5	1	0,333333333333333	1

Gambar 5.10 Tampilan Metrik Kriteria Ternormalisasi Pada Sistem

Langkah selanjutnya yaitu membuat metrik probabilitas kriteria. Nilai probabilitas kriteria dihitung menggunakan rumus [3.3]. Kode program untuk membuat metrik probabilitas kriteria yang terletak pada *function* *metode_entropy()* di *class* *c_dss* dapat dilihat pada Tabel 5.4.

Tabel 5.4 Kode Program Pembuatan Metrik Probabilitas Kriteria

```

489. //mencari jumlah nilai ternormalisasi per kriteria
490. for ($j = 0; $j < $jumkriteria; $j++) {
491.     $simpan = 0;
492.     for ($i = 0; $i < $jumdesa; $i++) {
493.         $simpan = $simpan + $normalisasi[$i][$j];
494.     }
495.     $jum[$j] = $simpan;
496. }
497. //3. MEMBUAT METRIK PROBABILITAS
498. for ($j = 0; $j < $jumkriteria; $j++) {
499.     for ($i = 0; $i < $jumdesa; $i++) {
500.         $probabilitas[$i][$j] = $normalisasi[$i][$j] / $jum[$j];
501.     }
502. }
    
```

Pencarian jumlah nilai yang sudah dinormalisasi tiap kriteria dilakukan pada baris program 490-496 pada Tabel 5.4, kemudian sistem membuat metrik probabilitas kriteria dengan cara membagi nilai normalisasi dengan jumlah nilai normalisasi lalu diulang sebanyak jumlah kelurahan dan jumlah kriteria pada baris program 498-501. Tampilan hasil metrik probabilitas kriteria pada sistem dapat dilihat pada Gambar 5.11.

Kelurahan	Jumlah ATM BBK	Jumlah ATM Kompetitor	Jumlah Nasabah	Jumlah Penduduk	Jumlah Tempat Keramaian	Jumlah Transaksi ATM
Kecamatan Jember Kidul	0.0000000000000000	0.0	0.0000000000000000	0.0	0.0000000000000000	0.0000000000000000
Kecamatan Kalayun	0.0000000000000000	0.0	0.0000000000000000	0.0	0.0000000000000000	0.0000000000000000
Kecamatan Kecamatan Agung	0.0000000000000000	0.0	0.0000000000000000	0.0	0.0000000000000000	0.0000000000000000
Kecamatan Kapatihan	0.0000000000000000	0.0	0.0000000000000000	0.0	0.0000000000000000	0.0000000000000000
Kecamatan Mangli	0.0000000000000000	0.0	0.0000000000000000	0.0	0.0000000000000000	0.0000000000000000
Kecamatan Sertaman	0.0000000000000000	0.0	0.0000000000000000	0.0	0.0000000000000000	0.0000000000000000
Kecamatan Jember Kota	0.0000000000000000	0.0	0.0000000000000000	0.0	0.0000000000000000	0.0000000000000000

Gambar 5.11 Tampilan Metrik Probabilitas Kriteria Pada Sistem

Langkah selanjutnya yaitu membuat metrik natural logaritma probabilitas kriteria. Kode program untuk membuat metrik natural logaritma probabilitas

kriteria yang terletak pada *function* metode *_entropy()* di *class c_dss* dapat dilihat pada Tabel 5.5.

Tabel 5.5 Kode Program Pembuatan Metrik Natural Logaritma Probabilitas Kriteria

```

504.   for ($j = 0; $j < $jumkriteria; $j++) {
505.       for ($i = 0; $i < $jumdesa; $i++) {$lenprobabilitas[$i][$j] =
506.   $probabilitas[$i][$j] * log($probabilitas[$i][$j], M_E);
507.   }
508.   }
    
```

Perhitungan membuat metrik natural logaritma probabilitas kriteria dengan cara mengalikan nilai probabilitas kriteria dengan LN nilai probabilitas lalu dilakukan perulangan sebanyak jumlah kelurahan dan kriteria dapat dilihat pada Tabel 5.5. Tampilan hasil metrik natural logaritma probabilitas kriteria pada sistem dapat dilihat pada Gambar 5.12.

Kelurahan	Jumlah ATM BRI	Jumlah ATM Komptel	Jumlah Razzab	Jumlah Ppduduk	Jumlah Tempat Karamalan	Jumlah Transaksi ATM
Kelurahan Jember	-0.2057114999986	0.32126798248982	-1.1857920875211	-0.230258932094	-0.21738519872781	-0.21799031534331
Klatihan Klaten	-0.2057114999986	0.32126798248982	-1.1857920875211	-0.2345020073258	-0.21738519872781	-0.3095541385888
Kelurahan Kebus Agung	-0.30811254200742	-0.1497850138777	-1.34419988414547	-0.230258932094	-0.21738519872781	-0.21799031534331
Kelurahan Kemuning	-0.2057114999986	0.32126798248982	-1.1857920875211	-0.2345020073258	-0.21738519872781	-0.3095541385888
Kelurahan Panggil	-0.25401930540002	-0.1488950126077	-1.21812621031488	-0.2315878248582	-0.21738519872781	-0.21799031534331
Kelurahan Sirtopah	-0.29401930540002	-0.15426798771200	-1.2141239-1181584	-0.2300285032094	-0.21738519872781	-0.3095541385888
Kelurahan Tegol Sewu	-0.24041234348884	-0.23426798771200	-1.24419988414547	-0.2118878248582	-0.21738519872781	-0.3095541385888

Gambar 5.12 Tampilan Metrik Natural Logaritma Probabilitas Kriteria Pada Sistem

Tahap selanjutnya yaitu menghitung jumlah LN probabilitas dan nilai *entropy* setiap kriteria menggunakan rumus [3.4]. Kode program untuk menghitung nilai *entropy* yang terletak pada *function* metode *_entropy()* di *class c_dss* dapat dilihat pada Tabel 5.6.

Tabel 5.6 Kode Program Menghitung Nilai *Entropy*

```

509.   //jumlah len probabilitas per kriteria
    
```

```

510.     for ($j = 0; $j < $jumkriteria; $j++) {
511.         $simpan = 0;
512.         for ($i = 0; $i < $jumdesa; $i++) {
513.             $simpan = $simpan + $lenprobabilitas[$i][$j];
514.         }
515.         $jumlen[$j] = $simpan;
516.     }
517.     //METRIK ENTROPY PER KRITERIA
518.     for ($j = 0; $j < $jumkriteria; $j++) {
519.         $metrikentropy[$j] = ((-1) / log(($jumdesa), M_E)) * $jumlen[$j];
520.     }

```

perhitungan jumlah LN probabilitas per kriteria dilakukan pada pada Tabel 5.6 yaitu pada baris program 510-516 lalu dihitung nilai *entropy* setiap kriteria pada baris program 518-520. Tampilan hasil nilai *entropy* setiap kriteria pada sistem dapat dilihat pada Gambar 5.13.



Kriteria	Jumlah Natural Logaritma Probabilitas	Nilai Entropy
Jumlah ATM BP	-1.8057113564035	0.92690706648378
Jumlah ATM Koperasi	-1.6341719700176	0.94762571411776
Jumlah Rawabah	-1.875507463181E	0.92529328012628
Jumlah Produk	1.1938964885578	0.97638145552888
Jumlah Tempat Perantara	1.8446214789658	0.94794782972287
Jumlah Transaksi ATM	-1.8437682223911	0.97321463342513

Gambar 5.13 Tampilan Nilai *Entropy* Setiap Kriteria Pada Sistem

Tahap selanjutnya yaitu menghitung lamda kriteria dan bobot *entropy* setiap kriteria menggunakan rumus [3.5] dan [3.6]. Kode program untuk perhitungan lamda kriteria dan bobot *entropy* yang terletak pada *function* metode_entropy() di class c_dss dapat dilihat pada Tabel 5.7.

Tabel 5.7 Kode Program Perhitungan Lamda Kriteria dan Bobot *Entropy*

```

522.         //jumlah nilai entropy
523.         for ($j = 0; $j < $jumkriteria; $j++) {
524.             $jumentropy = $jumentropy + $metrikentropy[$j];
525.         }
526.         //hitung lamda
527.         for ($j = 0; $j < $jumkriteria; $j++) {
528.             if(($jumkriteria - $jumentropy)==0){
529.                 $lamda[$j]=0;

```

```

530.     }
531.     else{
532.     $lamda[$j] = (1 / ($jumkriteria - $jumentropy)) * (1 - $metrikentropy[$j]);
533.     }}
534.     //IAMDA KALI BOBOT AWAL PER KRITERIA
535.     for ($j = 0; $j < $jumkriteria; $j++) {
536.         $lamdabobot[$j] = $lamda[$j] * $bobotawal[$j];
537.     }
538.     //JUMLAH LAMDA KALI BOBOT
539.     for ($j = 0; $j < $jumkriteria; $j++) {
540.         $jumlamda = $jumlamda + $lamdabobot[$j];
541.     }
542.     //HASIL BOBOT ENTROPY PER KRITERIA
543.     for ($j = 0; $j < $jumkriteria; $j++) {
544.         if ($lamdabobot[$j] == 0) {
545.             $entropy[$j] = 0;
546.         } else {
547.             $entropy[$j] = $lamdabobot[$j] / $jumlamda;
548.         }
549.     }

```

Perhitungan jumlah nilai *entropy*, lamda kriteria, lamda dikalikan bobot awal kriteria yang sudah tersimpan di *database* pada tahap sebelumnya, jumlah lamda terbobot dan bobot *entropy* setiap kriteria dilakukan pada Tabel 5.7. Tampilan hasil lamda kriteria dan bobot *entropy* setiap kriteria pada sistem dapat dilihat pada Gambar 5.14.

Kriteria	Lamda Kriteria	Bobot Awal	Lamda Terbobot	Bobot Entropy
Jumlah ATM BR	0.2738632720017	0.4	0.10954530511447	0.38762645482046
Jumlah ATM Kompetitor	0.2172495141325	0.1	0.02172495141325	0.11348525818451
Jumlah Nasabah	0.13543310142345	0.05	0.006771655117225	0.034145681875797
Jumlah Penduduk	0.052294411017762	0.01	0.00052294411017762	0.027521070231701
Jumlah Tempat Kerja/mal	0.19758392108955	0.1	0.019758392108955	0.102520935489289
Jumlah Transaksi ATM	0.10155293270978	0.1	0.010155293270978	0.1591160792424

Gambar 5.14 Tampilan Nilai Lamda dan Bobot *Entropy* Pada Sistem

Perhitungan bobot *entropy* merupakan tahap akhir dari metode *Entropy*. Perhitungan metode *Entropy* pada sistem untuk analisa Kecamatan Kaliwates menghasilkan hasil bobot kriteria yang sama dengan perhitungan manual metode *Entropy* yang dijelaskan pada sub-bab Analisis Data ke dalam Metode *Entropy*.

Dari persamaan hasil bobot *entropy* tersebut maka dapat disimpulkan bahwa penerapan metode *Entropy* pada sistem berhasil. Hal tersebut dibuktikan dengan hasil bobot *entropy* setiap kriteria pada penilaian Pelaksana IT dan *E-Channel*, Asisten Manjer Pemasaran Dana dan Pimpinan Bank untuk analisa Kecamatan Kaliwates yang dapat dilihat pada Gambar 5.15, Gambar 5.16 dan Gambar 5.17 sama dengan hasil bobot *entropy* perhitungan manual yaitu pada Tabel 3.31, Tabel 3.32 dan Tabel 3.33.

Bobot *entropy* yang dihasilkan dari perhitungan metode *Entropy* selanjutnya disimpan ke dalam *database* agar dapat digunakan dalam perhitungan perankingan lokasi menggunakan metode VIKOR. Kode program untuk menyimpan bobot *entropy* yang terletak pada *function* metode *entropy()* di *class c_dss* dapat dilihat pada Tabel 5.8.

Kriteria	Bobot Entropy
Jumlah ATM/UBI	0.20796640403943
Jumlah ATM/Repartitor	0.11370200878487
Jumlah Kasabah	0.03914656070767
Jumlah Pecokulak	0.029691122671
Jumlah Tempat Koneksi Data	0.11882904039969
Jumlah Transaksi ATM	0.1295201070484

Bobot Entropy merupakan gabungan dari bobot awal pengambil keputusan dan bobot hasil perhitungan data.
 Bobot Entropy menggambarkan seberapa penting dalam proses pengambilan keputusan.

Gambar 5.15 Hasil Pembobotan *Entropy* Pelaksana IT dan *E-Channel* Pada Sistem

Kriteria	Bobot Entropy
Jumlah Kasabah yang Punya Kartu ATM	0.14000790000000
Jumlah Tempat Koneksi Data	0.16000000000000
Jumlah Transaksi ATM	0.07000000000000

Bobot Entropy merupakan gabungan dari bobot awal pengambil keputusan dan bobot hasil perhitungan data.
 Bobot Entropy menggambarkan seberapa penting dalam proses pengambilan keputusan.

Gambar 5.16 Hasil Pembobotan *Entropy* Asisten Manjer Pemasaran Dana Pada Sistem

Kriteria	Bobot Entropy
Jumlah Instansi yang Bekerja Sama	0.3025005071219
Jumlah Tempat Karyawan	0.3116580021271
Jumlah Titik/ATM	0.3117937122820

Gambar 5.17 Hasil Pembobotan *Entropy* Pimpinan Bank Pada Sistem

Tabel 5.8 Kode Program Menyimpan Bobot *Entropy* ke *Database*

```

551.     for ($j = 0; $j < $jumkriteria; $j++) {
552.         $bobot_entropy = $entropy[$j];
553.         $id_kriteria = $id_krit[$j];
554.         $query = $this->m_dss->simpan_bobot_entropy($id_transaksi,
555. $id_user, $id_kriteria, $bobot_entropy);
556.     }
    
```

Hasil pembobotan *Entropy* yang dihasilkan pada Gambar 5.15 menunjukkan bahwa Jumlah ATM BRI mempunyai bobot paling banyak sehingga kriteria jumlah ATM BRI sangat berpengaruh atau lebih diutamakan dalam proses pengambilan keputusan yang dilakukan oleh Pelaksana IT dan *E-Channel*. Begitu juga pada Gambar 5.16 , kriteria yang mempunyai bobot *entropy* paling tinggi yaitu kriteria jumlah transaksi ATM merupakan kriteria yang paling berpengaruh dalam proses pengambilan keputusan yang dilakukan oleh Asisten Manajer Pemasaran Dana. Sedangkan pada Gambar 5.17 kriteria yang mempunyai bobot *entropy* paling tinggi yaitu jumlah instansi yang bekerjasama merupakan kriteria yang paling berpengaruh dalam proses pengambilan keputusan yang dilakukan oleh Pimpinan Bank.

5.3.3 Perhitungan Perankingan Lokasi Menggunakan Metode VIKOR

Masukan yang digunakan untuk perankingan lokasi menggunakan metode VIKOR adalah bobot *entropy* yang sudah disimpan di *database*. Perhitungan awal adalah menentukan nilai solusi ideal positif dan negatif untuk setiap kriteria dari metrik normalisasi kriteria seperti pada Gambar 5.10. Penentuan nilai solusi ideal positif (f_i^*) dan nilai solusi ideal negatif (f_i^-) menggunakan rumus [3.7] dan [3.8]. Kode program untuk menentukan f_i^* dan f_i^- yang terletak *function* metode_vikor() di *class c_dss* dapat dilihat pada Tabel 5.9.

Tabel 5.9 Kode Program Menghitung Nilai Solusi Ideal Positif dan Negatif Setiap Kriteria

```

650. //1.mencari nilai ideal positif dan negatif setiap kriteria dari metrik normalisasi
651. //ideal positif
652. for ($j = 0; $j < $jumkriteria; $j++) {
653.     for ($i = 0; $i < $jumdesa; $i++) {
654.         $temp[$i] = $normalisasi[$i][$j];
655.     }
656.     $idealpositif [$j] = max($temp);
657.     unset($temp);
658. }
659. //ideal negatif
660. for ($j = 0; $j < $jumkriteria; $j++) {
661.     for ($i = 0; $i < $jumdesa; $i++) {
662.         $temp[$i] = $normalisasi[$i][$j];
663.     }
664.     $idealnegasif [$j] = min($temp);
665.     unset($temp);
666. }

```

Nilai ideal positif dihitung menggunakan kode program baris 652-658 dengan mencari nilai maksimal dari nilai normalisasi pada Tabel 5.9. Sedangkan nilai solusi ideal negatif dihitung menggunakan kode program baris 660-665 dengan mencari nilai minimal dari nilai normalisasi. Nilai normalisasi disimpan ke dalam variabel temp lalu diulang sebanyak jumlah kelurahan dan kriteria kemudian dicari nilai minimal dan maksimalnya. Tampilan hasil perhitungan nilai solusi ideal positif dan negatif kriteria pada analisa Kecamatan Kaliwates oleh Pelaksana IT dan *E-Channel* pada sistem dapat dilihat pada Gambar 5.18.

Kriteria	Nilai Solusi Ideal Positif	Nilai Solusi Ideal Negatif
Jumlah ATM (SD)	5	0.25
Jumlah ATM Kompetitor	1	0.25
Jumlah Saluran	5	0.25
Jumlah Fasilitas	1	0.5
Jumlah Tempat Penjualan	5	0.25
Jumlah Penyalur ATM	1	0.5

Gambar 5.18 Tampilan Nilai Solusi Ideal Positif dan Negatif Pada Sistem

Langkah selanjutnya adalah menghitung nilai solusi tertinggi (S_j) dan nilai solusi terendah (R_j) untuk setiap alternatif berdasarkan metrik kriteria ternormalisasi. Bobot *entropy* yang telah dihasilkan pada perhitungan metode *Entropy* akan dimasukkan ke dalam rumus [3.9] dan [3.10]. Kode program untuk menentukan nilai S_j dan R_j yang terletak pada *function* metode_vikor() di *class* *c_dss* dapat dilihat pada Tabel 5.10.

Tabel 5.10 Kode Program Menghitung Nilai Solusi Tertinggi dan Terendah Setiap Alternatif

```

667. //2.mencari nilai solusi tertinggi dan terendah setiap alternatif
668. //nilai solusi ideal terbobot entropy
669. $bagi = array();
670. for ($j = 0; $j < $jumkriteria; $j++) {
671.     for ($i = 0; $i < $jumdesa; $i++) {
672.         $bagi[$j] = ($idealpositif [$j] - $idealnegasif [$j]);
673.         if ($bagi[$j] == '0') {
674.             $idealbobot[$i][$j] = 0;
675.         } else {
676.             $idealbobot[$i][$j] = $entropy[$j] * (($idealpositif [$j] -
677. $normalisasi[$i][$j]) / $bagi[$j]);
678.         }
679.     }
680. }
681. //S = Nilai solusi tertinggi, jumlah ideal terbobot
682. for ($i = 0; $i < $jumdesa; $i++) {
683.     $simpan = 0;
684.     for ($j = 0; $j < $jumkriteria; $j++) {
685.         $simpan = $simpan + $idealbobot[$i][$j];
686.     }
687.     $solusiteringgi[$i] = $simpan;
688. }
689. //R = Nilai solusi terendah, max ideal terbobot
690. for ($i = 0; $i < $jumdesa; $i++) {

```


691.	for (\$j = 0; \$j < \$jumkriteria; \$j++) {
692.	\$temp[\$j] = \$idealterbobot[\$i][\$j];
693.	}
694.	\$solusiterendah [\$i] = max(\$temp);
695.	unset(\$temp);
696.	}

Langkah pertama dalam menentukan nilai S_j dan R_j adalah dengan menghitung nilai solusi ideal terbobot yang disimpan ke dalam variabel *idealterbobot* pada Tabel 5.10. Perhitungan nilai S_j adalah menghitung jumlah dari nilai solusi ideal terbobot pada kode baris program 682-688. Nilai R_j dihitung dengan mencari nilai maksimal dari nilai solusi ideal terbobot pada kode baris program 690-696.

Perhitungan metode VIKOR selanjutnya adalah menghitung indeks VIKOR/rasio perankingan. Nilai rasio perankingan (Q_j) dapat dihitung menggunakan rumus [3.11]. Kode program untuk menentukan nilai Q_j yang terletak pada *function* metode *vikor()* di *class c_dss* dapat dilihat pada Tabel 5.11.

Tabel 5.11 Kode Program Menghitung Indeks VIKOR

697.	//menghitung nilai Smax
698.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
699.	\$temp[\$i] = \$solusitertinggi[\$i];
700.	}
701.	\$smax = max(\$temp);
702.	unset(\$temp);
703.	//menghitung nilai Smin
704.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
705.	\$temp[\$i] = \$solusitertinggi[\$i];
706.	}
707.	\$smin = min(\$temp);
708.	unset(\$temp);
709.	//menghitung nilai Rmax
710.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
711.	\$temp[\$i] = \$solusiterendah[\$i];
712.	}
713.	\$rmax = max(\$temp);
714.	unset(\$temp);
715.	//menghitung nilai Rmin
716.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
717.	\$temp[\$i] = \$solusiterendah[\$i];
718.	}
719.	\$rmin = min(\$temp);
720.	unset(\$temp);

```

721.
722.   $kurangs = $smax - $smin;
723.   $kurangr = $rmax - $rmin;
724.   //menghitung nilai Q per alternatif
725.   $v = 0.5;
726.   for ($i = 0; $i < $jumdesa; $i++) {
727.     if ($kurangs == '0') {
728.       $hasil1 = 0;
729.     } else {
730.       $hasil1 = ($v * (($solusiteringgi[$i] - $smin) / ($kurangs)));
731.     }
732.     if ($kurangr == '0') {
733.       $hasil2 = 0;
734.     } else {
735.       $hasil2 = ((1 - $v) * (($solusiterendah[$i] - $rmin) / ($kurangr)));
736.     }
737.     $rasio[$i] = $hasil1 + $hasil2;
738.   }
    
```

Tahap awal untuk menentukan nilai indeks VIKOR setiap alternatif adalah dengan menghitung nilai maksimal S dan R serta nilai minimal S dan R dengan menggunakan baris program 698-720 pada Tabel 5.11. Nilai maksimal S dikurangkan dengan nilai minimal S lalu disimpan ke dalam variabel kurangs begitu juga dengan nilai R akan disimpan ke dalam variabel kurangr. Nilai Q dihitung menggunakan baris program 726-738 yaitu dengan menjumlahkan variabel hasil1 dan hasil2 diulang sebanyak jumlah alernatif yang ada. Tampilan hasil perhitungan nilai S, R dan Q pada sistem dapat dilihat pada Gambar 5.19.

Alternatif	Nilai S	Nilai R	Nilai Q
Kabupaten Jember 4211	0.88921484321386	0.36748870462049	1
Kabupaten Kalibanteng	0.84190118034123	0.36748870462049	0.00000000000000
Kabupaten Kebonagung	0.82791165159439	0.1995187792084	0
Kabupaten Popandayan	0.5835800495026	0.0188949421048	0.00150891760308
Kabupaten Punggi	0.7047002223991	0.07845703854099	0.00001871486207
Kabupaten Sampang	0.43732026071210	0.07845703854099	0.00048804009137
Kabupaten Tegal Jawa	0.81087000000000	0.01481010000000	0.00000000000000

Gambar 5.19 Tampilan Hasil Perhitungan Nilai S, R dan Q Pada Sistem

Tahap selanjutnya pada perhitungan metode VIKOR yaitu mengurutkan nilai S, R dan Q secara *decreasing order*. Nilai S,R dan Q diurutkan dari nilai paling

kecil sampai nilai paling besar. Kode program untuk mengurutkan nilai S, R dan Q secara *decreasing order* yang terletak pada *function* metode_vikor() di *class c_dss* dapat dilihat pada Tabel 5.12.

Tabel 5.12 Kode Program Mengurutkan Nilai S,R dan Q dengan *Decreasing Order*

754.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
755.	\$urut [\$i] = array(
756.	'alternatif' => \$desa[\$i],
757.	'ns' => \$solusiteringgi[\$i]
758.);
759.	}
760.	\$b = array();
761.	\$c = array();
762.	foreach (\$urut as \$key => \$value) {
763.	\$b[\$key] = \$value['ns'];
764.	}
765.	natsort(\$b);
766.	foreach (\$b as \$key => \$value) {
767.	\$c[] = \$urut[\$key];
768.	}
769.	foreach (\$c as \$row) {
770.	\$nilais[] = \$row['ns'];
771.	\$desas[] = \$row['alternatif'];
772.	}
773.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
774.	\$rank [\$i] = array(
775.	'nilais' => \$nilais[\$i],
776.	'desas' => \$desas[\$i]
777.);
778.	}
779.	
780.	//mengurutkan Nilai R
781.	\$urutr = array();
782.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
783.	\$urutr [\$i] = array(
784.	'alternatif' => \$desa[\$i],
785.	'nr' => \$solusiterendah[\$i]
786.);
787.	}
788.	\$b = array();
789.	\$c = array();
790.	foreach (\$urutr as \$key => \$value) {
791.	\$b[\$key] = \$value['nr'];
792.	}
793.	natsort(\$b);
794.	foreach (\$b as \$key => \$value) {
795.	\$c[] = \$urutr[\$key];
796.	}
797.	foreach (\$c as \$row) {

```

798.     $nilair[] = $row['nr'];
799.     $desar[] = $row['alternatif'];
800. }
801. for ($i = 0; $i < $jumdesa; $i++) {
802.     $rankr [$i] = array(
803.         'nilair' => $nilair[$i],
804.         'desar' => $desar[$i]
805.     );
806. }
807.
808. //mengurutkan Nilai Q
809. $urutq = array();
810. for ($i = 0; $i < $jumdesa; $i++) {
811.     $urutq [$i] = array(
812.         'alternatif' => $desa[$i],
813.         'nq' => $rasio[$i]
814.     );
815. }
816. $bq = array();
817. $cq = array();
818. $rankq = array();
819. foreach ($urutq as $key => $value) {
820.     $bq[$key] = $value['nq'];
821. }
822. natsort($bq);
823. foreach ($bq as $key => $value) {
824.     $cq[] = $urutq[$key];
825. }
826. foreach ($cq as $row) {
827.     $nilaiq[] = $row['nq'];
828.     $desaq[] = $row['alternatif'];
829. }
830. for ($i = 0; $i < $jumdesa; $i++) {
831.     $rankq [$i] = array(
832.         'nilaiq' => $nilaiq[$i],
833.         'desaq' => $desaq[$i]
834.     );
835. }

```

Pengurutan nilai S menggunakan baris program 754-788 pada Tabel 5.12. Cara mengurutkan nilai dengan *decreasing order* adalah dengan menggunakan fungsi *natsort()*. Pengurutan nilai R menggunakan baris program 781-806 sedangkan pengurutan nilai Q menggunakan baris program 809-835. Tampilan hasil pengurutan nilai S, R dan Q pada sistem dapat dilihat pada Gambar 5.20.

Tahap perhitungan metode VIKOR selanjutnya adalah memeriksa kondisi penentuan pemberian ranking kelurahan berdasarkan penilaian masing-masing pengambil keputusan. Kode program untuk memeriksa kondisi penentuan pemberian ranking yang terletak pada *function* metode_vikor() di *class c_dss* dapat dilihat pada Tabel 5.13.

The image shows a screenshot of a software application interface with three data tables. A large watermark for 'UNIVERSITAS JEMBER' is overlaid on the center. The tables are as follows:

Desking Order 1	
Kelurahan	Nilai S
Kelurahan Kebon Agung	0.42082651920408
Kelurahan Sempukan	0.24879219171238
Kelurahan Kepatihan	0.00001832400428
Kelurahan Tegalselar	0.6188729626028
Kelurahan Kaliwates	0.94100198190122
Kelurahan Hangi	0.79647015513699
Kelurahan Jember Kidul	0.00001484121854

Desking Order 2	
Kelurahan	Nilai R
Kelurahan Kebon Agung	0.199840778204
Kelurahan Hangi	0.27040783554008
Kelurahan Sempukan	0.076478940099
Kelurahan Tegalselar	0.82461715220008
Kelurahan Kepatihan	0.9976684532028
Kelurahan Kaliwates	0.8170704540042
Kelurahan Jember Kidul	0.0076684532028

Desking Order 3	
Kelurahan	Nilai Q
Kelurahan Kebon Agung	0
Kelurahan Sempukan	0.084880419787
Kelurahan Tegalselar	0.031890100328
Kelurahan Hangi	0.68484971184887
Kelurahan Kepatihan	0.0018800798132
Kelurahan Kaliwates	0.7170276980708
Kelurahan Jember Kidul	0

Gambar 5.20 Tampilan Urutan Nilai S, R dan Q Pada Sistem

Tabel 5.13 Kode Program Pemeriksaan Kondisi Pemberian Ranking Kelurahan

837.	//5.CEK KONDISI 1 DAN 2
838.	//- Hitung DQ
839.	\$dq = 1 / (\$jumdesa - 1);
840.	//KONDISI 1
841.	\$hasil = array();
842.	if (\$nilaiq[1] - \$nilaiq[0] > \$dq \$nilaiq[1] - \$nilaiq[0] == \$dq) {
843.	//KONDISI 2
844.	if (\$desaq[0] == \$desar[0] && \$desaq[0] == \$desas[0]) {
845.	//nilainya yg ranking 1 tetep
846.	\$d = 0;
847.	\$bantu = 0;
848.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
849.	\$d++;
850.	if (\$i == 0) {
851.	\$hasil [\$i] = array(
852.	'nilaiq' => \$nilaiq[\$i],
853.	'desaq' => \$desaq[\$i],
854.	'rankingq' => \$d
855.);
856.	}
857.	if (\$i > 0) {
858.	if (\$nilaiq[\$i] == \$nilaiq[\$i - 1]) {
859.	\$d--;
860.	\$bantu++;
861.	\$hasil [\$i] = array(
862.	'nilaiq' => \$nilaiq[\$i],
863.	'desaq' => \$desaq[\$i],
864.	'rankingq' => \$d
865.);
866.	} else {
867.	\$d = \$d + \$bantu;
868.	\$bantu = 0;
869.	\$hasil [\$i] = array(
870.	'nilaiq' => \$nilaiq[\$i],
871.	'desaq' => \$desaq[\$i],
872.	'rankingq' => \$d
873.);
874.	}
875.	}
876.	}
877.	} else {
878.	//nilai ke 1 dan 2 jadi ranking 1
879.	\$m = 0;
880.	\$bantu1 = 0;
881.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
882.	if (\$i == 0 \$i == 1) {
883.	\$hasil [\$i] = array(
884.	'nilaiq' => \$nilaiq[\$i],
885.	'desaq' => \$desaq[\$i],

```
886.         'rankingq' => 1
887.     );
888. }
889. $m++;
890. if ($i > 1) {
891.     if ($nilaiq[$i] == $nilaiq[$i - 1]) {
892.         $m--;
893.         $bantu1++;
894.         $hasil [$i] = array(
895.             'nilaiq' => $nilaiq[$i],
896.             'desaq' => $desaq[$i],
897.             'rankingq' => $m
898.         );
899.     } else {
900.         $m = $m + $bantu1;
901.         $bantu1 = 0;
902.         $hasil [$i] = array(
903.             'nilaiq' => $nilaiq[$i],
904.             'desaq' => $desaq[$i],
905.             'rankingq' => $m
906.         );
907.     }
908. }
909. }
910. }
911. } else {
912.     $o = 0;
913.     $g = 1;
914.     $ban = 0;
915.     for ($i = 0; $i < $jumdesa; $i++) {
916.         if ($nilaiq[$i] - $nilaiq[0] < $dq || $nilaiq[$i] - $nilaiq[0] == $dq) {
917.             $hasil [$i] = array(
918.                 'nilaiq' => $nilaiq[$i],
919.                 'desaq' => $desaq[$i],
920.                 'rankingq' => $g
921.             );
922.             $ban++;
923.         } else {
924.             $o++;
925.             if ($i > 0) {
926.                 if ($nilaiq[$i] == $nilaiq[$i - 1]) {
927.                     $o--;
928.                     $ban++;
929.                     $hasil [$i] = array(
930.                         'nilaiq' => $nilaiq[$i],
931.                         'desaq' => $desaq[$i],
932.                         'rankingq' => $o
933.                     );
934.                 } else {
935.                     $o = $o + $ban;
```

936.		\$ban = 0;
937.		\$hasil [\$i] = array(
938.		'nilaiq' => \$nilaiq[\$i],
939.		'desaq' => \$desaq[\$i],
940.		'rankingq' => \$o
941.);
942.		}
943.		}
944.		}
945.		}
946.		}
947.		}

Pemeriksaan kondisi pertama yaitu $Q(j^{(2)}) - Q(j^{(1)}) \geq DQ$ dan kondisi kedua yaitu alternatif $j^{(1)}$ harus menjadi ranking yang terbaik pada S_j dan R_j dihitung menggunakan kode program pada Tabel 5.13. Setelah pemeriksaan kondisi selesai dilakukan, maka setiap kelurahan mendapatkan ranking sesuai dengan urutan nilai S, R dan Q yang memenuhi syarat kondisi 1 dan 2.

Pemberian ranking kelurahan merupakan tahap akhir dari metode VIKOR. Perhitungan metode VIKOR pada sistem untuk analisa Kecamatan Kaliwates menghasilkan hasil perankingan kelurahan yang sama dengan perhitungan manual metode VIKOR yang dijelaskan pada sub-bab Analisis Data ke dalam Metode VIKOR. Dari persamaan hasil perankingan kelurahan tersebut maka dapat disimpulkan bahwa penerapan metode VIKOR pada sistem berhasil. Hal tersebut dibuktikan dengan hasil perankingan kelurahan pada penilaian Pelaksana IT dan E-Channel, Asisten Manjer Pemasaran Dana dan Pimpinan Bank untuk analisa Kecamatan Kaliwates yang dapat dilihat pada Gambar 5.21, Gambar 5.22 dan Gambar 5.23 sama dengan hasil perankingan kelurahan melalui perhitungan manual yaitu pada Tabel 3.55, Tabel 3.57 dan Tabel 3.59.

Ranking kelurahan yang dihasilkan dari perhitungan metode VIKOR selanjutnya disimpan ke dalam *database* agar dapat digunakan dalam perhitungan perankingan lokasi oleh kelompok menggunakan metode Borda. Kemudian sistem mengubah status pengambil keputusan di *database* menjadi sudah melakukan analisa. Kode program untuk menyimpan hasil perankingan lokasi menggunakan metode VIKOR yang terletak pada *function* `metode_vikor()` di *class* `c_dss` dapat dilihat pada Tabel 5.14.

Kelurahan	Ranking
Kelurahan Kebon Agung	1
Kelurahan Sempuaji	2
Kelurahan Tegul Besar	3
Kelurahan Mangli	4
Kelurahan Kaulihan	5
Kelurahan Kalwates	7
Kelurahan Jember Kidul	8

Gambar 5.21 Hasil Perankingan Lokasi Pelaksana IT dan E-Channel Pada Sistem

Kelurahan	Ranking
Kelurahan Sempuaji	1
Kelurahan Kaputihan	2
Kelurahan Tegul Besar	3
Kelurahan Kalwates	4
Kelurahan Mangli	5
Kelurahan Jember Kidul	6
Kelurahan Kebon Agung	8

Gambar 5.22 Hasil Perankingan Lokasi Asisten Manajer Pemasaran Dana Pada Sistem

Kelurahan	Ranking
Kelurahan Tegul Besar	1
Kelurahan Kalwates	2
Kelurahan Sempuaji	3
Kelurahan Kaputihan	4
Kelurahan Kebon Agung	5
Kelurahan Jember Kidul	6
Kelurahan Mangli	8

Gambar 5.23 Hasil Perankingan Lokasi Pimpinan Bank Pada Sistem

Tabel 5.14 Kode Program Menyimpan Hasil Perankingan Lokasi ke *Database*

959.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
960.	\$iddel = \$id[\$i];
961.	\$ranking = \$rankingfix[\$i];
962.	\$query = \$this->m_dss->simpan_vikor(\$id_transaksi, \$id_user,
963.	\$iddel, \$ranking);
964.	}
965.	\$query1 = \$this->m_dss->ubah_status_dm(\$id_transaksi,
966.	\$id_user);

Hasil perankingan VIKOR berdasarkan penilaian Pelaksana IT dan *E-Channel* pada Gambar 5.21 menunjukkan bahwa kelurahan yang paling direkomendasikan untuk dibangun ATM adalah Kelurahan Kebun Agung. Hasil perankingan VIKOR berdasarkan penilaian Asisten Manajer Pemasaran Dana pada Gambar 5.22 menunjukkan bahwa kelurahan yang paling direkomendasikan untuk dibangun ATM adalah Kelurahan Sempusari, Kepatihan, Tegal Besar dan Kaliwates. Sedangkan untuk penilaian Pimpinan Bank pada Gambar 5.23 menunjukkan bahwa kelurahan yang paling direkomendasikan adalah Kelurahan Tegal Besar dan Kaliwates. Hasil perhitungan VIKOR tiap pengambil keputusan berbeda-beda karena kriteria yang dipilih, jumlah kriteria dan pemberian bobot awal setiap pengambil keputusan berbeda-beda.

5.3.4 Perhitungan Keputusan Kelompok Menggunakan Metode Borda

Masukan yang digunakan dalam perhitungan metode Borda adalah hasil perankingan lokasi setiap pengambil keputusan yang sudah disimpan di dalam *database*. Hasil perankingan lokasi pada Kecamatan Kaliwates ditampilkan dalam bentuk metrik ke dalam sistem sesuai pada Gambar 5.24.

Kelurahan	Gambar Sebaran	Jember Suro Iswahyudi	Kode Borda
Kecamatan Jemberkidul	8	7	8
Kecamatan Kaliwates	7	6	7
Kecamatan Kebondagung	6	5	6
Kecamatan Kepoharjo	5	4	5
Kecamatan Pangajene	4	3	4
Kecamatan Sampung	3	2	3
Kecamatan Tegay Diner	2	1	2

Gambar 5.24 Tampilan Metrik Ranking Kelurahan Pada Sistem

Tahap awal dalam perhitungan Borda adalah pemberian *point* Borda kepada setiap ranking kelurahan. Kelurahan yang terdapat di Kecamatan Kaliwates berjumlah 7 buah sehingga ranking 1 mendapatkan *point* 7, ranking 2 mendapatkan *point* 6 dan seterusnya sampai ranking 7 mendapatkan *point* 1. Kode program pemberian *point* Borda yang terletak pada *function* metode_borda() di *class* c_dsa dapat dilihat pada Tabel 5.15.

Tabel 5.15 Kode Program Penentuan *Point* Borda

```

1020. $x = 1;
1021. $y = $jumdesa;
1022. $z = 0;
1023. $hasilpoint = array();
1024. for ($j = 0; $j < $jumuser; $j++) {
1025.     for ($i = 0; $i < $jumdesa; $i++) {
1026.         if ($awal[$i][$j] == $y) {
1027.             $hasilpoint[$i][$j] = $x;
1028.         } else {
1029.             for ($z = 1; $z < $jumdesa; $z++) {
1030.                 if ($awal[$i][$j] == ($y - $z)) {
1031.                     $hasilpoint[$i][$j] = $x + $z;
1032.                 }
1033.             }
1034.         }
1035.     }
1036. }
    
```

Cara memberikan *point* Borda pada setiap ranking adalah dengan memeriksa apakah ranking tersebut sama dengan jumlah kelurahan sesuai pada Tabel 5.15.

Apabila sesuai maka akan diberikan *point* x yaitu 1. Apabila ranking tersebut sama dengan jumlah kelurahan dikurangi 1 maka akan diberikan *point* sebanyak $x+z$ yaitu 2 begitu seterusnya sampai perulangan sebanyak jumlah kelurahan dan jumlah desa selesai. Tampilan hasil pemberian *point* Borda untuk analisa Kecamatan Kaliwates pada sistem dapat dilihat pada Gambar 5.25.

Kelurahan	Gamta Sebaran	Jelit Sanyo Iswahyudi	Kodek Bachi
Kelurahan Sumber Kidal	3	4	7
Kelurahan Sidanek	3	2	7
Kelurahan Kebon Agung	4	4	4
Kelurahan Pajitaban	1	3	4
Kelurahan Mangi	3	4	8
Kelurahan Simpatali	4	4	4
Kelurahan Tegal Besa	1	2	4

Gambar 5.25 Tampilan Metrik *Point* Borda Pada Sistem

Tahap perhitungan Borda selanjutnya adalah menghitung Borda *count* untuk masing-masing kelurahan. Borda *count* dihitung dengan menjumlahkan semua *point* yang dimiliki masing-masing kelurahan. Kode program perhitungan Borda *count* terletak pada *function* metode *borda()* di *class* *c_dss* dapat dilihat pada Tabel 5.16.

Tabel 5.16 Kode Program Perhitungan Borda *Count*

1039.	<code>\$jumpoint = array();</code>
1040.	<code>for (\$i = 0; \$i < \$jumdesa; \$i++) {</code>
1041.	<code> \$simpan = 0;</code>
1042.	<code> for (\$j = 0; \$j < \$jumuser; \$j++) {</code>
1043.	<code> \$simpan = \$simpan + \$hasilpoint[\$i][\$j];</code>
1044.	<code> }</code>
1045.	<code> \$jumpoint[\$i] = \$simpan;</code>
1046.	<code>}</code>

Borda *count* yang sudah didapatkan oleh setiap kelurahan kemudian di urutkan dari nilai yang paling banyak sampai nilai paling kecil. Kelurahan dengan Borda *count* yang paling banyak akan menjadi ranking 1 dan seterusnya. Apabila terdapat Borda *count* yang sama maka ranking yang didapatkan kelurahan tersebut

juga sama. Kode program pemberian ranking kelurahan tersebut terletak pada *function* metode_borda() di *class c_dss* dapat dilihat pada Tabel 5.17.

Tabel 5.17 Kode Program Pemberian Ranking Kelurahan Metode Borda

1048.	//4. URUTKAN DARI BESAR KE KECIL
1049.	\$a = 0;
1050.	foreach (\$kelurahan as \$row) {
1051.	\$desa[\$a] = \$row['nama_kelurahan'];
1052.	\$a++;
1053.	}
1054.	\$nilaiborda = array();
1055.	\$desaborda = array();
1056.	\$urut = array();
1057.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
1058.	\$urut [\$i] = array(
1059.	'alternatif' => \$desa[\$i],
1060.	'point' => \$jumpoint[\$i]
1061.);
1062.	}
1063.	\$b = array();
1064.	\$c = array();
1065.	foreach (\$urut as \$key => \$value) {
1066.	\$b[\$key] = \$value['point'];
1067.	}
1068.	arsort(\$b);
1069.	foreach (\$b as \$key => \$value) {
1070.	\$c[] = \$urut[\$key];
1071.	}
1072.	foreach (\$c as \$row) {
1073.	\$nilaiborda[] = \$row['point'];
1074.	\$desaborda[] = \$row['alternatif'];
1075.	}
1076.	//BERIKAN RANKING, NILAI SAMA DAPAT RANKING SAMA
1077.	\$d = 0;
1078.	\$bantu = 0;
1079.	for (\$i = 0; \$i < \$jumdesa; \$i++) {
1080.	\$d++;
1081.	if (\$i == 0) {
1082.	\$hasilborda [\$i] = array(
1083.	'ranking' => \$nilaiborda[\$i],
1084.	'desa' => \$desaborda[\$i],
1085.	'borda' => \$d
1086.);
1087.	}
1088.	if (\$i > 0) {
1089.	if (\$nilaiborda[\$i] == \$nilaiborda[\$i - 1]) {
1090.	\$d--;
1091.	\$bantu++;
1092.	\$hasilborda [\$i] = array(

```

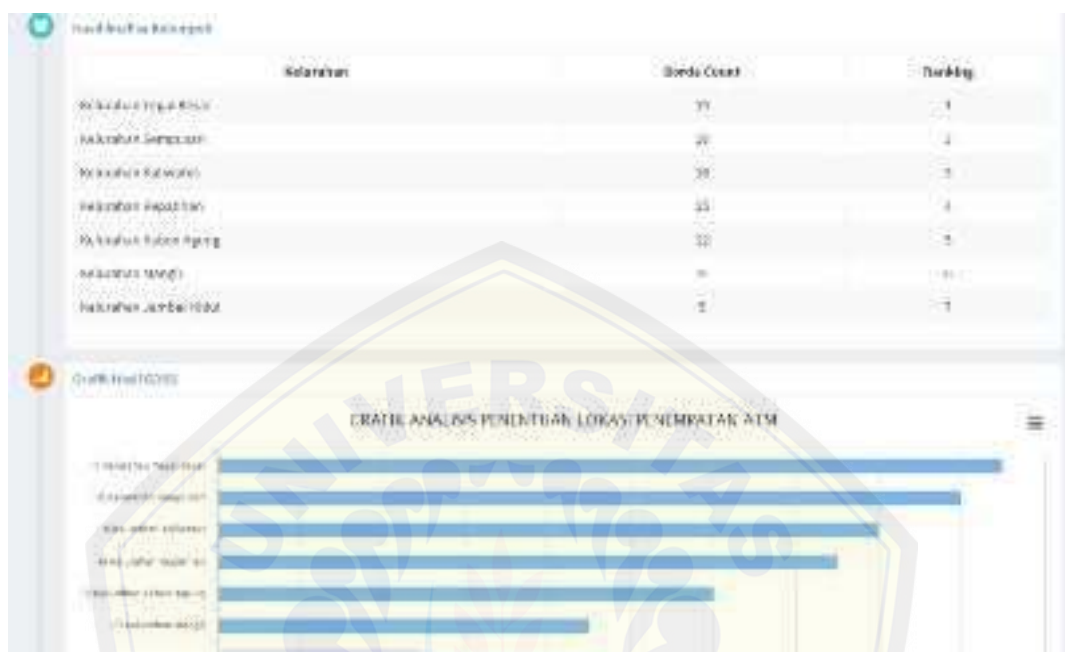
1093.         'ranking' => $nilaiborda[$i],
1094.         'desa' => $desaborda[$i],
1095.         'borda' => $d
1096.     );
1097.     } else {
1098.         $d = $d + $bantu;
1099.         $bantu = 0;
1100.         $hasilborda [$i] = array(
1101.             'ranking' => $nilaiborda[$i],
1102.             'desa' => $desaborda[$i],
1103.             'borda' => $d
1104.         );
1105.     }
1106. }
1107. }
    
```

Hasil dari perankingan lokasi menggunakan metode Borda merupakan hasil keputusan kelompok dalam proses pengambilan keputusan yang dilakukan. Perhitungan metode Borda pada sistem untuk Kecamatan Kaliwates menghasilkan hasil perankingan kelurahan yang sama dengan perhitungan manual metode Borda yang dijelaskan pada sub-bab Analisis Data ke dalam Metode Borda. Dari persamaan hasil perankingan kelurahan tersebut maka dapat disimpulkan bahwa penerapan metode Borda pada sistem berhasil. Hal tersebut dibuktikan dengan hasil perankingan kelurahan hasil keputusan kelompok untuk Kecamatan Kaliwates yang dapat dilihat pada Gambar 5.26 sama dengan hasil perankingan kelurahan melalui perhitungan manual yaitu pada Tabel 3.63. Hasil keputusan kelompok pada Kecamatan Kaliwates menunjukkan bahwa rekomendasi lokasi penempatan ATM adalah pada Kelurahan Tegal Besar karena mendapatkan Borda *count* terbanyak.

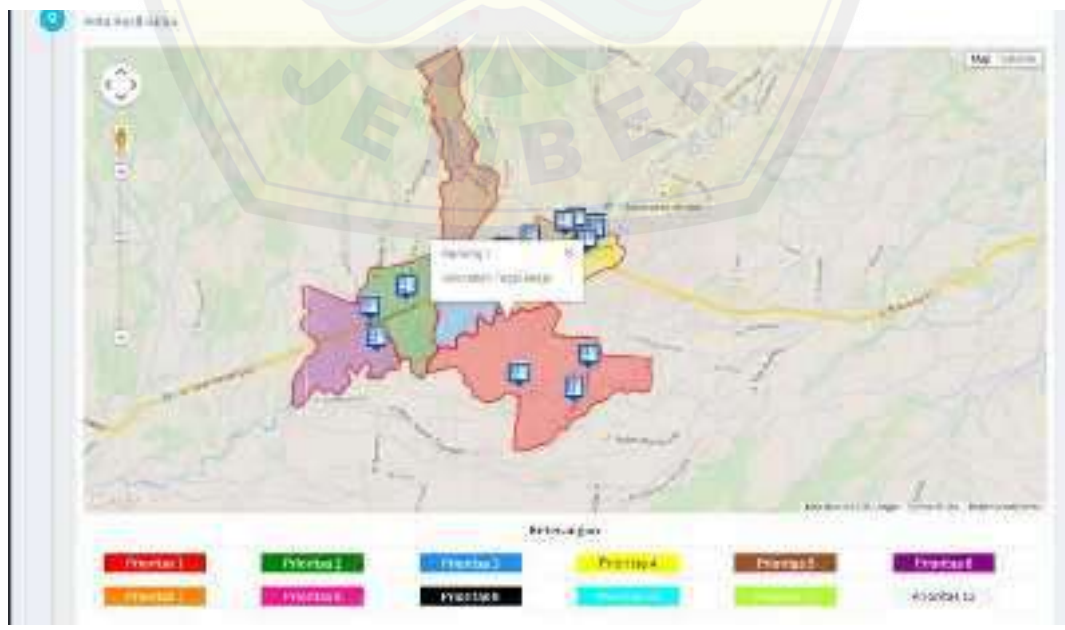
Kelurahan	Borda Count	Ranking
Kelurahan Tegal Besar	27	1
Kelurahan Sittandjaja	24	2
Kelurahan Kaliwates	23	3
Kelurahan Klatihan	20	4
Kelurahan Piten Ageng	17	5
Kelurahan Mangi	9	6
Kelurahan Leber Kidul	9	7

Gambar 5.26 Hasil Keputusan Kelompok Analisa Kecamatan Kaliwates Pada Sistem

Hasil dari keputusan kelompok kemudian ditampilkan ke dalam sistem dalam bentuk tabel, grafik dan peta. Tampilan hasil keputusan kelompok pada sistem dapat dilihat pada Gambar 5.27 dan Gambar 5.28.



Gambar 5.27 Tampilan Hasil Keputusan Kelompok Dalam Bentuk Tabel dan Grafik



Gambar 5.28 Tampilan Hasil Keputusan Kelompok Dalam Bentuk Peta

5.4 Hasil Implementasi Sistem Informasi Geografis Pada Sistem

Implementasi Sistem Informasi Geografis pada GDSS Penentuan Lokasi Penempatan ATM terletak pada fitur melihat informasi peta, melihat hasil analisa DSS dan GDSS, mengelola data kecamatan dan kelurahan serta mengelola nilai kategori informasi. Dengan tampilan berbasis SIG diharapkan dapat mempermudah penggambaran hasil pengambilan keputusan. Implementasi SIG pada penelitian ini menggunakan *Google Maps* dalam menampilkan peta persebaran ATM dan kategori, peta detail kecamatan dan kelurahan, peta detail nilai kategori informasi serta peta hasil analisa DSS dan GDSS. Penelitian ini juga menggunakan *Open Street Maps* dalam proses menambah data kecamatan dan kelurahan serta menambah nilai kategori informasi. Penjelasan tentang implementasi SIG pada penelitian ini akan dijelaskan pada sub-bab berikut ini:

5.4.1 Implementasi *Google Maps* Pada Sistem

Penelitian ini menggunakan *Google Maps* dalam menampilkan peta persebaran ATM dan kategori, peta detail kecamatan dan kelurahan, peta detail nilai kategori informasi serta peta hasil analisa DSS dan GDSS. Tahapan untuk menampilkan *Google Maps* pada fitur melihat informasi peta, fitur melihat detail data kecamatan, data kelurahan, dan nilai kategori informasi adalah sama. Perbedaan terletak hanya pada data yang diambil dari *database* yang akan ditampilkan ke dalam peta. Sehingga peneliti hanya akan menjelaskan tahapan menampilkan *Google Maps* pada fitur melihat informasi peta dan melihat hasil GDSS.

1. Fitur Melihat Informasi Peta

Fitur melihat informasi peta merupakan implementasi SIG yang berfungsi untuk menampilkan persebaran ATM BRI, ATM kompetitor (ATM BCA dan Mandiri), Perumahan, Rumah Sakit dan kategori informasi lainnya di Kabupaten Jember. Data ATM BRI dan kategori lainnya pada masing-masing kelurahan ditampilkan dalam *info window* yang akan muncul apabila pengguna mengklik area atau titik lokasi pada peta. Fitur ini juga dilengkapi dengan pemilihan kecamatan

dan kategori yang digunakan untuk menampilkan data sesuai kecamatan dan kategori yang dipilih. Cara menampilkan peta persebaran ATM BRI dan kategori informasi adalah dengan menggunakan *library Googlemaps* untuk menampilkan *Google Maps* ke dalam sistem. Kode Program untuk menampilkan fitur melihat informasi peta terletak pada *function home()* di *class c_home* dapat dilihat pada Tabel 5.18.

Tabel 5.18 Kode Program Menampilkan Peta Persebaran ATM dan Kategori

12.	<code>public function home(\$type) {</code>
13.	<code> \$this->load->library('googlemaps');</code>
14.	<code> \$config['center'] = '-8.239383846027533, 113.6823472357595';</code>
15.	<code> \$config['zoom'] = 10;</code>
16.	<code> \$this->googlemaps->initialize(\$config);</code>
17.	<code> \$kelurahan = \$this->m_lokasi->data_kelurahan();</code>
18.	<code> \$kat = \$this->m_kriteria->data_kategori();</code>
19.	<code> foreach (\$kelurahan as \$value) {</code>
20.	<code> \$marker2 = array();</code>
21.	<code> \$polygon = str_replace("MULTIPOLYGON", "", \$value->lat);</code>
22.	<code> \$polygon = str_replace(",", "", \$polygon);</code>
23.	<code> \$polygon = str_replace(")", "", \$polygon);</code>
24.	<code> \$polygon = str_replace(" ", "", \$polygon);</code>
25.	<code> \$polygon = str_replace(" ", "", \$polygon);</code>
26.	<code> \$data = explode(";", \$polygon);</code>
27.	<code> \$label = \$value->label;</code>
28.	<code> \$nama = \$value->nama_kelurahan;</code>
29.	<code> \$marker2['fillColor'] = '#87CEEB';</code>
30.	<code> for (\$i = 0; \$i < count(\$data); \$i++) {</code>
31.	<code> \$koordinat = explode(",", \$data[\$i]);</code>
32.	<code> \$marker2['points'][] = \$koordinat[1] . "," . \$koordinat[0];</code>
33.	<code> \$marker2['onclick'] = "</code>
34.	<code> var infowindow = new google.maps.InfoWindow(</code>
35.	<code> {</code>
36.	<code> size: new google.maps.Size(150,50)</code>
37.	<code> });</code>
38.	<code> infowindow.setContent('\$label');</code>
39.	<code> if (event) {</code>
40.	<code> point = event.latLng;</code>
41.	<code> }</code>
42.	<code> infowindow.setPosition(point);</code>
43.	<code> infowindow.open(map);";</code>
44.	<code> }</code>
45.	<code> }</code>
46.	<code> \$this->googlemaps->add_polygon(\$marker2);</code>
47.	<code> }</code>
48.	<code> foreach (\$kat as \$value) {</code>
49.	<code> \$markerSkh = array();</code>
50.	<code> \$icon = \$value->icon;</code>

51.	<code>\$markerSkIh['icon'] = "http://localhost/dss_atm/temp_upload/\$icon";</code>
52.	<code>\$markerSkIh['position'] = "\$value->lat, \$value->long";</code>
53.	<code>\$markerSkIh['infowindow_content'] = "<p>\$value-</code>
54.	<code>>nama</p><p>\$value->alamat</p><p>\$value->info</p>";</code>
55.	<code>\$this->googlemaps->add_marker(\$markerSkIh);</code>
56.	<code>}</code>
57.	<code>\$data1['kecamatan'] = null;</code>
58.	<code>\$data1['kategori'] = null;</code>
59.	<code>\$data1['map'] = \$this->googlemaps->create_map();</code>
60.	<code>\$data1['option1'] = \$this->m_lokasi->combo_kecamatan();</code>
61.	<code>\$data1['kat'] = \$this->m_kriteria->jumlah_kategori();</code>
62.	<code>\$data1['option2'] = \$this->m_kriteria->combo_kategori();</code>
63.	<code>\$data = \$this->session();</code>
64.	<code>if (\$type == '1') {</code>
65.	<code> \$data['content'] = \$this->load->view('admin/home_admin', \$data1,</code>
66.	<code>true);</code>
67.	<code> \$this->load->view('layout/template', \$data);</code>
68.	<code> } else {</code>
69.	<code> \$data['content'] = \$this->load->view('dm/home_dm', \$data1, true);</code>
70.	<code> \$this->load->view('layout/template', \$data);</code>
71.	<code> } }</code>

Tahapan untuk menampilkan peta dengan menggunakan *library Googlemaps* adalah menghubungkan sistem dengan *library Googlemaps* dengan menggunakan baris kode program ke 13 pada Tabel 5.18. Langkah selanjutnya adalah mengatur letak koordinat lokasi yang akan pertama kali ditampilkan ke dalam peta menggunakan kode program baris ke 14. Data kelurahan dan kategori yang ingin ditampilkan diambil dari *database* pada baris kode program 17 -18. Langkah selanjutnya adalah membuat marker peta, mengatur warna marker, mengatur *info window* dan mengubah *geom* pada data kelurahan menjadi tampilan *polygon* pada peta nanti. Kemudian data kategori yang sudah diambil dari *database* dimasukkan ke dalam marker peta. Data kategori tersebut akan ditampilkan ke dalam titik pada marker yang mempunyai *icon* yang berbeda-beda. Setelah konfigurasi *googlemaps* selesai dilakukan maka sistem menampilkan halaman depan dari masing-masing aktor. Tampilan fitur melihat informasi peta pada sistem menggunakan *Google Maps* dapat dilihat pada Gambar 5.29.

2. Fitur Melihat Hasil GDSS

Fitur melihat hasil GDSS merupakan implementasi SIG yang berfungsi untuk menampilkan hasil pengambilan keputusan kelompok untuk suatu kecamatan ke dalam sebuah peta. Peta hasil GDSS menampilkan area kelurahan pada kecamatan yang dianalisa dengan warna yang berbeda sesuai dengan ranking yang diperoleh dari kelurahan tersebut. Persebaran ATM BRI yang ada di kelurahan tersebut juga ditampilkan ke dalam peta. Tujuan dari penggambaran kelurahan dengan warna yang berbeda dan persebaran ATM BRI yang ada di kelurahan tersebut adalah untuk memudahkan pengguna dalam mengetahui letak kelurahan yang direkomendasikan menjadi lokasi penempatan ATM dan mengetahui lokasi mana saja yang belum ditempatkan sebuah ATM pada kelurahan tersebut. Pada fitur melihat hasil gdss ini, area kelurahan yang ditampilkan dibagi menjadi 12 prioritas dengan warna yang berbeda-beda. Prioritas pertama merupakan lokasi yang paling direkomendasikan menjadi lokasi penempatan ATM baru, prioritas kedua merupakan lokasi selanjutnya yang direkomendasikan dan begitu seterusnya. Cara menampilkan peta hasil GDSS adalah dengan menggunakan *library Googlemaps* untuk menampilkan *Google Maps* ke dalam sistem. Potongan kode program untuk menampilkan fitur melihat hasil GDSS terletak pada *function* `lihat_hasil_gdss()` di *class* `c_dss` dapat dilihat pada Tabel 5.19.

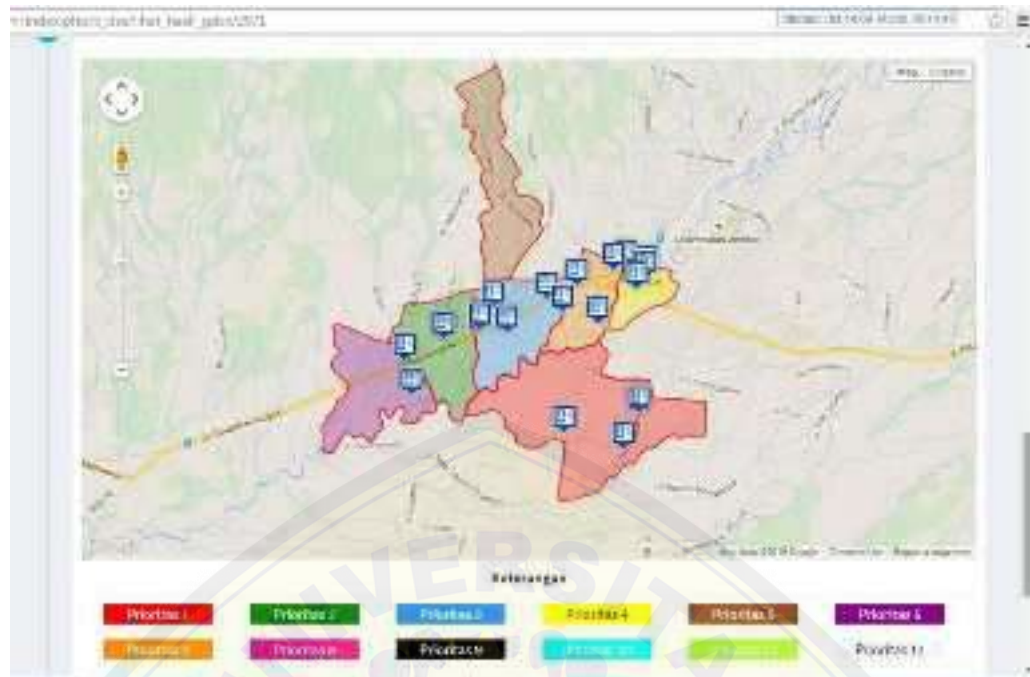
Tabel 5.19 Kode Program Fitur Melihat Hasil GDSS

283.	<code>\$this->load->library('googlemaps');</code>
284.	<code>\$config['center'] = '-8.239383846027533, 113.6823472357595';</code>
285.	<code>\$config['zoom'] = 10;</code>
286.	<code>\$this->googlemaps->initialize(\$config);</code>
287.	<code>\$kelurahan = \$this->m_dss->peta_gdss(\$id_transaksi);</code>
288.	<code>\$kat = \$this->m_dss->data_atm(\$id_transaksi);</code>
289.	<code>foreach (\$kelurahan as \$value) {</code>
290.	<code> \$marker2 = array();</code>
291.	<code> \$polygon = str_replace("MULTIPOLYGON", "", \$value->lat);</code>
292.	<code> \$polygon = str_replace("(", "", \$polygon);</code>
293.	<code> \$polygon = str_replace(")", "", \$polygon);</code>
294.	<code> \$polygon = str_replace(",", ";", \$polygon);</code>
295.	<code> \$polygon = str_replace(" ", "", \$polygon);</code>
296.	<code> \$data = explode(";", \$polygon);</code>
297.	<code> \$label = \$value->label;</code>
298.	<code> \$nama = \$value->nama_kelurahan;</code>
299.	<code> \$ranking = \$value->ranking;</code>

```
300.     if ($ranking == '1') {
301.         $marker2['fillColor'] = '#FF0000';
302.     }
303.     if ($ranking == '2') {
304.         $marker2['fillColor'] = '#008000';
305.     }
306.     if ($ranking == '3') {
307.         $marker2['fillColor'] = '#1E90FF';
308.     }
309.     if ($ranking == '4') {
310.         $marker2['fillColor'] = '#FFFF00';
311.     }
312.     if ($ranking == '5') {
313.         $marker2['fillColor'] = '#A0522D';
314.     }
315.     if ($ranking == '6') {
316.         $marker2['fillColor'] = '#8B008B';
317.     }
318.     if ($ranking == '7') {
319.         $marker2['fillColor'] = '#FF8C00';
320.     }
321.     if ($ranking == '8') {
322.         $marker2['fillColor'] = '#FF1493';
323.     }
324.     if ($ranking == '9') {
325.         $marker2['fillColor'] = '#000000';
326.     }
327.     if ($ranking == '10') {
328.         $marker2['fillColor'] = '#00FFFF';
329.     }
330.     if ($ranking == '11') {
331.         $marker2['fillColor'] = '#ADFF2F';
332.     }
333.     if ($ranking == '12') {
334.         $marker2['fillColor'] = '#F5F5F5';
335.     }
336.     for ($i = 0; $i < count($data); $i++) {
337.         $koordinat = explode(" ", $data[$i]);
338.         $marker2['points'][] = $koordinat[1] . ", " . $koordinat[0];
339.         $marker2['onclick'] = "
340.             var infowindow = new google.maps.InfoWindow(
341.
342.                 {
343.                     size: new google.maps.Size(150,50)
344.                 });
345.             infowindow.setContent('<p>Ranking
346. $ranking.</p><p>$nama</p>');
347.             if (event) {
348.                 point = event.latLng;
349.             }
```

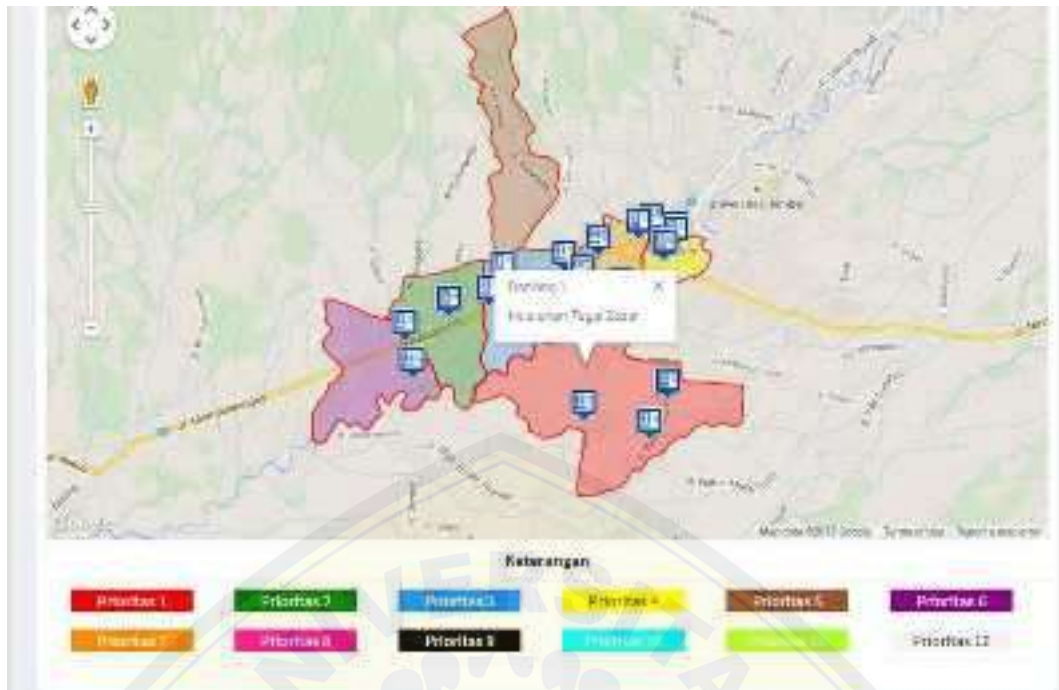
350.	infowindow.setPosition(point);
351.	infowindow.open(map);";
352.	}
353.	\$this->googlemaps->add_polygon(\$marker2);
354.	}
355.	foreach (\$kat as \$value) {
356.	\$markerSkIh = array();
357.	\$icon = \$value->icon;
358.	\$markerSkIh['icon'] = "http://localhost/dss_atm/temp_upload/\$icon";
359.	\$markerSkIh['position'] = "\$value->lat, \$value->long";
360.	\$markerSkIh['infowindow_content'] = "<p>\$value-
361.	>nama</p><p>\$value->alamat</p><p>\$value->info</p>";
362.	\$this->googlemaps->add_marker(\$markerSkIh);
363.	}

Tahapan untuk menampilkan peta dengan menggunakan *library Googlemaps* adalah menghubungkan sistem dengan *library Googlemaps* dengan menggunakan baris kode program ke 283 pada Tabel 5.19. Langkah selanjutnya adalah mengatur letak koordinat lokasi yang akan pertama kali ditampilkan ke dalam peta menggunakan kode program baris ke 284. Data hasil GDSS dan data ATM BRI yang ingin ditampilkan diambil dari *database* pada baris kode program 287-288. Langkah selanjutnya adalah membuat marker peta, mengatur warna marker, mengatur *info window* dan mengubah *geom* pada data kelurahan menjadi tampilan *polygon* pada peta nanti. Kemudian data ATM BRI yang sudah diambil dari *database* dimasukkan ke dalam marker peta. Data ATM BRI tersebut akan ditampilkan ke dalam titik pada marker. Pengaturan warna area kelurahan menggunakan kode program baris ke 300-335. Warna area kelurahan berbeda-beda sesuai dengan ranking kelurahan yang dihasilkan dari perhitungan metode Borda. Setelah konfigurasi *googlemaps* selesai dilakukan maka sistem menampilkan halaman hasil analisa GDSS. Tampilan fitur melihat hasil GDSS menggunakan *Google Maps* pada sistem dapat dilihat pada Gambar 5.31.



Gambar 5.31 Tampilan Fitur Melihat Hasil GDSS

Terdapat 7 kelurahan yang ada di Kecamatan Kaliwates yang dibagi menjadi 7 warna area berbeda pada Gambar 5.31. Area kelurahan yang mendapat ranking pertama diwarnai dengan warna merah, area kelurahan yang mendapat ranking kedua diwarnai dengan warna hijau, area kelurahan yang mendapat ranking ketiga diwarnai dengan warna biru tua, area kelurahan yang mendapat ranking keempat diwarnai dengan warna kuning, area kelurahan yang mendapat ranking kelima diwarnai dengan warna coklat, area kelurahan yang mendapat ranking keenam diwarnai dengan warna ungu, dan area kelurahan yang mendapat ranking ketujuh diwarnai dengan warna *orange*. Terdapat persebaran ATM BRI di setiap kelurahan yang ditampilkan. Aktor dapat melihat informasi lain dari kelurahan dan ATM dengan cara menekan area atau titik lokasi pada peta sehingga sistem akan menampilkan *info window* tentang kelurahan atau ATM yang dipilih. Tampilan *info window* kelurahan pada fitur melihat hasil GDSS dapat dilihat pada Gambar 5.32.



Gambar 5.32 Tampilan *Info Window* Kelurahan Pada Fitur Melihat Hasil GDSS

Berdasarkan implementasi *Google Maps* pada berbagai fitur yang dimiliki oleh sistem, *Google Maps* dapat digunakan di dalam sistem ini untuk menampilkan data dalam bentuk peta karena memiliki visualisasi yang jelas dan menarik. Visualisasi pada *Google Maps* dapat memudahkan pengguna melihat penampakan lokasi pada permukaan bumi.

5.4.2 Implementasi *Open Street Maps* Pada Sistem

Penelitian ini menggunakan *Open Street Maps* dalam fitur menambah data kecamatan dan kelurahan serta menambah nilai kategori informasi. *Open Street Maps* digunakan dalam proses menambahkan data spasial karena terdapat fungsi untuk memudahkan pengguna menggambar sebuah peta pada komputer sehingga pengguna dapat dengan mudah menambahkan titik, garis maupun area ke dalam peta dengan cara menggambarinya langsung pada peta.

Tahapan untuk menampilkan *Open Street Maps* (OSM) dalam fitur menambah data kecamatan adalah menambahkan *javascript* OSM ke dalam kode program tampilan sistem lalu memanggil *function* yang ada di *javascript* tersebut.

Javascript yang dipanggil yaitu *OpenLayers.js* dan *GeoJSON.js*. Tampilan OSM pada fitur menambah data kecamatan dapat dilihat pada Gambar 5.33.



Gambar 5.33 Tampilan *Open Street Map* Pada Fitur Menambah Data Kecamatan

Aktor dapat menambahkan area kecamatan dengan memilih tanda *tool polygon* pada pojok kanan atas peta. Kemudian aktor menggambar area kecamatan yang baru sesuai dengan keinginannya. Apabila aktor sudah selesai menggambar area, maka aktor klik dua kali pada peta lalu sistem menjalankan *function serialize(feature)* pada *javascript* yang ada pada tampilan sistem. *Function serialize(feature)* akan menghasilkan tipe gambar dan hasil koordinat gambar yang akan ditampilkan pada *field* Hasil Drawing seperti pada Gambar 5.34.



Gambar 5.34 Tampilan Tipe dan Koordinat Gambar Menggunakan OSM

Tahapan untuk menampilkan OSM dalam fitur menambah data kelurahan sama dengan tahapan pada fitur menambah data kecamatan. Sedangkan tahapan untuk menampilkan OSM dalam fitur menambah nilai kategori informasi peta hampir sama dengan tahapan pada fitur menambah data kecamatan akan tetapi jika pada fitur menambah data kecamatan aktor memilih tanda *tool polygon* untuk menggambar area, pada fitur menambah nilai kategori informasi peta aktor harus memilih *tool point* untuk menggambar satu titik lokasi.

Berdasarkan implementasi OSM pada berbagai fitur yang dimiliki oleh sistem, OSM dapat digunakan di dalam sistem ini untuk menambahkan data spasial dengan langkah-langkah yang mudah yaitu menggambar langsung titik, area maupun garis pada peta. Langkah-langkah mendapatkan koordinat lokasi pada OSM juga tidak terlalu rumit karena hanya memanggil *javascript* yang disediakan oleh OSM.

5.5 Hasil Perbandingan Proses Pengambilan Keputusan Secara Konvensional dan Menggunakan Sistem

GDSS Penentuan Lokasi Penempatan ATM diharapkan dapat memudahkan para pengambil keputusan di PT. BRI (Persero) Tbk Cabang Jember untuk mendapatkan hasil keputusan kelompok dalam menentukan lokasi penempatan ATM dan memberikan rekomendasi lokasi penempatan ATM yang sesuai dengan kriteria pengambil keputusan. Harapan tersebut akan terpenuhi apabila sistem dapat memberikan rekomendasi lokasi penempatan ATM sesuai dengan kriteria dan proses pengambilan keputusan dengan menggunakan sistem lebih mudah diselesaikan daripada proses pengambilan keputusan yang lampau sebelum menggunakan sistem (konvensional). Tingkat kemudahan proses pengambilan keputusan menggunakan sistem dapat dilihat dari hasil pengujian sistem oleh para pengambil keputusan.

Proses pengambilan keputusan kelompok untuk menentukan lokasi penempatan ATM baru pada PT. BRI (Persero) Tbk Cabang Jember sebelum menggunakan sistem mengalami beberapa kesulitan. Kesulitan-kesulitan yang dihasilkan pada proses pengambilan keputusan secara konvensional diharapkan

dapat diatasi dengan menggunakan GDSS Penentuan Lokasi Penempatan ATM. Kesulitan-kesulitan tersebut antara lain:

1. Mengumpulkan para pengambil keputusan untuk mendapatkan hasil keputusan bersama karena setiap pengambil keputusan mempunyai kesibukan masing-masing sehingga sulit untuk berkumpul di dalam satu tempat membahas penentuan lokasi penempatan ATM.
2. Penilaian dari masing-masing pengambil keputusan berbeda-beda.
3. Penentuan lokasi penempatan ATM didasarkan dari penilaian subjektif dengan sebagian besar hanya melihat kriteria yang dianggap paling penting pada proses pengambilan keputusan.
4. Membutuhkan waktu yang lama untuk mendapatkan hasil keputusan kelompok karena kesulitan yang dijelaskan pada poin 1 dan kesulitan dalam menggabungkan penilaian dari masing-masing pengambil keputusan menjadi keputusan kelompok.

Proses pengambilan keputusan kelompok untuk menentukan lokasi penempatan ATM baru pada PT. BRI (Persero) Tbk Cabang Jember dengan menggunakan GDSS Penentuan Lokasi Penempatan ATM menghasilkan beberapa kemudahan yang dirasakan oleh para pengambil keputusan antara lain :

1. Meminimalisir waktu yang digunakan untuk menentukan lokasi penempatan ATM yang sesuai dengan kriteria.
2. Meningkatkan kualitas keputusan karena proses perhitungan pengambilan keputusan menggunakan metode *Entropy*, VIKOR dan Borda sehingga menghasilkan hasil keputusan yang lebih objektif.
3. Menemukan cara yang tepat untuk menggabungkan penilaian para pengambil keputusan yang berbeda-beda dengan menggunakan metode *voting* yaitu metode Borda
4. Setiap pengambil keputusan dapat mengkalkulasikan keputusan secara *online* dari manapun dan dapat dilakukan pada saat yang bersamaan maupun tidak bersamaan.

Berdasarkan kemudahan-kemudahan yang dirasakan oleh pengambil keputusan tersebut maka dapat disimpulkan bahwa GDSS Penentuan Lokasi

Penempatan ATM dapat memberikan kemudahan dalam proses pengambilan keputusan untuk menentukan lokasi penempatan ATM yang sesuai dengan kriteria daripada proses pengambilan keputusan secara konvensional. Akan tetapi hasil dari lokasi penempatan ATM yang direkomendasikan oleh sistem belum tentu akan dipakai oleh para pengambil keputusan mengingat GDSS Penentuan Lokasi Penempatan ATM ini hanyalah sistem pendukung keputusan yang memberikan rekomendasi lokasi sesuai perhitungan yang dilakukan. Keputusan akhir penentuan lokasi penempatan ATM tetap ditentukan oleh para pengambil keputusan.



BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

1. Metode *Entropy* digunakan untuk pembobotan kriteria, metode VIKOR digunakan untuk perankingan lokasi oleh setiap pengambil keputusan dan metode Borda digunakan untuk perankingan lokasi hasil keputusan kelompok pada proses pengambilan keputusan untuk menentukan lokasi penempatan ATM pada PT. BRI (Persero) Tbk Cabang Jember.
2. *Group Decision Support System* (GDSS) penentuan lokasi penempatan Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember telah dibuat dengan 3 hak akses yaitu admin, pengambil keputusan dan nasabah dengan fitur utama adalah fitur memroses pengambilan keputusan yang dilakukan oleh pengambil keputusan.
3. Penentuan lokasi penempatan ATM pada Kecamatan Kaliwates oleh Pelaksana IT dan *E-Channel*, Asisten Manajer Pemasaran Dana dan Pimpinan PT. BRI (Persero) Tbk Cabang Jember menghasilkan rekomendasi lokasi penempatan ATM berupa ranking kelurahan dengan ranking pertama sampai terakhir adalah Kelurahan Tegal Besar, Sempusari, Kepatihan, Kebon Agung, Kaliwates, dan Jember Kidul
4. Metode *Entropy* mampu memberikan bobot kepada kriteria berdasarkan bobot awal yang diberikan pengambil keputusan dan perhitungan variasi data setiap kriteria. Kriteria dengan variasi nilai dan bobot awal tertinggi akan mendapatkan bobot tertinggi.
5. Metode VIKOR mampu menentukan ranking alternatif dengan mempertimbangkan jarak alternatif dengan nilai solusi ideal positif yaitu

jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap alternatif (S) dan jarak alternatif dengan nilai solusi negatif yaitu seluruh nilai terburuk yang dicapai untuk setiap alternatif (R) serta indeks VIKOR(Q) sebagai acuan dalam perankingan. Kelurahan yang mempunyai nilai S,R dan Q paling kecil akan menjadi ranking terbaik karena mempunyai jarak yang lebih dekat dengan solusi ideal positif.

6. Metode Borda mampu meningkatkan kualitas keputusan dan menggabungkan penilaian para pengambil keputusan yang berbeda-beda menjadi suatu keputusan kelompok. Kelurahan yang sering mendapat ranking terbaik hasil dari penilaian masing-masing pengambil keputusan akan mendapatkan nilai Borda yang lebih banyak daripada kelurahan yang sering mendapat ranking rendah. Kelurahan dengan nilai Borda terbanyak merupakan lokasi penempatan ATM yang paling direkomendasikan.
7. Proses pengambilan keputusan untuk menentukan lokasi penempatan ATM menggunakan GDSS penentuan lokasi penempatan ATM dapat mengatasi kesulitan-kesulitan pada proses pengambilan keputusan secara konvensional di PT. BRI (Persero) Tbk Cabang Jember.

6.2 Saran

Beberapa saran dan masukan berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu :

1. Diperlukan adanya perbandingan sistem yang dibangun dengan metode lain.
2. Membuat tampilan sistem menjadi lebih menarik.

DAFTAR PUSTAKA

- Baskoro, P. S., Ahsan, A. S., & Basofi, A. (2011). Sistem Pengambilan Keputusan Penempatan ATM (Automated Teller Machine) Berdasarkan Penyebaran Nasabah di Surabaya Menggunakan GIS. 2. Diambil kembali dari <http://www2.eepis-its.edu/en/ta/1497/Sistem-Pengambilan-Keputusan-Penempatan-Atm--%28automated-Teller-Machine%29-Berdasarkan-Penyebaran-Nasabah-Di-Surabaya-Menggunakan-Gis>
- Bondor, C. I., Kacso, I. M., Lenghel, A., Istrate, D., & Muresan, A. (2013, February 25). VIKOR Method for Diabetic Nephropathy Risk Factors Analysis. *Applied Medical Informatics*, 32, No.1, 43-52.
- Developers, G. (2015). Google Maps Javascript API V3. Diambil kembali dari <https://developers.google.com/maps/documentation/javascript/>
- Emirson, J. (2002). *Hukum Surat Berharga dan Perkembangannya di Indonesia*. Jakarta: PT.Prenhallindo.
- Huang, J. J., Tzeng, G. H., & Liu, H. H. (2009). A revised VIKOR model for multiple criteria decision making - The Perspective of Regret Theory. *Cutting-Edge Research Topics on Multiple Criteria Decision Making*, 761-768.
- Jamila, & Hartati, S. (2011, Juni). Sistem Pendukung Keputusan Pemilihan Subkontrak Menggunakan Metode Entropy dan TOPSIS. *IJCCS*, 5, No.2.
- Mahdi. (2013). Sitem Pendukung Keputusan Kelompok Pemilihan Dayah Terbaik Menggunakan Metode Promethee dan Metode Borda. *Jurnal Politeknik Negeri Lhokseumawe*, 225-232.
- Masykur, F. (2014, Nopember). Implementasi Sistem Informasi Geografis Menggunakan Google Maps API Dalam Pemetaan Asal Mahasiswa. *Jurnal SIMETRIS*, 5, No.2.
- Mcleod, R., & Schell. (2007). *Sistem Informasi Manajemen* (9 ed.). Jakarta: PT. Index.
- Muliawan, R. (2014). Analisis dan Pemetaan Kinerja Simpang Tak Bersinyal Di Jalan Arteri Berbasis Spasial. Diambil kembali dari

<https://repository.unhas.id/.../17.%20JURNAL%20TUGAS%20AKHIR.doc>

- Niam, A., Suprayogi, A., & Awaludin, M. (2014, Oktober). Aplikasi Opens Street Map untuk Sistem Informasi Geografis Kantor Pelayanan Umum (Studi Kasus : Kota Salatiga). *Jurnal Geodesi Undip*, 3, No.4.
- NTB, B. P. (2015). Tutorial ArcGIS 10 Tingkat Dasar. *GIZ-DecGG*. Diambil kembali dari http://bappeda.ntbprov.go.id/wp-content/uploads/2013/09/Bab02_PengantarArcGIS10.pdf
- Nurdiansyah, M., Basofi, A., & Fariza, A. (2010). Sistem Informasi Geografis untuk Penentuan Lokasi SPBU Baru di Surabaya. *EEPIS Final Project*.
- Opricovic, S., & Tzeng, G. H. (2004). Compromise solution by MCDM methods: a comparative analysis of VIKOR and TOPSIS. *European Journal of Operational Research*, 2, 445-455.
- Pressman, R. (2002). *Rekayasa Perangkat Lunak*. (C. Harnaningrum, Penerj.) Yogyakarta: Andi.
- Rahardianti, E. D., & Azhari, S. (2013). Group Decision Support System Untuk Penentuan Lokasi KKN Dengan Menggunakan Metode Entropy, Promethee dan Borda (Studi Kasus pada Lembaga Pengabdian Kepada Masyarakat Universitas Ahmad Dahlan Yogyakarta). *IJCCS*.
- Ratna, L. (2000). Pencarian Suatu Alternatif yang Terbaik dengan Metode Nilai Borda. *Jurnal Manajemen Informatika, Gematika*, 3(2), 79-87.
- Reilly, B. (2002). Social Choice in the South Seas: Electoral Innovation and the Borda Count in the Pacific Island Countries. *International Political Science Review*, 23, No.4, 355-372.
- Roger S. Pressman. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku*. Yogyakarta: Andi.
- Sommerville, I. (2001). *Software Engineering*. Addison Wesley.
- Triyanti, V., & Gadis, M. T. (2008, Juni). Pemilihan Supplier Untuk Industri Makanan Menggunakan Metode Promethee. *Journal of Logistics and Supply Chain Management*, 1(2), 83-92.

- Turban, E. (2011). *Decision Support and Business Intelligence Systems* (9 ed.). New Jersey, USA, Prentice Hall: Pearson Education, Inc.
- Turban, McLean, & Wetherbe. (2005). *Information Technology For Management*. USA: John Wiley & Sons, Inc.
- Ula, M., & Azhari, S. (2013, January). Sistem Pendukung Keputusan Kelompok Penentuan Kelayakan Lokasi Pemukiman. *IJCCS*, 7, 89-100.
- Waghmode, M., & P.P.Jamsandekar. (2004, January). Decision Support Systems and its Applications in different sectors : Literature Review. *International Journal of Advance Research in Computer Science and Management Studies*, 2(1), 110-115. Diambil kembali dari <http://www.ijarcsms.com>
- Winarso, G., Trisakti, B., & K.Harsanugraha, W. (2010, Desember). Google Earth sebagai Referensi Alternatif Titik Kontrol Tanah. *INDERAJA*, 1, No.1, 27.
- Wirjayasaputra, R., & Hartati, S. (2012). Sistem Pendukung Keputusan Pengelolaan Spare Part. *IJCSS*, 6(1), 11-22.

LAMPIRAN

A. *Usecase Skenario*A.1 *Usecase Skenario Login*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *login* dapat dilihat pada Tabel 1.

Tabel 1 *Usecase Skenario Login*

Nomor <i>Usecase</i>	UC-01
Nama	<i>Login</i>
Aktor	Admin dan Pengambil Keputusan
<i>Pre Condition</i>	Admin dan Pengambil Keputusan harus mempunyai nama pengguna dan <i>password</i> dengan level masing - masing untuk masuk ke dalam sistem dan menjalankan sistem
<i>Post Condition</i>	Admin dan Pengambil Keputusan berhasil <i>login</i> atau masuk ke sistem.
SKENARIO NORMAL LOGIN	
Aktor	Sistem
1. Menjalankan sistem	2. Menampilkan <i>Form Login</i>
3. Memasukkan <i>username</i> dan <i>password</i>	5. Memeriksa data <i>login</i> ke <i>database</i>
4. Menekan tombol " <i>login</i> "	6. Menampilkan halaman <i>dashboard</i> sesuai level aktor masing – masing
SKENARIO ALTERNATIF LOGIN	
Jika aktor tidak memasukkan <i>username</i> dan atau <i>password</i>	
Aktor	Sistem
3. <i>Username</i> (kosong) atau <i>Password</i> (kosong)	5. Memeriksa kelengkapan masukan <i>username</i> dan <i>password</i> pada sistem
4. Menekan tombol " <i>login</i> "	

- 6. Menampilkan *message* “*username* atau *password* harus diisi” di bawah masukan yang kosong
- 7. Menampilkan *form login*

Jika aktor memasukkan *username* atau *password* salah

Aktor	Sistem
3. Mengisi <i>username</i> dan <i>password</i>	
4. Menekan tombol “ <i>login</i> ”	
	5. Memeriksa masukan <i>username</i> dan <i>password</i> di <i>database</i>
	6. Menampilkan <i>alert</i> “Maaf! <i>username</i> atau <i>password</i> tidak ditemukan”
	7. Menampilkan <i>form login</i>

A.2 *Usecase* Skenario *Logout*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *usecase* skenario *logout* dapat dilihat pada Tabel 2.

Tabel 2 *Usecase* Skenario *Logout*

Nomor <i>Usecase</i>	UC-02
Nama	<i>Logout</i>
Aktor	Admin dan Pengambil Keputusan
<i>Pre Condition</i>	Admin dan Pengambil Keputusan harus sudah melakukan <i>login</i> dan masuk ke dalam sistem
<i>Post Condition</i>	Admin dan Pengambil Keputusan berhasil <i>logout</i> atau keluar dari sistem.

SKENARIO NORMAL LOGOUT

Aktor	Sistem
1. Klik nama pengguna di bagian pojok kanan atas halaman	
2. Menekan tombol “ <i>logout</i> ”	
	3. Menghapus ijin akses sistem
	4. Menampilkan <i>form login</i>

A.3 *Usecase* Skenario Melihat Informasi Peta

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat informasi peta dapat dilihat pada Tabel 3.

Tabel 3 Usecase Skenario Melihat Informasi Peta

Nomor <i>Usecase</i>	UC-03
Nama	Melihat Informasi Peta
Aktor	Admin, Pengambil Keputusan dan Nasabah
<i>Pre Condition</i>	Admin, Pengambil Keputusan harus melakukan <i>login</i> ke dalam sistem, Nasabah harus sudah menjalankan sistem
<i>Post Condition</i>	Admin, Pengambil Keputusan dan Nasabah berhasil melihat informasi peta
SKENARIO NORMAL MELIHAT INFORMASI PETA	
Aktor	Sistem
1. Memilih menu <i>Dashboard</i>	2. Menampilkan halaman <i>dashboard</i> yang berisi peta kecamatan di Kabupaten Jember dan semua kategori informasi peta
3. Pilih kecamatan dan kategori yang akan ditampilkan pada peta	4. Mengambil data kelurahan dan kategori informasi dari <i>database</i> sesuai dengan kecamatan dan kategori yang dipilih
	5. Memasukkan data kelurahan dan kategori informasi ke dalam peta
	6. Menampilkan peta beserta informasinya sesuai dengan kecamatan dan kategori yang dipilih
SKENARIO NORMAL MELIHAT DETAIL INFORMASI PETA	
Aktor	Sistem
3. Klik salah satu <i>marker icon</i> , area atau garis pada peta	4. Mengambil detail kategori informasi dari <i>database</i> sesuai <i>marker icon</i> , garis atau area yang dipilih
	5. Menampilkan info <i>window</i> pada peta sesuai detail yang ingin dilihat
6. Klik <i>close</i> pada info window	

7. Menampilkan peta beserta informasinya

A.4 *Usecase* Skenario Mengelola Data *User*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data *user* dapat dilihat pada Tabel 4.

Tabel 4 *Usecase* Diagram Mengelola Data *User*

Nomor <i>Usecase</i>	UC-04
Nama	Mengelola Data <i>User</i>
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu <i>User</i>
<i>Post Condition</i>	Admin berhasil melihat, menambah, mengubah dan menghapus data <i>user</i>
SKENARIO NORMAL MELIHAT DATA USER	
Aktor	Sistem
1. Memilih menu <i>User</i>	2. Mengambil data <i>user</i> dari <i>database</i>
	3. Menampilkan data <i>user</i> pada tabel di halaman data <i>user</i>
SKENARIO NORMAL MELIHAT DETAIL USER	
Aktor	Sistem
4. Menekan tombol “Detail <i>User</i> ” pada baris data yang ingin dilihat detailnya	5. Mengambil detail <i>user</i> yang dipilih dari <i>database</i>
	6. Menampilkan detail <i>user</i> pada halaman detail <i>user</i>
7. Klik tombol “Back”	8. Mengambil data <i>user</i> dari <i>database</i>
	9. Menampilkan data <i>user</i> pada tabel di halaman data <i>user</i>
SKENARIO NORMAL MENAMBAH DATA USER	
Aktor	Sistem
4. Menekan tombol “Tambah <i>User</i> ”	5. Menampilkan <i>form</i> isian data <i>user</i>
6. Mengisi seluruh isian <i>form</i> data <i>user</i>	
7. Menekan tombol “Save”	

- 8. Menyimpan masukan data ke *database*
- 9. Menampilkan *alert* "Selamat! Data *user* sukses disimpan"
- 10. Mengambil data *user* dari *database*
- 11. Menampilkan data *user* pada tabel di halaman data *user*

SKENARIO NORMAL MENGOSONGKAN ISIAN FORM TAMBAH DATA USER

- | Aktor | Sistem |
|---------------------------|--|
| 7. Menekan tombol "Clear" | 8. Menampilkan <i>form</i> isian data <i>user</i> yang sudah dikosongkan |

SKENARIO NORMAL TAMBAH DATA KEMBALI KE HALAMAN DATA USER

- | Aktor | Sistem |
|-----------------------|--|
| 6. Klik tombol "Back" | 7. Mengambil data <i>user</i> dari <i>database</i> |
| | 8. Menampilkan data <i>user</i> pada tabel di halaman data <i>user</i> |

SKENARIO NORMAL MENGUBAH DATA USER

- | Aktor | Sistem |
|--|--|
| 4. Menekan tombol "Ubah User" pada baris data yang akan diubah | 5. Mengambil data <i>user</i> yang dipilih dari <i>database</i> |
| | 6. Menampilkan data <i>user</i> ke dalam <i>form</i> ubah data <i>user</i> |
| 7. Mengubah isian <i>form</i> ubah data <i>user</i> | |
| 8. Menekan tombol "Save Edit" | 9. Menyimpan data yang diubah ke <i>database</i> |
| | 10. Menampilkan <i>alert</i> "Selamat! Data <i>user</i> sukses diubah" |
| | 11. Mengambil data <i>user</i> dari <i>database</i> |
| | 12. Menampilkan data <i>user</i> pada tabel di halaman data <i>user</i> |

SKENARIO NORMAL MENGEMBALIKAN DATA USER SEPERTI SEMULA

- | Aktor | Sistem |
|---------------------------|---|
| 8. Menekan tombol "Clear" | 9. Menampilkan <i>form</i> isian data <i>user</i> yang belum diubah |

SKENARIO NORMAL UBAH DATA KEMBALI KE HALAMAN DATA USER

Aktor	Sistem
7. Klik tombol " <i>Back</i> "	8. Mengambil data <i>user</i> dari <i>database</i> 9. Menampilkan data <i>user</i> pada tabel di halaman data <i>user</i>

SKENARIO NORMAL HAPUS FOTO USER

Aktor	Sistem
7. Klik tombol "Hapus foto"	8. Menampilkan <i>alert</i> "Apakah Anda yakin akan menghapus foto ini?" disertai dengan tombol OK dan <i>Cancel</i>
9. Menekan tombol "OK" pada <i>alert</i>	10. Menghapus foto dari data <i>user</i> di <i>database</i> 11. Mengambil data <i>user</i> yang dipilih dari <i>database</i> 12. Menampilkan data <i>user</i> ke dalam <i>form</i> ubah data <i>user</i>

SKENARIO NORMAL BATAL MENGHAPUS FOTO USER

Aktor	Sistem
9. Menekan tombol " <i>Cancel</i> " pada <i>alert</i>	10. Mengambil data <i>user</i> yang dipilih dari <i>database</i> 11. Menampilkan data <i>user</i> ke dalam <i>form</i> ubah data <i>user</i>

SKENARIO NORMAL MENGHAPUS DATA USER

Aktor	Sistem
4. Menekan tombol "Hapus <i>User</i> " pada baris data yang akan dihapus	5. Menampilkan <i>alert</i> "Apakah Anda yakin akan menghapus data ini?" disertai dengan tombol OK dan <i>Cancel</i>
6. Menekan tombol "OK" pada <i>alert</i>	7. Menghapus data <i>user</i> yang dipilih dari <i>database</i> 8. Menampilkan <i>alert</i> "Selamat! Data <i>user</i> sukses dihapus"

9. Mengambil data *user* dari *database*
10. Menampilkan data *user* pada tabel di halaman data *user*

SKENARIO NORMAL BATAL MENGHAPUS DATA USER

- | Aktor | Sistem |
|---|---|
| 6. Menekan tombol “ <i>Cancel</i> ” pada <i>alert</i> | 7. Mengambil data <i>user</i> dari <i>database</i> |
| | 8. Menampilkan data <i>user</i> pada halaman data <i>user</i> |

SKENARIO ALTERNATIF MENAMBAH DATA USER

Jika aktor tidak mengisi data *user* secara lengkap

- | Aktor | Sistem |
|---|--|
| 6. Mengisi isian <i>form</i> data <i>user</i> tidak lengkap | 8. Memeriksa masukan data <i>user</i> |
| 7. Menekan tombol “ <i>Save</i> ” | 9. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> isian data <i>user</i> . Tulisan akan muncul dibawah masing-masing <i>field</i> masukan |

Jika aktor mengisi masukan angka dengan huruf

- | Aktor | Sistem |
|---|---|
| 6. Mengisi masukan angka pada <i>form</i> data <i>user</i> dengan huruf | 8. Memeriksa masukan data <i>user</i> |
| 7. Menekan tombol “ <i>Save</i> ” | 9. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan yang akan muncul adalah “input harus berupa angka” pada <i>form</i> isian data <i>user</i> |

Jika *personal number* sudah terdapat pada *database*

- | | |
|---|--|
| 6. Mengisi masukan <i>personal number</i> yang sama dengan data yang ada pada <i>database</i> | 8. Memeriksa masukan data <i>user</i> ke <i>database</i> |
| 7. Menekan tombol “ <i>Save</i> ” | 9. Menampilkan <i>alert</i> “ <i>Personal number duplicate entry</i> ” pada <i>form</i> isian data <i>user</i> |
-

SKENARIO ALTERNATIF MENGUBAH DATA USER	
Jika aktor tidak mengisi <i>form</i> ubah data <i>user</i> secara lengkap	
Aktor	Sistem
7. Mengubah isian <i>form</i> ubah data <i>user</i> secara tidak lengkap	
8. Menekan tombol " <i>Save Edit</i> "	9. Memeriksa ubahan data <i>user</i>
	10. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> ubah data <i>user</i> . Tulisan akan muncul dibawah masing-masing <i>field</i> masukan
Jika aktor mengisi masukan angka dengan huruf	
Aktor	Sistem
7. Mengubah isian berupa masukan angka pada <i>form</i> ubah data <i>user</i> dengan huruf	
8. Menekan tombol " <i>Save Edit</i> "	9. Memeriksa ubahan data <i>user</i>
	10. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan akan muncul "input harus berupa angka" pada <i>form</i> ubah data <i>user</i>
Jika <i>personal number</i> sudah terdapat pada <i>database</i>	
7. Mengisi masukan <i>personal number</i> yang sama dengan data yang ada pada <i>database</i>	
8. Menekan tombol " <i>Save Edit</i> "	9. Memeriksa ubahan data <i>user</i> ke <i>database</i>
	10. Menampilkan <i>alert</i> " <i>Personal number duplicate entry</i> " pada <i>form</i> ubah data <i>user</i>

A.5 *Usecase* Skenario Mengelola Data Kecamatan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data kecamatan dapat dilihat pada Tabel 5.

Tabel 5 *Usecase* Skenario Mengelola Data Kecamatan

Nomor <i>Usecase</i>	UC-05
----------------------	-------

Nama	Mengelola Data Kecamatan
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu Lokasi dan submenu Data Kecamatan
<i>Post Condition</i>	Admin berhasil melihat, menambah, mengubah dan menghapus data kecamatan

SKENARIO NORMAL MELIHAT DATA KECAMATAN

Aktor	Sistem
1. Memilih menu Lokasi	
2. Memilih submenu Data Kecamatan	
	3. Mengambil data kecamatan dari <i>database</i>
	4. Menampilkan data kecamatan pada tabel di halaman data kecamatan

SKENARIO NORMAL MELIHAT DETAIL KECAMATAN

Aktor	Sistem
5. Menekan tombol "Detail Kecamatan" pada baris data yang akan dilihat detailnya	
	6. Mengambil data kecamatan yang dipilih dari <i>database</i>
	7. Menampilkan detail dan peta data kecamatan pada halaman detail data kecamatan
8. Melihat detail data kecamatan	
9. Klik tombol <i>back</i>	
	10. Mengambil data kecamatan dari <i>database</i>
	11. Menampilkan data kecamatan pada tabel di halaman data kecamatan

SKENARIO NORMAL MENAMBAH DATA KECAMATAN

Aktor	Sistem
5. Menekan tombol "Tambah Kecamatan"	
	6. Menampilkan <i>form</i> isian tambah data kecamatan dan peta <i>digital</i>
7. Mengisi seluruh isian <i>form</i> tambah data kecamatan	
8. Klik <i>icon drawing polygon</i> pada peta	
9. Menggambar <i>polygon</i> baru	
10. Klik 2 kali pada peta jika sudah selesai menggambar	
	11. Menampilkan hasil gambar pada

- field* masukan “Hasil *Drawing*”
- 12. Menekan tombol “*Parse JSON*”
 - 13. Mengambil koordinat dan tipe *polygon*
 - 14. Menampilkan koordinat dan tipe *polygon* yang telah dibuat pada *field* masukan “*Geom*”
 - 15. Menekan tombol “*Save*”
 - 16. Menyimpan data ke *database*
 - 17. Menampilkan alert “Selamat! Data kecamatan sukses disimpan”
 - 18. Mengambil data kecamatan dari *database*
 - 19. Menampilkan data kecamatan pada tabel di halaman data kecamatan

SKENARIO NORMAL BATAL MENAMBAH POLYGON

- | Aktor | Sistem |
|---|---|
| 12. Menekan tombol “ <i>Clear Map</i> ” | 13. Menghapus hasil <i>drawing polygon</i> pada peta dan menghapus hasil gambar pada <i>field</i> masukan “Hasil <i>Drawing</i> ” |

SKENARIO NORMAL MENGOSONGKAN ISIAN FORM TAMBAH DATA KECAMATAN

- | Aktor | Sistem |
|-------------------------------------|---|
| 15. Menekan tombol “ <i>Clear</i> ” | 16. Menampilkan <i>form</i> isian data kecamatan yang sudah dikosongkan |

SKENARIO NORMAL TAMBAH DATA KEMBALI KE HALAMAN DATA KECAMATAN

- | Aktor | Sistem |
|--------------------------------|--|
| 7. Klik tombol “ <i>Back</i> ” | 8. Mengambil data kecamatan dari <i>database</i> |
| | 9. Menampilkan data kecamatan pada tabel di halaman data kecamatan |

SKENARIO NORMAL MENGUBAH DATA KECAMATAN

- | Aktor | Sistem |
|---|---|
| 5. Menekan tombol “Ubah Kecamatan” pada baris data yang akan diubah | 6. Mengambil data kecamatan yang dipilih dari <i>database</i> |

- 7. Menampilkan *form* isian ubah data kecamatan dan peta
- 8. **Mengisi isian *form* ubah data kecamatan**
- 9. Klik *icon drawing polygon* pada peta
- 10. Menggambar *polygon* sesuai dengan perubahan area kecamatan
- 11. Klik 2 kali pada peta jika sudah selesai menggambar
- 12. Menampilkan hasil gambar pada *field* masukan "*Hasil Drawing*"
- 13. **Menekan tombol "*Parse JSON*"**
- 14. Mengambil koordinat dan tipe *polygon*
- 15. Menampilkan koordinat dan tipe *polygon* yang telah dibuat pada *field* masukan "*Geom*"
- 16. **Menekan tombol "*Save Edit*"**
- 17. Mengubah data dari *database*
- 18. Menampilkan alert "Selamat! Data kecamatan sukses diubah"
- 19. Mengambil data kecamatan dari *database*
- 20. Menampilkan data kecamatan pada tabel di halaman data kecamatan

SKENARIO NORMAL BATAL MENAMBAH POLYGON

- | Aktor | Sistem |
|---|--|
| 13. Menekan tombol " <i>Clear Map</i> " | 14. Menghapus hasil <i>drawing polygon</i> pada peta dan menghapus hasil gambar pada <i>field</i> masukan " <i>Hasil Drawing</i> " |

SKENARIO NORMAL UBAH DATA KEMBALI KE HALAMAN DATA KECAMATAN

- | Aktor | Sistem |
|--------------------------------|---|
| 8. Klik tombol " <i>Back</i> " | 9. Mengambil data kecamatan dari <i>database</i> |
| | 10. Menampilkan data kecamatan pada tabel di halaman data kecamatan |

SKENARIO NORMAL MENGUBAH DATA KECAMATAN TANPA KOORDINAT

- | Aktor | Sistem |
|-------|--------|
|-------|--------|

8. Mengisi seluruh isian *form* ubah data kecamatan
9. Menekan tombol “*Save Edit*”
10. Mengubah data dari *database*
11. Menampilkan alert “Selamat! Data kecamatan sukses diubah”
12. Mengambil data kecamatan dari *database*
13. Menampilkan data kecamatan pada tabel di halaman data kecamatan

SKENARIO NORMAL MENGEMBALIKAN DATA KECAMATAN SEPERTI SEMULA

- | Aktor | Sistem |
|-------------------------------------|---|
| 16. Menekan tombol “ <i>Clear</i> ” | |
| | 17. Menampilkan <i>form</i> isian data kecamatan yang sudah dikosongkan |

SKENARIO NORMAL UBAH DATA TANPA KOORDINAT KEMBALI KE HALAMAN DATA KECAMATAN

- | Aktor | Sistem |
|--------------------------------|---|
| 8. Klik tombol “ <i>Back</i> ” | |
| | 9. Mengambil data kecamatan dari <i>database</i> |
| | 10. Menampilkan data kecamatan pada tabel di halaman data kecamatan |

SKENARIO NORMAL MENGHAPUS DATA KECAMATAN

- | Aktor | Sistem |
|---|--|
| 5. Menekan tombol “Hapus Kecamatan” pada baris data yang akan dihapus | |
| | 6. Menampilkan <i>alert</i> “Apakah Anda yakin akan menghapus data ini?” disertai dengan tombol OK dan <i>Cancel</i> |
| 7. Menekan tombol “OK” pada <i>alert</i> | |
| | 8. Menghapus data kecamatan yang dipilih dari <i>database</i> |
| | 9. Menampilkan <i>alert</i> “Selamat! Data kecamatan sukses dihapus” |
| | 10. Mengambil data kecamatan dari <i>database</i> |
| | 11. Menampilkan data kecamatan pada tabel di halaman data kecamatan |

SKENARIO NORMAL BATAL MENGHAPUS DATA KECAMATAN

Aktor	Sistem
7. Menekan tombol “ <i>Cancel</i> ” pada <i>alert</i>	
	8. Menampilkan data kecamatan pada tabel di halaman data kecamatan

SKENARIO ALTERNATIF MENAMBAH DATA KECAMATAN

Jika aktor tidak mengisi *form* tambah data kecamatan secara lengkap

Aktor	Sistem
7. Mengisi isian <i>form</i> tambah data kecamatan secara tidak lengkap	
8. Menekan tombol “ <i>Save</i> ”	
	9. Memeriksa masukan data kecamatan
	10. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> tambah data kecamatan. Tulisan akan muncul dibawah masing-masing masukan

Jika aktor mengisi masukan angka dengan huruf

Aktor	Sistem
7. Mengubah isian berupa masukan angka pada <i>form</i> tambah data kecamatan dengan huruf	
8. Menekan tombol “ <i>Save</i> ”	
	9. Memeriksa masukan data kecamatan
	10. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan akan muncul “input harus berupa angka” pada <i>form</i> tambah data kecamatan

SKENARIO ALTERNATIF MENGUBAH DATA KECAMATAN

Jika aktor tidak mengisi *form* ubah data kecamatan secara lengkap

Aktor	Sistem
8. Mengisi isian <i>form</i> ubah data kecamatan secara tidak lengkap	
9. Menekan tombol “ <i>Save Edit</i> ”	
	10. Memeriksa ubahan data kecamatan
	11. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> ubah data kecamatan. Tulisan akan muncul dibawah masing-masing masukan

Jika aktor mengisi masukan angka dengan huruf

Aktor	Sistem
8. Mengubah isian berupa masukan angka pada <i>form</i> ubah data kecamatan dengan huruf	
9. Menekan tombol “ <i>Save Edit</i> ”	
	10. Memeriksa ubahan data kecamatan
	11. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan akan muncul “input harus berupa angka” pada <i>form</i> ubah data kecamatan

A.6 *Usecase* Skenario Mengelola Data Kelurahan

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatuf *usecase* skenario mengelola data kelurahan dapat dilihat pada Tabel 6.

Tabel 6 *Usecase* Skenario Mengelola Data Kelurahan

Nomor <i>Usecase</i>	UC-06
Nama	Mengelola Data Kelurahan
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu Lokasi dan submenu Data Kelurahan
<i>Post Condition</i>	Admin berhasil melihat, menambah, mengubah dan menghapus data kelurahan

SKENARIO NORMAL MELIHAT DATA KELURAHAN

Aktor	Sistem
1. Memilih menu Lokasi	
2. Memilih submenu Data Kelurahan	
	3. Mengambil data kelurahan dari <i>database</i>
	4. Menampilkan data kelurahan pada tabel di halaman data kelurahan

SKENARIO NORMAL MELIHAT DETAIL KELURAHAN

Aktor	Sistem
5. Menekan tombol “Detail Kelurahan” pada baris data yang akan dilihat detailnya	
	6. Mengambil data kelurahan yang dipilih dari <i>database</i>
	7. Menampilkan detail dan peta data kelurahan pada halaman detail data

- | | |
|--|---|
| <ul style="list-style-type: none"> 8. Melihat detail data kelurahan 9. Klik tombol <i>back</i> | <p>kelurahan</p> <ul style="list-style-type: none"> 10. Mengambil data kelurahan dari <i>database</i> 11. Menampilkan data kelurahan pada tabel di halaman data kelurahan |
|--|---|

SKENARIO NORMAL MENAMBAH DATA KELURAHAN

- | Aktor | Sistem |
|--|--|
| <ul style="list-style-type: none"> 5. Menekan tombol “Tambah Kelurahan” | <ul style="list-style-type: none"> 6. Menampilkan <i>form</i> isian tambah data kelurahan dan peta |
| <ul style="list-style-type: none"> 7. Mengisi seluruh isian <i>form</i> tambah data kelurahan 8. Klik <i>icon drawing polygon</i> pada peta 9. Menggambar <i>polygon</i> baru 10. Klik 2 kali pada peta jika sudah selesai menggambar | <ul style="list-style-type: none"> 11. Menampilkan hasil gambar pada <i>field</i> masukan “<i>Hasil Drawing</i>” |
| <ul style="list-style-type: none"> 12. Menekan tombol “<i>Parse JSON</i>” | <ul style="list-style-type: none"> 13. Mengambil koordinat dan tipe <i>polygon</i> 14. Menampilkan koordinat dan tipe <i>polygon</i> yang telah dibuat pada <i>field</i> masukan “<i>Geom</i>” |
| <ul style="list-style-type: none"> 15. Menekan tombol “<i>Save</i>” | <ul style="list-style-type: none"> 16. Menyimpan data ke <i>database</i> 17. Menampilkan alert “Selamat! Data kelurahan sukses disimpan” 18. Mengambil data kelurahan dari <i>database</i> 19. Menampilkan data kelurahan pada tabel di halaman data kelurahan |

SKENARIO NORMAL BATAL MENAMBAH POLYGON

- | Aktor | Sistem |
|---|--|
| <ul style="list-style-type: none"> 12. Menekan tombol “<i>Clear Map</i>” | <ul style="list-style-type: none"> 13. Menghapus hasil <i>drawing polygon</i> pada peta |

SKENARIO NORMAL MENGOSONGKAN ISIAN *FORM* TAMBAH DATA KELURAHAN

Aktor	Sistem
-------	--------

15. Menekan tombol “*Clear*”

16. Menampilkan *form* isian data kelurahan yang sudah dikosongkan

SKENARIO NORMAL TAMBAH DATA KEMBALI KE HALAMAN DATA KELURAHAN

Aktor	Sistem
7. Klik tombol “ <i>Back</i> ”	8. Mengambil data kelurahan dari <i>database</i>
	9. Menampilkan data kelurahan pada tabel di halaman data kelurahan

SKENARIO NORMAL MENGUBAH DATA KELURAHAN

Aktor	Sistem
5. Menekan tombol “Ubah Kelurahan” pada baris data yang akan diubah	6. Mengambil data kelurahan yang dipilih dari <i>database</i>
	7. Menampilkan <i>form</i> isian ubah data kelurahan dan peta
8. Mengisi seluruh isian <i>form</i> ubah data kelurahan	
9. Klik <i>icon drawing polygon</i> pada peta	
10. Menggambar <i>polygon</i> sesuai dengan perubahan area kelurahan	
11. Klik 2 kali pada peta jika sudah selesai menggambar	
12. Menampilkan hasil gambar pada <i>field</i> masukan “ <i>Hasil Drawing</i> ”	
13. Menekan tombol “<i>Parse JSON</i>”	
	14. Mengambil koordinat dan tipe <i>polygon</i>
	15. Menampilkan koordinat dan tipe <i>polygon</i> yang telah dibuat pada <i>field</i> masukan “ <i>Geom</i> ”
16. Menekan tombol “<i>Save Edit</i>”	
	17. Mengubah data dari <i>database</i>
	18. Menampilkan alert “Selamat! Data kelurahan sukses diubah”
	19. Mengambil data kelurahan dari <i>database</i>
	20. Menampilkan data kelurahan pada

tabel di halaman data kelurahan

SKENARIO NORMAL BATAL MENAMBAH POLYGON

- | Aktor | Sistem |
|---|--|
| 13. Menekan tombol " <i>Clear Map</i> " | 14. Menghapus hasil <i>drawing polygon</i> pada peta |

SKENARIO NORMAL MENGUBAH DATA KELURAHAN TANPA KOORDINAT

- | Aktor | Sistem |
|--|---|
| 8. Mengisi seluruh isian <i>form</i> ubah data kelurahan | |
| 9. Menekan tombol " <i>Save Edit</i> " | 10. Mengubah data dari <i>database</i> |
| | 11. Menampilkan alert "Selamat! Data kelurahan sukses diubah" |
| | 12. Mengambil data kelurahan dari <i>database</i> |
| | 13. Menampilkan data kelurahan pada tabel di halaman data kelurahan |

SKENARIO NORMAL MENGEMBALIKAN DATA KELURAHAN SEPERTI SEMULA

- | Aktor | Sistem |
|-------------------------------------|---|
| 16. Menekan tombol " <i>Clear</i> " | 17. Menampilkan <i>form</i> isian data kelurahan yang sudah dikosongkan |

SKENARIO NORMAL UBAH DATA KEMBALI KE HALAMAN DATA KELURAHAN

- | Aktor | Sistem |
|--------------------------------|---|
| 8. Klik tombol " <i>Back</i> " | 9. Mengambil data kelurahan dari <i>database</i> |
| | 10. Menampilkan data kelurahan pada tabel di halaman data kelurahan |

SKENARIO NORMAL MENGHAPUS DATA KELURAHAN

- | Aktor | Sistem |
|---|--|
| 5. Menekan tombol "Hapus Kelurahan" pada baris data yang akan dihapus | 6. Menampilkan <i>alert</i> "Apakah Anda yakin akan menghapus data ini?" disertai dengan tombol OK dan <i>Cancel</i> |
| 7. Menekan tombol "OK" pada <i>alert</i> | |

8. Menghapus data kelurahan yang dipilih dari *database*
9. Menampilkan *alert* "Selamat! Data kelurahan sukses dihapus"
10. Mengambil data kelurahan dari *database*
11. Menampilkan data kelurahan pada tabel di halaman data kelurahan

SKENARIO NORMAL BATAL MENGHAPUS DATA KELURAHAN

- | Aktor | Sistem |
|---|--|
| 7. Menekan tombol " <i>Cancel</i> " pada <i>alert</i> | 8. Menampilkan data kelurahan pada tabel di halaman data kelurahan |

SKENARIO ALTERNATIF MENAMBAH DATA KELURAHAN

Jika aktor tidak mengisi *form* tambah data kelurahan secara lengkap

- | Aktor | Sistem |
|---|---|
| 7. Mengisi isian <i>form</i> tambah data kelurahan secara tidak lengkap | 9. Memeriksa masukan data kelurahan |
| 8. Menekan tombol " <i>Save</i> " | 10. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> tambah data kelurahan. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan |

SKENARIO ALTERNATIF MENGUBAH DATA KELURAHAN

Jika aktor tidak mengisi *form* ubah data kelurahan secara lengkap

- | Aktor | Sistem |
|---|---|
| 8. Mengisi isian <i>form</i> ubah data kelurahan secara tidak lengkap | 10. Memeriksa ubahan data kelurahan |
| 9. Menekan tombol " <i>Save Edit</i> " | 11. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> ubah data kelurahan. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan |

A.7 *Usecase* Skenario Mengelola Data Kriteria

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data kriteria dapat dilihat pada Tabel 7.

Tabel 7 *Usecase* Skenario Mengelola Data Kriteria

Nomor <i>Usecase</i>	UC-07
Nama	Mengelola Data Kriteria
Aktor	Admin
<i>Pre Condition</i>	Admin memilih menu Data Kriteria
<i>Post Condition</i>	Admin berhasil melihat, menambah, mengubah dan menghapus data kriteria
SKENARIO NORMAL MELIHAT DATA KRITERIA	
Aktor	Sistem
1. Memilih menu Data Kriteria	2. Mengambil data kriteria dari <i>database</i>
	3. Menampilkan data kriteria pada tabel di halaman data kriteria
SKENARIO NORMAL MENAMBAH DATA KRITERIA	
Aktor	Sistem
4. Menekan tombol "Tambah Kriteria"	5. Menampilkan <i>form</i> isian data kriteria
6. Mengisi seluruh isian <i>form</i> data kriteria	8. Menyimpan masukan ke <i>database</i>
7. Menekan tombol "Save"	9. Menampilkan <i>alert</i> "Selamat! Data kriteria sukses disimpan"
	10. Mengambil data kriteria dari <i>database</i>
	11. Menampilkan data kriteria pada tabel di halaman data kriteria
SKENARIO NORMAL MENGOSONGKAN ISIAN FORM TAMBAH DATA KRITERIA	
Aktor	Sistem
7. Menekan tombol "Clear"	8. Menampilkan <i>form</i> isian data kriteria yang sudah dikosongkan
SKENARIO NORMAL TAMBAH DATA KEMBALI KE HALAMAN DATA KRITERIA	
Aktor	Sistem

- 6. Klik tombol “*Back*”
- 7. Mengambil data kriteria dari *database*
- 8. Menampilkan data kriteria pada tabel di halaman data kriteria

SKENARIO NORMAL MENGUBAH DATA KRITERIA

- | Aktor | Sistem |
|--|--|
| 4. Menekan tombol “Ubah Kriteria” pada baris data yang akan diubah | |
| | 5. Mengambil data kriteria yang dipilih dari <i>database</i> |
| | 6. Menampilkan data kriteria ke dalam <i>form</i> ubah data kriteria |
| 7. Mengubah isian <i>form</i> ubah data kriteria | |
| 8. Menekan tombol “<i>Save Edit</i>” | |
| | 9. Menyimpan data yang diubah ke <i>database</i> |
| | 10. Menampilkan <i>alert</i> “Selamat! Data kriteria sukses diubah” |
| | 11. Mengambil data kriteria dari <i>database</i> |
| | 12. Menampilkan data kriteria pada tabel di halaman data kriteria |

SKENARIO NORMAL MENGEMBALIKAN DATA KRITERIA SEPERTI SEMULA

- | Aktor | Sistem |
|------------------------------------|--|
| 8. Menekan tombol “ <i>Clear</i> ” | |
| | 9. Menampilkan <i>form</i> isian data kriteria yang belum diubah |

SKENARIO NORMAL UBAH DATA KEMBALI KE HALAMAN DATA KRITERIA

- | Aktor | Sistem |
|--------------------------------|--|
| 7. Klik tombol “ <i>Back</i> ” | |
| | 8. Mengambil data kriteria dari <i>database</i> |
| | 9. Menampilkan data kriteria pada tabel di halaman data kriteria |

SKENARIO NORMAL MENGHAPUS DATA KRITERIA

- | Aktor | Sistem |
|--|--|
| 4. Menekan tombol “Hapus Kriteria” pada baris data yang akan dihapus | |
| | 5. Menampilkan <i>alert</i> “Apakah Anda |

- yakin akan menghapus data ini?" disertai dengan tombol OK dan *Cancel*
6. Menekan tombol "OK" pada *alert*
 7. Menghapus data kriteria yang dipilih dari *database*
 8. Menampilkan *alert* "Selamat! Data kriteria sukses dihapus"
 9. Mengambil data kriteria dari *database*
 10. Menampilkan data kriteria pada tabel di halaman data kriteria

SKENARIO NORMAL BATAL MENGHAPUS DATA KRITERIA

- | Aktor | Sistem |
|---|---|
| 6. Menekan tombol " <i>Cancel</i> " pada <i>alert</i> | |
| | 7. Menampilkan data kriteria pada halaman data kriteria |

SKENARIO ALTERNATIF MENAMBAH DATA KRITERIA

Jika aktor tidak mengisi *form* data kriteria secara lengkap

- | Aktor | Sistem |
|--|--|
| 6. Mengisi isian <i>form</i> data kriteria tidak lengkap | |
| 7. Menekan tombol " <i>Save</i> " | |
| | 8. Memeriksa masukan data kriteria |
| | 9. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> isian data kriteria. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan |

SKENARIO ALTERNATIF MENGUBAH DATA KRITERIA

Jika aktor tidak mengisi *form* ubah data kriteria secara lengkap

- | Aktor | Sistem |
|---|--|
| 7. Mengubah isian <i>form</i> ubah data kriteria secara tidak lengkap | |
| 8. Menekan tombol " <i>Save Edit</i> " | |
| | 9. Memeriksa ubahan data kriteria |
| | 10. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> ubah data kriteria. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan |

A.8 *Usecase* Skenario Mengelola Batasan Kriteria

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola batasan kriteria dapat dilihat pada Tabel 8.

Tabel 8 *Usecase* Skenario Mengelola Batasan Kriteria

Nomor <i>Usecase</i>	UC-08
Nama	Mengelola Batasan Kriteria
Aktor	Admin
<i>Pre Condition</i>	Admin memilih tombol Batasan Kriteria pada baris data kriteria yang dipilih
<i>Post Condition</i>	Admin berhasil melihat, menambah, dan mengubah batasan kriteria

SKENARIO NORMAL MELIHAT BATASAN KRITERIA

Aktor	Sistem
1. Memilih menu Data Kriteria	2. Mengambil data kriteria dari <i>database</i>
	3. Menampilkan data kriteria pada tabel di halaman data kriteria
4. Menekan tombol "Batasan Kriteria" pada salah satu baris data kriteria yang dipilih	5. Memeriksa ada atau tidaknya batasan kriteria dari kriteria yang dipilih pada <i>database</i>
	6. Terdapat batasan kriteria dari kriteria yang dipilih
	7. Mengambil batasan kriteria yang sudah ada di <i>database</i> berdasarkan kriteria yang dipilih
	8. Menampilkan batasan kriteria pada tabel di halaman batasan kriteria
8. Melihat batasan kriteria	

SKENARIO NORMAL KEMBALI KE HALAMAN DATA KRITERIA

Aktor	Sistem
9. Klik tombol "Back"	10. Mengambil data kriteria dari <i>database</i>
	11. Menampilkan data kriteria pada tabel di halaman data kriteria

SKENARIO NORMAL MENGUBAH BATASAN KRITERIA

Aktor	Sistem
-------	--------

- 9. Menekan tombol “Ubah Batasan Kriteria” pada baris data yang akan diubah
- 10. Mengambil batasan kriteria yang dipilih dari *database*
- 11. Menampilkan batasan kriteria ke dalam *form* ubah batasan kriteria
- 12. **Mengubah isian *form* ubah batasan kriteria**
- 13. **Menekan tombol “*Save Edit*”**
- 14. Menyimpan data yang diubah ke *database*
- 15. Menampilkan *alert* “Selamat! Batasan kriteria sukses diubah”
- 16. Mengambil batasan kriteria dari *database*
- 17. Menampilkan batasan kriteria pada tabel di halaman batasan kriteria

**SKENARIO NORMAL MENGEMBALIKAN BATASAN KRITERIA
SEPERTI SEMULA**

- | Aktor | Sistem |
|-------------------------------------|--|
| 13. Menekan tombol “ <i>Clear</i> ” | 14. Menampilkan <i>form</i> isian batasan kriteria yang belum diubah |

**SKENARIO NORMAL UBAH DATA KEMBALI KE HALAMAN
BATASAN KRITERIA**

- | Aktor | Sistem |
|---------------------------------|---|
| 12. Klik tombol “ <i>Back</i> ” | 13. Mengambil batasan kriteria dari <i>database</i> |
| | 14. Menampilkan batasan kriteria pada tabel di halaman batasan kriteria |

SKENARIO NORMAL MENAMBAH BATASAN KRITERIA

- | Aktor | Sistem |
|---|---|
| 4. Menekan tombol “Batasan Kriteria” pada salah satu baris data kriteria yang dipilih | 5. Memeriksa apakah ada data batasan kriteria pada kriteria yang dipilih dari <i>database</i> |
| | 6. Tidak terdapat batasan kriteria pada kriteria yang dipilih |
| | 7. Menampilkan <i>form</i> isian batasan kriteria |
| 8. Mengisi seluruh isian <i>form</i> batasan kriteria | |

- | | |
|--|---|
| <p>9. Menekan tombol “<i>Save All</i>”</p> | <p>10. Menyimpan masukan ke <i>database</i></p> <p>11. Menampilkan <i>alert</i> “Selamat! Batasan kriteria sukses disimpan”</p> <p>12. Mengambil batasan kriteria dari <i>database</i></p> <p>13. Menampilkan batasan kriteria pada tabel di halaman batasan kriteria</p> |
|--|---|

SKENARIO NORMAL MENGOSONGKAN ISIAN *FORM* TAMBAH BATASAN KRITERIA

- | Aktor | Sistem |
|---|--|
| <p>9. Menekan tombol “<i>Clear</i>”</p> | <p>10. Menampilkan <i>form</i> isian batasan kriteria yang sudah dikosongkan</p> |

SKENARIO NORMAL TAMBAH DATA KEMBALI KE HALAMAN DATA KRITERIA

- | Aktor | Sistem |
|-------------------------------------|---|
| <p>8. Klik tombol “<i>Back</i>”</p> | <p>9. Mengambil data kriteria dari <i>database</i></p> <p>10. Menampilkan data kriteria pada tabel di halaman data kriteria</p> |

SKENARIO ALTERNATIF MENAMBAH BATASAN KRITERIA

Jika aktor tidak mengisi *form* batasan kriteria secara lengkap

- | Aktor | Sistem |
|---|---|
| <p>8. Mengisi isian <i>form</i> batasan kriteria tidak lengkap</p> <p>9. Menekan tombol “<i>Save</i>”</p> | <p>10. Memeriksa masukan batasan kriteria</p> <p>11. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> isian batasan kriteria. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan</p> |

Jika aktor mengisi masukan angka dengan huruf

- | Aktor | Sistem |
|---|--|
| <p>8. Mengisi masukan angka pada <i>form</i> batasan kriteria dengan huruf</p> <p>9. Menekan tombol “<i>Save</i>”</p> | <p>10. Memeriksa masukan batasan kriteria</p> <p>11. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan yang akan muncul adalah</p> |
-

“input harus berupa angka” pada *form* isian batasan kriteria

SKENARIO ALTERNATIF MENGUBAH BATASAN KRITERIA

Jika aktor tidak mengisi *form* ubah batasan kriteria secara lengkap

Aktor	Sistem
12. Mengubah isian <i>form</i> ubah batasan kriteria secara tidak lengkap	
13. Menekan tombol “ <i>Save Edit</i> ”	14. Memeriksa ubahan batasan kriteria 15. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> ubah batasan kriteria. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan

Jika aktor mengisi masukan angka dengan huruf

Aktor	Sistem
12. Mengubah isian berupa masukan angka pada <i>form</i> ubah batasan kriteria dengan huruf	
13. Menekan tombol “ <i>Save Edit</i> ”	14. Memeriksa ubahan batasan kriteria 15. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan akan muncul “input harus berupa angka” pada <i>form</i> ubah batasan kriteria

A.9 *Usecase* Skenario Mengelola Nilai Kriteria

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat nilai kriteria dapat dilihat pada Tabel 9.

Tabel 9 *Usecase* Skenario Mengelola Nilai Kriteria

Nomor <i>Usecase</i>	UC-09
Nama	Mengelola Nilai Kriteria
Aktor	Admin
<i>Pre Condition</i>	Admin memilih tombol Nilai Kriteria pada halaman batasan kriteria
<i>Post Condition</i>	Admin berhasil melihat, menambah, mengubah dan menghapus nilai kriteria

SKENARIO NORMAL MELIHAT NILAI KRITERIA

Aktor	Sistem
1. Memilih menu Data Kriteria	2. Mengambil data kriteria dari <i>database</i>
	3. Menampilkan data kriteria pada tabel di halaman data kriteria
4. Menekan tombol “Batasan Kriteria” pada salah satu baris data kriteria yang dipilih	5. Mengambil batasan kriteria dari <i>database</i>
	6. Menampilkan batasan kriteria pada tabel di halaman data batasan kriteria
7. Menekan tombol “Nilai Kriteria”	8. Memeriksa ada atau tidaknya nilai kriteria dari kriteria yang dipilih pada <i>database</i>
	9. Terdapat nilai kriteria dari kriteria yang dipilih
	10. Mengambil nilai kriteria dari <i>database</i> berdasarkan kriteria yang dipilih
	11. Menampilkan nilai kriteria pada tabel di halaman nilai kriteria

SKENARIO NORMAL MENAMBAH NILAI KRITERIA BARU

Aktor	Sistem
12. Menekan tombol “Tambah Nilai Kriteria”	13. Menampilkan <i>form</i> isian nilai kriteria
14. Mengisi seluruh isian <i>form</i> nilai kriteria	16. Menyimpan masukan ke <i>database</i>
15. Menekan tombol “Save”	17. Menampilkan <i>alert</i> “Selamat! Nilai kriteria sukses disimpan”
	18. Mengambil nilai kriteria dari <i>database</i>
	19. Menampilkan nilai kriteria pada tabel di halaman nilai kriteria

SKENARIO NORMAL TAMBAH NILAI KRITERIA BARU KEMBALI KE HALAMAN NILAI KRITERIA

Aktor	Sistem
14. Klik tombol “Back”	15. Mengambil nilai kriteria dari <i>database</i>

16. Menampilkan nilai kriteria pada tabel di halaman nilai kriteria

SKENARIO NORMAL MENGOSONGKAN ISIAN *FORM* TAMBAH NILAI KRITERIA

- | Aktor | Sistem |
|-------------------------------------|---|
| 15. Menekan tombol " <i>Clear</i> " | 16. Menampilkan <i>form</i> isian nilai kriteria yang sudah dikosongkan |

SKENARIO NORMAL MENGUBAH NILAI KRITERIA

- | Aktor | Sistem |
|---|---|
| 12. Menekan tombol "Ubah Nilai Kriteria" pada baris data yang akan diubah | 13. Mengambil nilai kriteria yang dipilih dari <i>database</i> |
| 15. Mengubah isian <i>form</i> ubah nilai kriteria | 14. Menampilkan nilai kriteria ke dalam <i>form</i> ubah nilai kriteria |
| 16. Menekan tombol " <i>Save Edit</i> " | 17. Menyimpan data yang diubah ke <i>database</i> |
| | 18. Menampilkan <i>alert</i> "Selamat! Nilai kriteria sukses diubah" |
| | 19. Mengambil nilai kriteria dari <i>database</i> |
| | 20. Menampilkan nilai kriteria pada tabel di halaman nilai kriteria |

SKENARIO NORMAL MENGEMBALIKAN NILAI KRITERIA SEPERTI SEMULA

- | Aktor | Sistem |
|-------------------------------------|--|
| 16. Menekan tombol " <i>Clear</i> " | 17. Menampilkan <i>form</i> isian nilai kriteria yang belum diubah |

SKENARIO NORMAL UBAH NILAI KRITERIA KEMBALI KE HALAMAN NILAI KRITERIA

- | Aktor | Sistem |
|---------------------------------|---|
| 15. Klik tombol " <i>Back</i> " | 16. Mengambil nilai kriteria dari <i>database</i> |
| | 17. Menampilkan nilai kriteria pada tabel di halaman nilai kriteria |

SKENARIO NORMAL MENGHAPUS NILAI KRITERIA

- | Aktor | Sistem |
|---------------------------------|--------|
| 12. Menekan tombol "Hapus Nilai | |

Kriteria” pada baris data yang akan dihapus

- 13. Menampilkan *alert* “Apakah Anda yakin akan menghapus data ini?” disertai dengan tombol OK dan *Cancel*
- 14. **Menekan tombol “OK” pada *alert***
- 15. Menghapus nilai kriteria yang dipilih dari *database*
- 16. Menampilkan *alert* “Selamat! Nilai kriteria sukses dihapus”
- 17. Mengambil nilai kriteria dari *database*
- 18. Menampilkan nilai kriteria pada tabel di halaman nilai kriteria

SKENARIO NORMAL BATAL MENGHAPUS NILAI KRITERIA

Aktor	Sistem
14. Menekan tombol “ <i>Cancel</i> ” pada <i>alert</i>	15. Menampilkan nilai kriteria pada halaman nilai kriteria

SKENARIO NORMAL MENAMBAH NILAI KRITERIA UNTUK SETIAP KELURAHAN

Aktor	Sistem
7. Menekan tombol “Nilai Kriteria”	8. Memeriksa ada atau tidaknya nilai kriteria dari <i>database</i> berdasarkan kriteria yang dipilih
	9. Tidak terdapat nilai kriteria berdasarkan kriteria yang dipilih
	10. Menampilkan <i>form</i> isian nilai kriteria
11. Mengisi seluruh isian <i>form</i> nilai kriteria	
12. Menekan tombol “ <i>Save All</i> ”	13. Menyimpan masukan ke <i>database</i>
	14. Menampilkan <i>alert</i> “Selamat! Nilai kriteria sukses disimpan”
	15. Mengambil nilai kriteria dari <i>database</i>
	16. Menampilkan nilai kriteria pada tabel di halaman nilai kriteria

SKENARIO NORMAL MENGOSONGKAN ISIAN FORM TAMBAH NILAI KRITERIA UNTUK SETIAP KELURAHAN

Aktor	Sistem
-------	--------

12. Menekan tombol “*Clear*”

13. Menampilkan *form* isian nilai kriteria yang sudah dikosongkan

SKENARIO NORMAL TAMBAH NILAI KRITERIA UNTUK SETIAP KELURAHAN KEMBALI KE HALAMAN BATASAN KRITERIA

Aktor

Sistem

11. Klik tombol “*Back*”

12. Mengambil batasan kriteria dari *database*

13. Menampilkan batasan kriteria pada tabel di halaman batasan kriteria

SKENARIO ALTERNATIF MENAMBAH NILAI KRITERIA

Jika aktor tidak mengisi *form* nilai kriteria secara lengkap

Aktor

Sistem

14. Mengisi isian *form* nilai kriteria tidak lengkap

15. Menekan tombol “*Save*”

16. Memeriksa masukan nilai kriteria

17. Menampilkan *message* dengan menunjuk isian mana yang tidak diisi pada *form* isian nilai kriteria. Tulisan akan muncul dibawah masing-masing *field* masukan

Jika aktor mengisi masukan angka dengan huruf

Aktor

Sistem

14. Mengisi masukan angka pada *form* nilai kriteria dengan huruf

15. Menekan tombol “*Save*”

16. Memeriksa masukan nilai kriteria

17. Menampilkan *message* dengan menunjuk isian mana yang salah. Tulisan yang akan muncul adalah “input harus berupa angka” pada *form* isian nilai kriteria

SKENARIO ALTERNATIF MENGUBAH NILAI KRITERIA

Jika aktor tidak mengisi *form* ubah nilai kriteria secara lengkap

Aktor

Sistem

15. Mengubah isian *form* ubah nilai kriteria secara tidak lengkap

16. Menekan tombol “*Save Edit*”

17. Memeriksa ubahan nilai kriteria

18. Menampilkan *message* dengan menunjuk isian mana yang tidak diisi

pada *form* ubah nilai kriteria. Tulisan akan muncul dibawah masing-masing *field* masukan

Jika aktor mengisi masukan angka dengan huruf	
Aktor	Sistem
15. Mengubah isian berupa masukan angka pada <i>form</i> nilai batasan kriteria dengan huruf 16. Menekan tombol " <i>Save Edit</i> "	17. Memeriksa ubahan nilai kriteria 18. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan akan muncul "input harus berupa angka" pada <i>form</i> ubah nilai kriteria

A.10 *Usecase* Skenario Mengelola Data Kategori Informasi Peta

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data kategori informasi peta dapat dilihat pada Tabel 10.

Tabel 10 *Usecase* Skenario Mengelola Data Kategori Informasi Peta

Nomor <i>Usecase</i>	UC-10
Nama	Mengelola Data Kategori Informasi Peta
Aktor	Admin
<i>Pre Condition</i>	Admin memilih tombol Kategori Informasi Peta pada baris data kriteria yang dipilih
<i>Post Condition</i>	Admin berhasil melihat, menambah, mengubah dan menghapus data kategori informasi peta

SKENARIO NORMAL MELIHAT DATA KATEGORI INFORMASI PETA

Aktor	Sistem
1. Memilih menu Data Kriteria 4. Menekan tombol "Batasan Kriteria" pada salah satu baris data kriteria yang dipilih	2. Mengambil data kriteria dari <i>database</i> 3. Menampilkan data kriteria pada tabel di halaman data kriteria 5. Mengambil batasan kriteria dari <i>database</i> 6. Menampilkan batasan kriteria pada

- | | |
|-----------------------------------|--|
| 7. Menekan tombol “Kategori Peta” | tabel di halaman batasan kriteria |
| | 8. Mengambil data kategori informasi peta dari <i>database</i> |
| | 9. Menampilkan data kategori informasi peta pada tabel di halaman kategori informasi |

SKENARIO NORMAL MENAMBAH DATA KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|--|---|
| 10. Menekan tombol “Tambah Kategori” | 11. Menampilkan <i>form</i> isian data kategori informasi peta |
| 12. Mengisi seluruh isian <i>form</i> data kategori informasi peta | |
| 13. Menekan tombol “Save” | 14. Menyimpan masukan ke <i>database</i> |
| | 15. Menampilkan <i>alert</i> “Selamat! Kategori informasi peta sukses disimpan” |
| | 16. Mengambil data kategori informasi peta dari <i>database</i> |
| | 17. Menampilkan data kategori informasi pada tabel di halaman kategori informasi peta |

SKENARIO NORMAL MENGOSONGKAN ISIAN FORM TAMBAH DATA KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|----------------------------|--|
| 13. Menekan tombol “Clear” | 14. Menampilkan <i>form</i> isian kategori informasi peta yang sudah dikosongkan |

SKENARIO NORMAL TAMBAH DATA KEMBALI KE HALAMAN KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|------------------------|---|
| 12. Klik tombol “Back” | 13. Mengambil data kategori informasi peta dari <i>database</i> |
| | 14. Menampilkan data kategori informasi pada tabel di halaman kategori informasi peta |

SKENARIO NORMAL MENGUBAH KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|--------------------------|--------|
| 10. Menekan tombol “Ubah | |

Kategori” pada baris data yang akan diubah

- 11. Mengambil data kategori informasi peta yang dipilih dari *database*
- 12. Menampilkan data kategori informasi peta ke dalam *form* ubah data kategori informasi peta
- 13. **Mengubah isian *form* ubah data kategori informasi peta**
- 14. **Menekan tombol “*Save Edit*”**
- 15. Menyimpan data yang diubah ke *database*
- 16. Menampilkan *alert* “Selamat! Kategori informasi peta sukses diubah”
- 17. Mengambil data kategori informasi peta dari *database*
- 18. Menampilkan data kategori informasi peta pada tabel di halaman data kategori informasi peta

SKENARIO NORMAL MENGEMBALIKAN DATA KATEGORI INFORMASI PETA SEPERTI SEMULA

- | Aktor | Sistem |
|-------------------------------------|--|
| 14. Menekan tombol “ <i>Clear</i> ” | 15. Menampilkan <i>form</i> isian data kategori informasi peta yang belum diubah |

SKENARIO NORMAL HAPUS ICON

- | Aktor | Sistem |
|------------------------------|--|
| 13. Klik tombol “Hapus Icon” | 14. Menghapus icon dari data kategori di <i>database</i> |
| | 15. Menampilkan <i>form</i> isian data kategori informasi peta |

SKENARIO NORMAL UBAH DATA KEMBALI KE HALAMAN KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|---------------------------------|---|
| 13. Klik tombol “ <i>Back</i> ” | 14. Mengambil data kategori informasi peta dari <i>database</i> |
| | 15. Menampilkan data kategori informasi pada tabel di halaman kategori informasi peta |

SKENARIO NORMAL MENGHAPUS DATA KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|-------|--------|
|-------|--------|

- 10. Menekan tombol “Hapus Kategori” pada baris data yang akan dihapus
- 11. Menampilkan *alert* “Apakah Anda yakin akan menghapus data ini?” disertai dengan tombol OK dan *Cancel*
- 12. Menekan tombol “OK” pada *alert*
- 13. Menghapus data kategori informasi peta yang dipilih dari *database*
- 14. Menampilkan *alert* “Selamat! Kategori Informasi Peta sukses dihapus”
- 15. Mengambil data kategori informasi peta dari *database*
- 16. Menampilkan data kategori informasi peta pada tabel di halaman data kategori informasi peta

SKENARIO NORMAL BATAL MENGHAPUS DATA KATEGORI INFORMASI PETA

Aktor	Sistem
12. Menekan tombol “ <i>Cancel</i> ” pada <i>alert</i>	13. Menampilkan data kategori informasi peta pada tabel di halaman data kategori informasi peta

SKENARIO ALTERNATIF MENAMBAH DATA KATEGORI INFORMASI PETA

Jika aktor tidak mengisi *form* tambah data kategori informasi peta secara lengkap

Aktor	Sistem
12. Mengisi isian <i>form</i> data kategori informasi peta tidak lengkap	
13. Menekan tombol “ <i>Save</i> ”	14. Memeriksa masukan kategori informasi peta
	15. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> isian data kategori informasi peta. Tulisan akan muncul dibawah masing-masing <i>field</i> inputan

SKENARIO ALTERNATIF MENGUBAH DATA KATEGORI INFORMASI PETA

Jika aktor tidak mengisi form ubah data kategori informasi peta secara lengkap

Aktor	Sistem
13. Mengubah isian form ubah data kategori informasi peta secara tidak lengkap	
14. Menekan tombol "Save Edit"	
	15. Memeriksa ubahan kategori informasi peta
	16. Menampilkan message dengan menunjuk isian mana yang tidak diisi pada form ubah data kategori informasi peta. Tulisan akan muncul dibawah masing-masing field masukan

A.11 Usecase Skenario Mengelola Nilai Kategori Informasi Peta

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario mengelola nilai kategori informasi peta dapat dilihat pada Tabel 11.

Tabel 11 Usecase Skenario Mengelola Nilai Kategori Informasi Peta

Nomor Usecase	UC-11
Nama	Mengelola Nilai Kategori Informasi Peta
Aktor	Admin
Pre Condition	Admin memilih tombol Nilai Kategori pada baris data kategori informasi peta yang dipilih
Post Condition	Admin berhasil melihat, menambah, mengubah dan menghapus nilai kategori informasi peta

SKENARIO NORMAL MELIHAT NILAI KATEGORI INFORMASI PETA

Aktor	Sistem
1. Memilih menu Data Kriteria	
	2. Mengambil data kriteria dari database
	3. Menampilkan data kriteria pada tabel di halaman data kriteria
4. Menekan tombol "Batasan Kriteria" pada salah satu baris data kriteria yang dipilih	
	5. Mengambil batasan kriteria dari database

- 6. Menampilkan batasan kriteria pada tabel di halaman batasan kriteria
- 7. Menekan tombol “Kategori Peta”
- 8. Mengambil data kategori informasi peta dari *database*
- 9. Menampilkan data kategori informasi peta pada tabel di halaman kategori informasi
- 10. Menekan tombol “Nilai Kategori” pada salah satu baris data kategori informasi peta yang dipilih
- 11. Mengambil nilai kategori dari *database* berdasarkan data kategori informasi peta yang dipilih
- 12. Menampilkan nilai kategori informasi peta pada tabel di halaman nilai kategori informasi peta

SKENARIO NORMAL MELIHAT DETAIL NILAI KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|--|---|
| 13. Menekan tombol “Detail Nilai” pada salah satu baris data nilai kategori yang dipilih | 14. Mengambil detail nilai kategori dari <i>database</i> berdasarkan nilai kategori yang dipilih |
| | 15. Menampilkan detail nilai kategori informasi peta pada halaman detail nilai kategori |
| 16. Melihat detail nilai kategori informasi peta | |
| 17. Klik tombol “Back” | 18. Mengambil nilai kategori dari <i>database</i> |
| | 19. Menampilkan nilai kategori informasi peta pada tabel di halaman nilai kategori informasi peta |

SKENARIO NORMAL MENAMBAH NILAI KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|-----------------------------------|--|
| 13. Menekan tombol “Tambah Nilai” | |
| | 14. Menampilkan <i>form</i> isian tambah nilai kategori informasi peta |

- 15. Mengisi seluruh isian *form* tambah nilai kategori informasi peta
- 16. Klik *icon drawing* area atau garis atau titik pada peta
- 17. Menggambar area, titik atau garis baru
- 18. Klik 2 kali pada peta jika sudah selesai menggambar
- 19. Menampilkan hasil gambar pada *field* masukan “*Hasil Drawing*”
- 20. Menekan tombol “*Parse JSON*”
- 21. Mengambil koordinat gambar yang telah dibuat
- 22. Menampilkan koordinat gambar yang telah dibuat pada *field* masukan “*Geom*”
- 23. Menekan tombol “*Save*”
- 24. Menyimpan masukan data ke *database*
- 25. Menampilkan *alert* “Selamat! Nilai kategori sukses disimpan”
- 26. Mengambil nilai kategori informasi peta dari *database*
- 27. Menampilkan nilai kategori pada tabel di halaman nilai kategori informasi peta

SKENARIO NORMAL BATAL MENAMBAH AREA, TITIK ATAU GARIS

- | Aktor | Sistem |
|---|--|
| 20. Menekan tombol “ <i>Clear Map</i> ” | 21. Menghapus hasil <i>drawing</i> pada peta |

SKENARIO NORMAL MENGOSONGKAN ISIAN *FORM* TAMBAH NILAI KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|-------------------------------------|---|
| 23. Menekan tombol “ <i>Clear</i> ” | 24. Menampilkan <i>form</i> isian tambah nilai kategori informasi peta yang sudah dikosongkan |

SKENARIO NORMAL TAMBAH DATA KEMBALI KE HALAMAN NILAI KATEGORI INFORMASI PETA

- | Aktor | Sistem |
|---------------------------------|---|
| 16. Klik tombol “ <i>Back</i> ” | 17. Mengambil nilai kategori dari <i>database</i> |
| | 18. Menampilkan nilai kategori |

informasi peta pada tabel di halaman nilai kategori informasi peta

SKENARIO NORMAL MENGUBAH NILAI KATEGORI INFORMASI PETA

Aktor	Sistem
13. Menekan tombol “Ubah Nilai” pada baris data nilai kategori informasi peta yang akan diubah	14. Mengambil nilai kategori informasi peta yang dipilih dari <i>database</i> 15. Menampilkan nilai kategori informasi peta ke dalam <i>form</i> ubah nilai kategori informasi peta
16. Mengubah isian <i>form</i> ubah nilai kategori informasi peta	
17. Klik <i>icon drawing</i> area atau garis atau titik pada peta	
18. Menggambar area, titik atau garis baru	
19. Klik 2 kali pada peta jika sudah selesai menggambar	
21. Menekan tombol “Parse JSON”	20. Menampilkan hasil gambar pada <i>field</i> masukan “Hasil <i>Drawing</i> ”
	22. Mengambil koordinat gambar yang telah dibuat
	23. Menampilkan koordinat gambar yang telah dibuat pada <i>field</i> masukan “ <i>Geom</i> ”
24. Menekan tombol “Save Edit”	25. Menyimpan data yang diubah ke <i>database</i>
	26. Menampilkan <i>alert</i> “Selamat! Nilai kategori informasi peta sukses diubah”
	27. Mengambil nilai kategori informasi peta dari <i>database</i>
	28. Menampilkan nilai kategori informasi peta pada tabel di halaman nilai kategori informasi peta

SKENARIO NORMAL UBAH DATA BATAL MENAMBAH AREA, TITIK ATAU GARIS

Aktor	Sistem
21. Menekan tombol “ <i>Clear Map</i> ”	22. Menghapus hasil <i>drawing</i> pada peta

**SKENARIO NORMAL MENGEMBALIKAN NILAI KATEGORI
INFORMASI PETA SEPERTI SEMULA**

- | Aktor | Sistem |
|-------------------------------------|---|
| 24. Menekan tombol " <i>Clear</i> " | 25. Menampilkan <i>form</i> isian tambah nilai kategori informasi peta yang sudah dikosongkan |

**SKENARIO NORMAL UBAH DATA KEMBALI KE HALAMAN NILAI
KATEGORI INFORMASI PETA**

- | Aktor | Sistem |
|---------------------------------|---|
| 16. Klik tombol " <i>Back</i> " | 17. Mengambil nilai kategori dari <i>database</i> |
| | 18. Menampilkan nilai kategori informasi peta pada tabel di halaman nilai kategori informasi peta |

**SKENARIO NORMAL MENGUBAH NILAI KATEGORI INFORMASI
PETA TANPA KOORDINAT**

- | Aktor | Sistem |
|--|---|
| 13. Menekan tombol "Ubah Nilai" pada baris data nilai kategori informasi peta yang akan diubah | 14. Mengambil nilai kategori informasi peta yang dipilih dari <i>database</i> |
| | 15. Menampilkan nilai kategori informasi peta ke dalam <i>form</i> ubah nilai kategori informasi peta |
| 16. Mengubah isian <i>form</i> ubah nilai kategori informasi peta | |
| 17. Menekan tombol " <i>Save Edit</i> " | 18. Menyimpan data yang diubah ke <i>database</i> |
| | 19. Menampilkan <i>alert</i> "Selamat! Nilai kategori informasi peta sukses diubah" |
| | 20. Mengambil nilai kategori informasi peta dari <i>database</i> |
| | 21. Menampilkan nilai kategori informasi peta pada tabel di halaman nilai kategori informasi peta |

**SKENARIO NORMAL MENGEMBALIKAN NILAI KATEGORI
INFORMASI PETA SEPERTI SEMULA**

- | Aktor | Sistem |
|-------------------------------------|---|
| 17. Menekan tombol " <i>Clear</i> " | 18. Menampilkan <i>form</i> isian tambah nilai kategori informasi peta yang sudah |

dikosongkan

SKENARIO NORMAL UBAH DATA TANPA KOORDINAT KEMBALI KE HALAMAN NILAI KATEGORI INFORMASI PETA

Aktor	Sistem
16. Klik tombol "Back"	17. Mengambil nilai kategori dari <i>database</i>
	18. Menampilkan nilai kategori informasi peta pada tabel di halaman nilai kategori informasi peta

SKENARIO NORMAL MENGHAPUS NILAI KATEGORI INFORMASI PETA

Aktor	Sistem
13. Menekan tombol "Hapus Nilai" pada baris data nilai kategori informasi peta yang akan dihapus	14. Menampilkan <i>alert</i> "Apakah Anda yakin akan menghapus data ini?" disertai dengan tombol OK dan <i>Cancel</i>
15. Menekan tombol "OK" pada <i>alert</i>	16. Menghapus nilai kategori informasi peta yang dipilih dari <i>database</i>
	17. Menampilkan <i>alert</i> "Selamat! Nilai Kategori Informasi Peta sukses dihapus"
	18. Mengambil nilai kategori informasi peta dari <i>database</i>
	19. Menampilkan nilai kategori informasi peta pada tabel di halaman nilai kategori informasi peta

SKENARIO NORMAL BATAL MENGHAPUS NILAI KATEGORI INFORMASI PETA

Aktor	Sistem
15. Menekan tombol "Cancel" pada <i>alert</i>	16. Menampilkan nilai kategori informasi peta pada halaman nilai kategori informasi peta

SKENARIO ALTERNATIF MENAMBAH NILAI KATEGORI INFORMASI PETA

Jika aktor tidak mengisi *form* nilai kategori informasi peta secara lengkap

Aktor	Sistem
15. Mengisi isian <i>form</i> tambah nilai	

- kategori informasi peta tidak lengkap
16. Menekan tombol "Save"
 17. Memeriksa masukan nilai kategori informasi peta
 18. Menampilkan *message* dengan menunjuk isian mana yang tidak diisi pada *form* isian nilai kategori informasi peta. Tulisan akan muncul dibawah masing-masing *field* masukan

SKENARIO ALTERNATIF MENGUBAH NILAI KATEGORI INFORMASI PETA

Jika aktor tidak mengisi *form* ubah nilai kategori informasi peta secara lengkap

Aktor	Sistem
16. Mengubah isian <i>form</i> ubah nilai kategori informasi peta secara tidak lengkap	
17. Menekan tombol "Save Edit"	18. Memeriksa ubahan nilai kategori informasi peta
	19. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> ubah nilai kategori informasi peta. Tulisan akan muncul dibawah masing-masing <i>field</i> inputan

A.12 *Usecase* Skenario Mengelola Biodata

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola biodata dapat dilihat pada Tabel 12.

Tabel 12 *Usecase* Skenario Mengelola Biodata

Nomor <i>Usecase</i>	UC-13
Nama	Mengelola Biodata
Aktor	Pengambil Keputusan
<i>Pre Condition</i>	Pengambil Keputusan harus memilih menu Biodata
<i>Post Condition</i>	Pengambil Keputusan berhasil mengubah biodata

SKENARIO NORMAL MENGUBAH BIODATA

Aktor	Sistem
1. Memilih menu Biodata	

- | | |
|---|---|
| <ul style="list-style-type: none"> 4. Mengubah isian <i>form</i> biodata 5. Menekan tombol “<i>Save Edit</i>” | <ul style="list-style-type: none"> 2. Mengambil biodata Pengambil Keputusan dari <i>database</i> 3. Menampilkan biodata pada <i>form</i> biodata Pengambil Keputusan 6. Menyimpan data yang diubah ke <i>database</i> 7. Menampilkan <i>alert</i> “Selamat! Data <i>user</i> sukses diubah” 8. Mengambil biodata Pengambil Keputusan dari <i>database</i> 9. Menampilkan biodata pada <i>form</i> biodata Pengambil Keputusan |
|---|---|

SKENARIO NORMAL HAPUS FOTO USER

- | Aktor | Sistem |
|--|--|
| <ul style="list-style-type: none"> 4. Klik tombol “Hapus foto” 6. Menekan tombol “OK” pada <i>alert</i> | <ul style="list-style-type: none"> 5. Menampilkan <i>alert</i> “Apakah Anda yakin akan menghapus foto ini?” disertai dengan tombol OK dan <i>Cancel</i> 7. Menghapus foto dari data <i>user</i> di <i>database</i> 8. Mengambil biodata Pengambil Keputusan dari <i>database</i> 9. Menampilkan biodata pada <i>form</i> biodata Pengambil Keputusan |

SKENARIO NORMAL BATAL MENGHAPUS FOTO USER

- | Aktor | Sistem |
|---|--|
| <ul style="list-style-type: none"> 7. Menekan tombol “<i>Cancel</i>” pada <i>alert</i> | <ul style="list-style-type: none"> 8. Mengambil biodata Pengambil Keputusan dari <i>database</i> 9. Menampilkan biodata pada <i>form</i> biodata Pengambil Keputusan |
-

SKENARIO NORMAL MENGEMBALIKAN BIODATA SEPERTI SEMULA

Aktor	Sistem
5. Menekan tombol " <i>Clear</i> "	6. Menampilkan biodata pada <i>form</i> biodata Pengambil Keputusan yang belum diubah

SKENARIO ALTERNATIF MENGUBAH BIODATA

Jika aktor tidak mengisi *form* biodata secara lengkap

Aktor	Sistem
4. Mengubah isian <i>form</i> biodata secara tidak lengkap	
5. Menekan tombol " <i>Save Edit</i> "	6. Memeriksa ubahan biodata
	7. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> biodata Pengambil Keputusan. Tulisan akan muncul dibawah masing-masing <i>field</i> masukan

Jika aktor mengisi masukan angka dengan huruf

Aktor	Sistem
4. Mengubah isian berupa masukan angka pada <i>form</i> biodata dengan huruf	
5. Menekan tombol " <i>Save Edit</i> "	6. Memeriksa ubahan biodata
	7. Menampilkan <i>message</i> dengan menunjuk isian mana yang salah. Tulisan akan muncul "input harus berupa angka" pada <i>form</i> biodata

Jika *personal number* sudah terdapat pada *database*

4. Mengisi inputan <i>personal number</i> yang sama dengan data yang ada pada <i>database</i>	
5. Menekan tombol " <i>Save Edit</i> "	6. Memeriksa ubahan biodata ke <i>database</i>
	7. Menampilkan <i>alert</i> " <i>Personal number duplicate entry</i> " pada <i>form</i> biodata

A.13 *Usecase* Skenario Melihat Data ATM

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *usecase* skenario melihat data ATM dapat dilihat pada Tabel 13.

Tabel 13 *Usecase* Skenario Melihat Data ATM

Nomor <i>Usecase</i>	UC-15
Nama	Melihat Data ATM
Aktor	Nasabah
<i>Pre Condition</i>	Nasabah memilih menu ATM BRI
<i>Post Condition</i>	Nasabah berhasil melihat data ATM BRI
SKENARIO NORMAL MELIHAT DATA ATM	
Aktor	Sistem
1. Memilih menu ATM BRI	2. Mengambil data kategori ATM BRI dari <i>database</i>
	3. Menampilkan data ATM pada tabel di halaman data ATM
SKENARIO NORMAL MELIHAT DETAIL ATM	
Aktor	Sistem
4. Menekan tombol “Detail ATM” pada baris data yang ingin dilihat detailnya	5. Mengambil detail ATM yang dipilih dari <i>database</i>
	6. Menampilkan detail ATM ke dalam peta di halaman detail ATM
7. Klik tombol “Back”	8. Mengambil data kategori ATM BRI dari <i>database</i>
	9. Menampilkan data ATM pada tabel di halaman data ATM

A.14 *Usecase* Skenario Mengubah Akun *User*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengubah akun *user* dapat dilihat pada Tabel 14.

Tabel 14 *Usecase* Skenario Mengubah Akun *User*

Nomor <i>Usecase</i>	UC-16
Nama	Mengubah Akun <i>User</i>
Aktor	Admin dan Pengambil Keputusan

Pre Condition

Admin dan Pengambil Keputusan harus sudah melakukan *login* dan masuk ke dalam sistem

Post Condition

Admin dan Pengambil Keputusan berhasil mengubah akun miliknya

SKENARIO NORMAL MENGUBAH AKUN USER

Aktor	Sistem
1. Klik nama pengguna di bagian pojok kanan atas halaman	
2. Menekan tombol "Account Setting"	
	3. Menampilkan pop up berisi form ubah akun user
4. Mengisi seluruh isian form ubah akun	
5. Menekan tombol "Save"	
	6. Menyimpan ubahan data akun <i>user</i> ke <i>database</i>
	7. Menampilkan halaman <i>dashboard</i>

SKENARIO NORMAL BATAL MENGUBAH AKUN USER

Aktor	Sistem
5. Menekan tombol "Close"	
	6. Menampilkan halaman sebelumnya

SKENARIO ALTERNATIF MENGUBAH AKUN USER

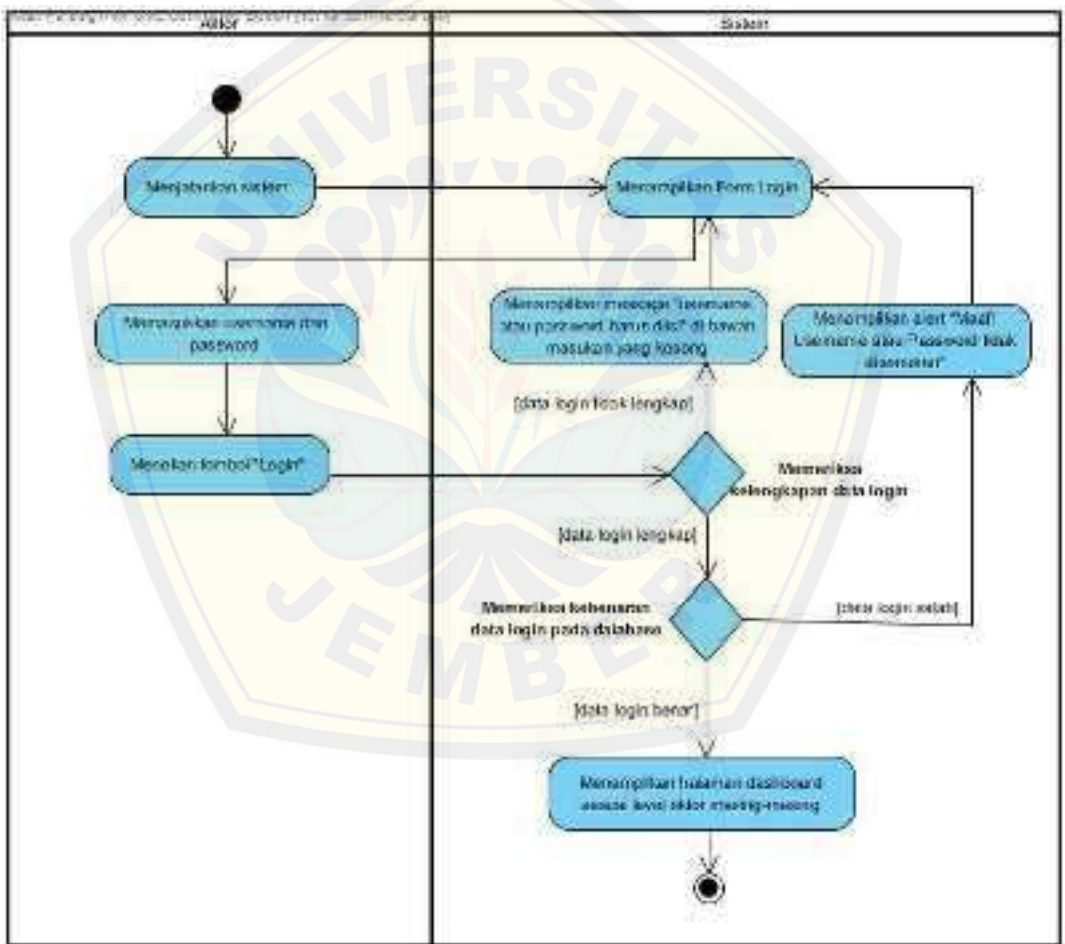
Jika aktor tidak mengisi *form* ubah akun secara lengkap

Aktor	Sistem
4. Mengisi isian <i>form</i> ubah akun secara tidak lengkap	
5. Menekan tombol "Save"	
	6. Memeriksa ubahan akun
	7. Menampilkan <i>message</i> dengan menunjuk isian mana yang tidak diisi pada <i>form</i> ubah akun. Tulisan akan muncul dibawah masing-masing masukan

B. Activity Diagram

B.1 Activity Diagram Login

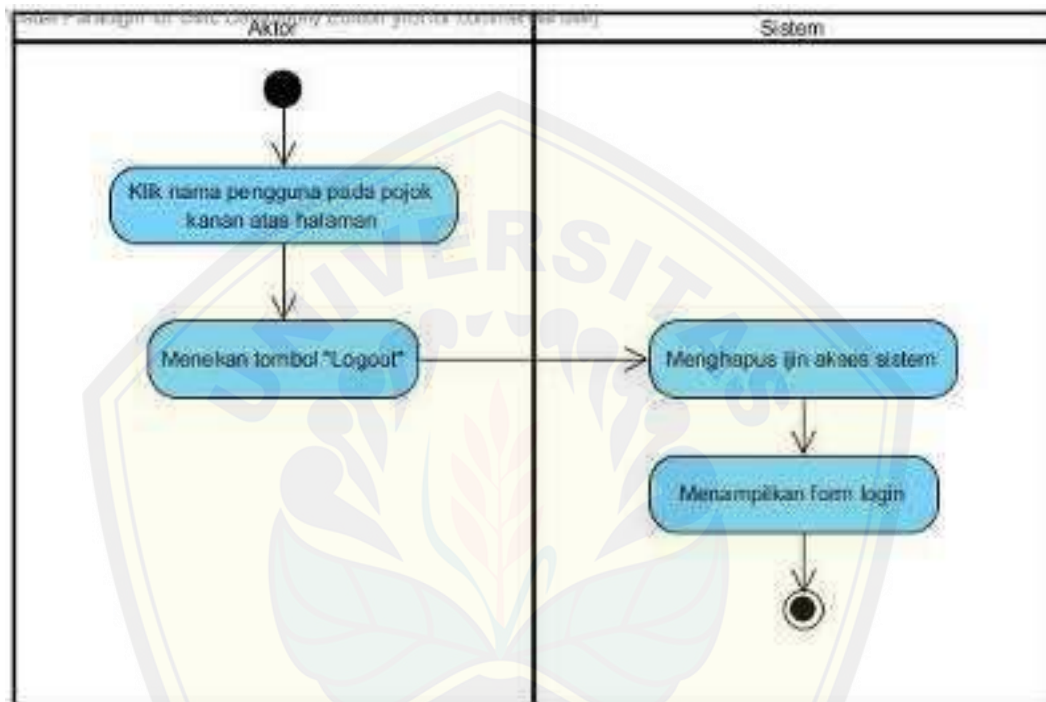
Activity diagram login dapat dilakukan oleh administrator dan decision maker. Activity diagram login menjelaskan tentang bagaimana administrator dan decision maker dapat masuk ke dalam sistem Group Decision Support System (GDSS) penentuan lokasi penempatan ATM. Activity diagram login dapat dilihat pada Gambar 1.



Gambar 1 Activity Diagram Login

B.2 Activity Diagram Logout

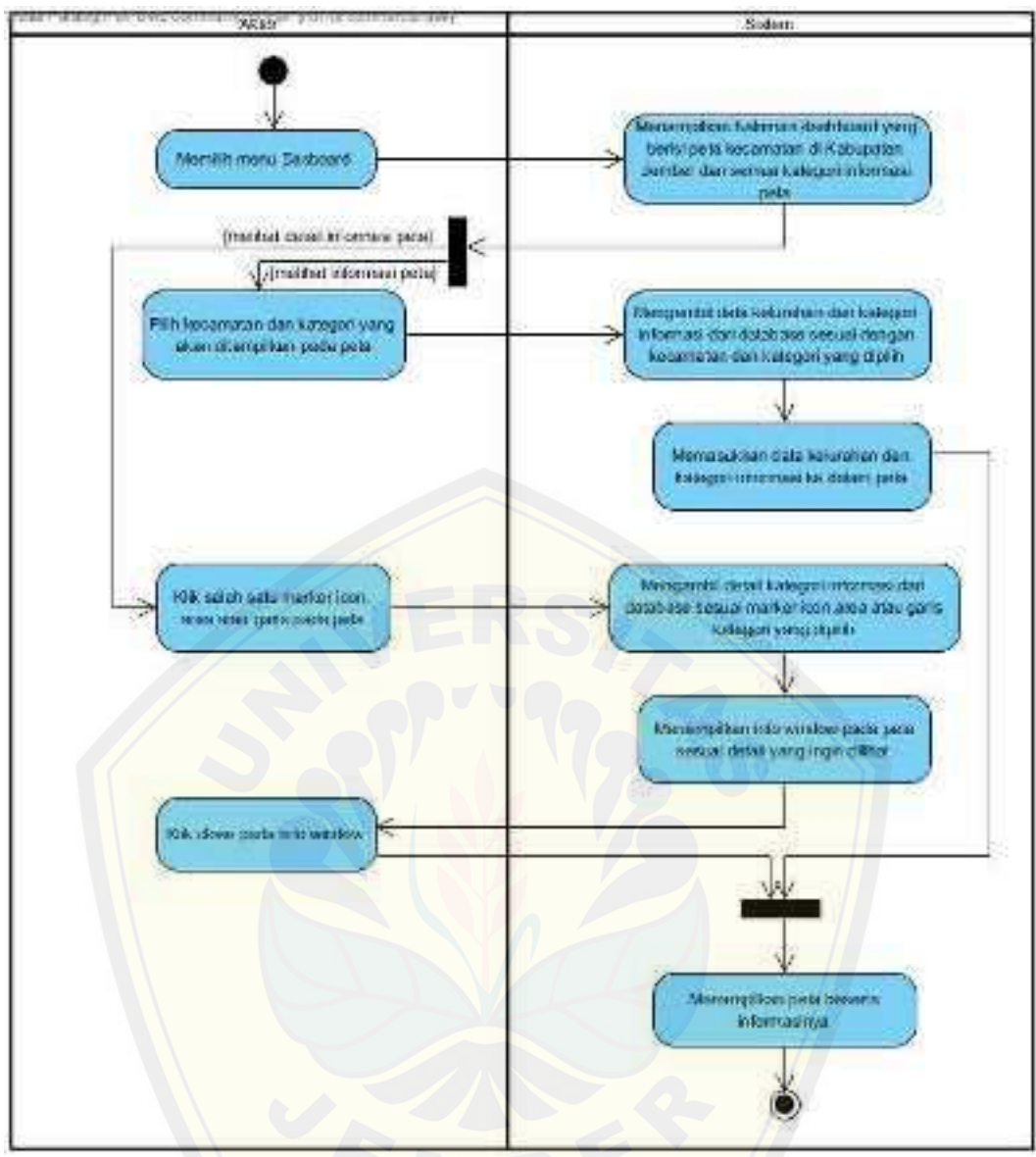
Activity diagram *logout* dapat dilakukan oleh *administrator* dan *decision maker*. Activity diagram *logout* menjelaskan tentang bagaimana *administrator* dan *decision maker* dapat keluar dari sistem *Group Decision Support System (GDSS)* penentuan lokasi penempatan ATM. Activity diagram *logout* dapat dilihat pada Gambar 2.



Gambar 2 Activity Diagram Logout

B.3 Activity Diagram Melihat Informasi Peta

Activity diagram melihat informasi peta dapat dilakukan oleh *administrator*, *decision maker* dan *user*. Activity diagram melihat informasi peta menjelaskan tentang bagaimana *administrator*, *decision maker* dan *user* menggunakan *Group Decision Support System (GDSS)* penentuan lokasi penempatan ATM untuk melihat informasi persebaran ATM BRI dan kategori lainnya yang ditampilkan dalam peta. Activity diagram melihat informasi peta dapat dilihat pada Gambar 3.



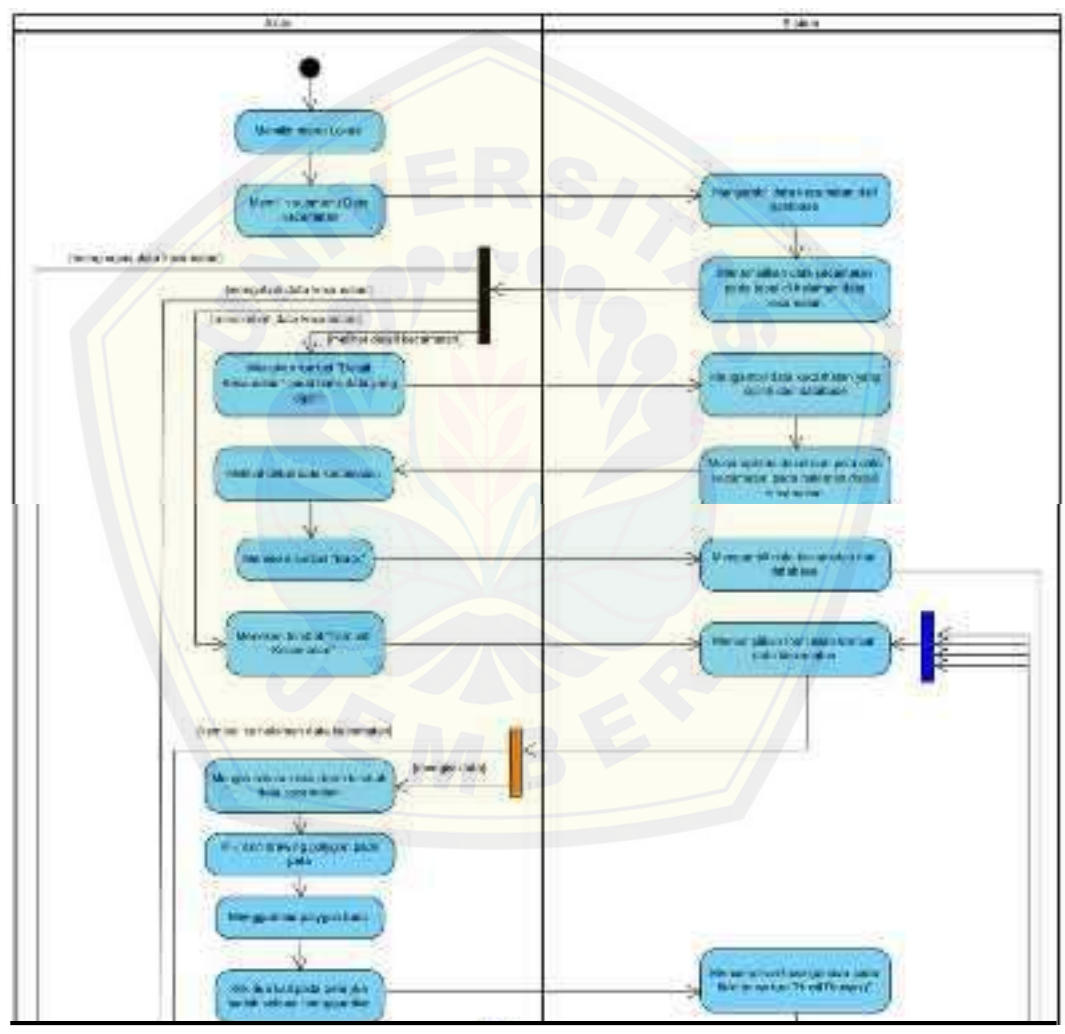
Gambar 3 Activity Diagram Melihat Informasi Peta

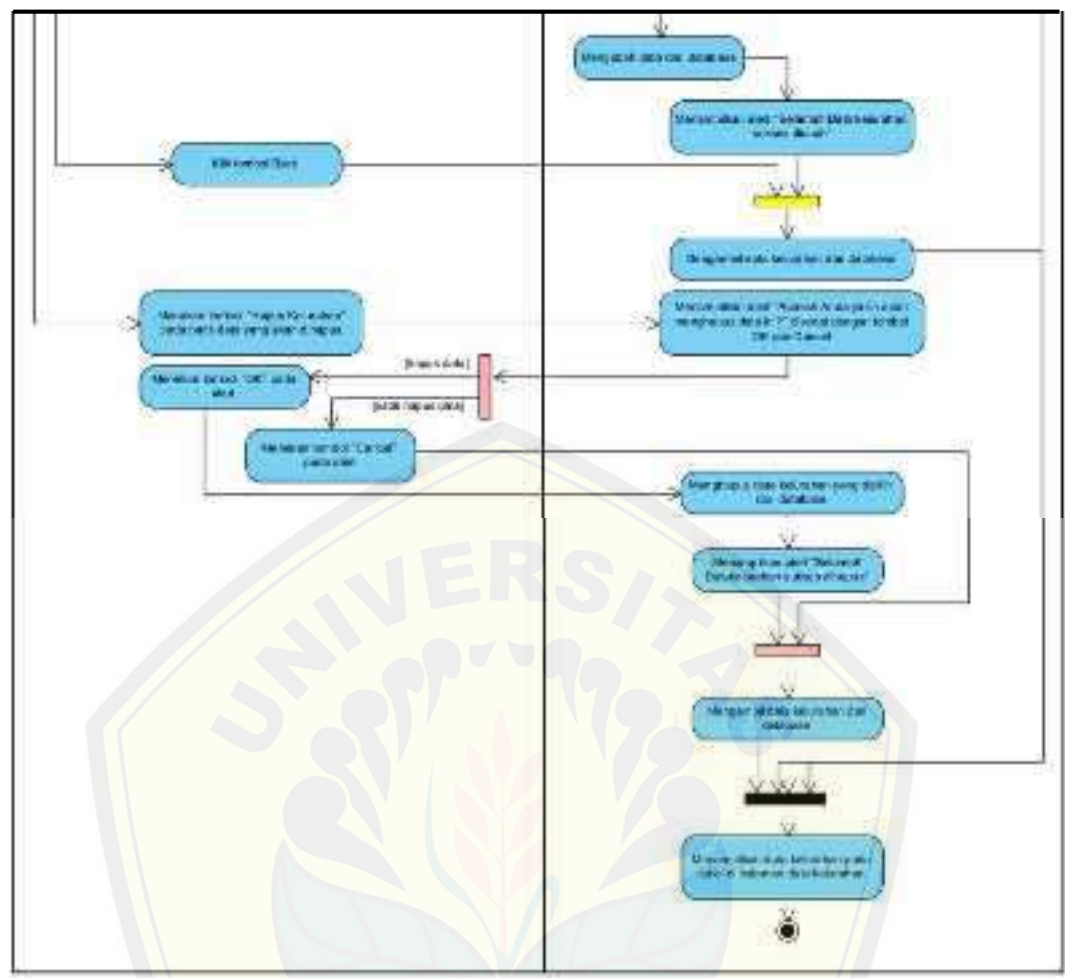
B.4 Activity Diagram Mengelola Data User

Activity diagram mengelola data user dapat dilakukan oleh administrator. Activity diagram mengelola data user menjelaskan tentang administrator menggunakan Group Decision Support System (GDSS) penentuan lokasi penempatan ATM untuk melihat, menambah, mengubah dan menghapus data user. Activity diagram mengelola data user dapat dilihat pada Gambar 4.

B.5 Activity Diagram Mengelola Data Kecamatan

Activity diagram mengelola data kecamatan dapat dilakukan oleh administrator. Activity diagram mengelola data kecamatan menjelaskan tentang administrator menggunakan Group Decision Support System (GDSS) penentuan lokasi penempatan ATM untuk melihat detail kecamatan dalam bentuk peta, menambah, mengubah dan menghapus data kecamatan. Activity diagram mengelola data kecamatan dapat dilihat pada Gambar 5.

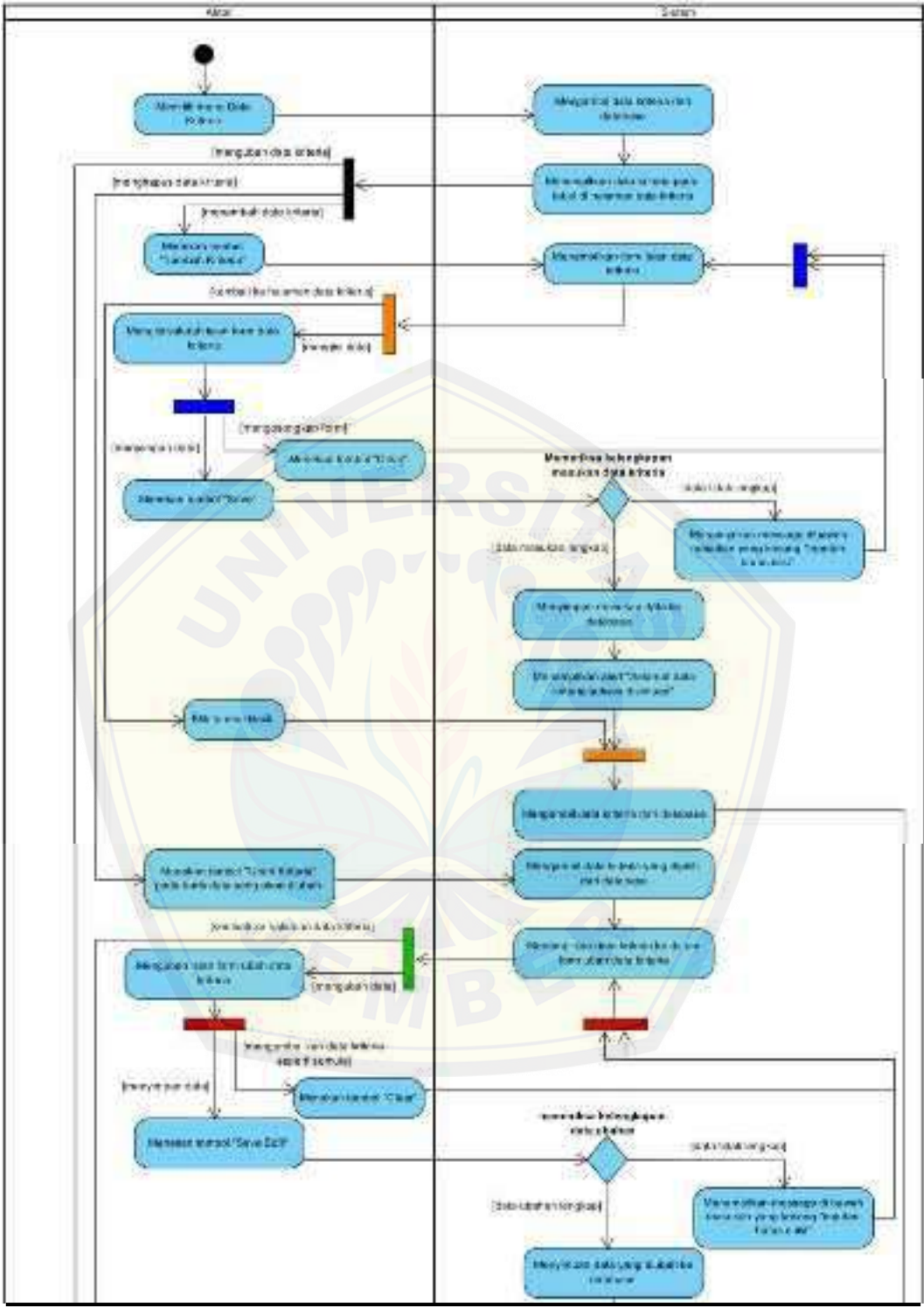


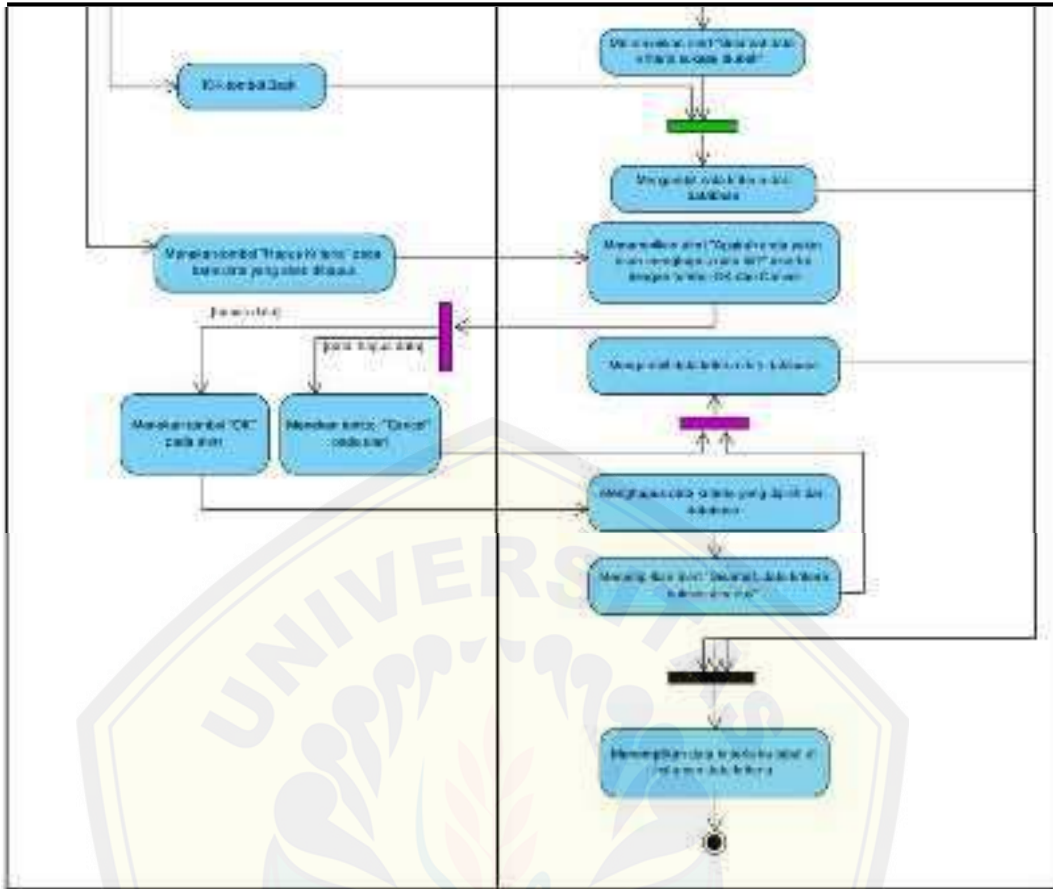


Gambar 6 Activity Diagram Mengelola Data Kelurahan

B.7 Activity Diagram Mengelola Data Kriteria

Activity diagram mengelola data kriteria dapat dilakukan oleh *administrator*. Activity diagram mengelola data kriteria menjelaskan tentang *administrator* menggunakan *Group Decision Support System (GDSS)* penentuan lokasi penempatan ATM untuk melihat, menambah, mengubah dan menghapus data kriteria yang digunakan dalam proses pengambilan keputusan. Activity diagram mengelola data kriteria dapat dilihat pada Gambar 7.

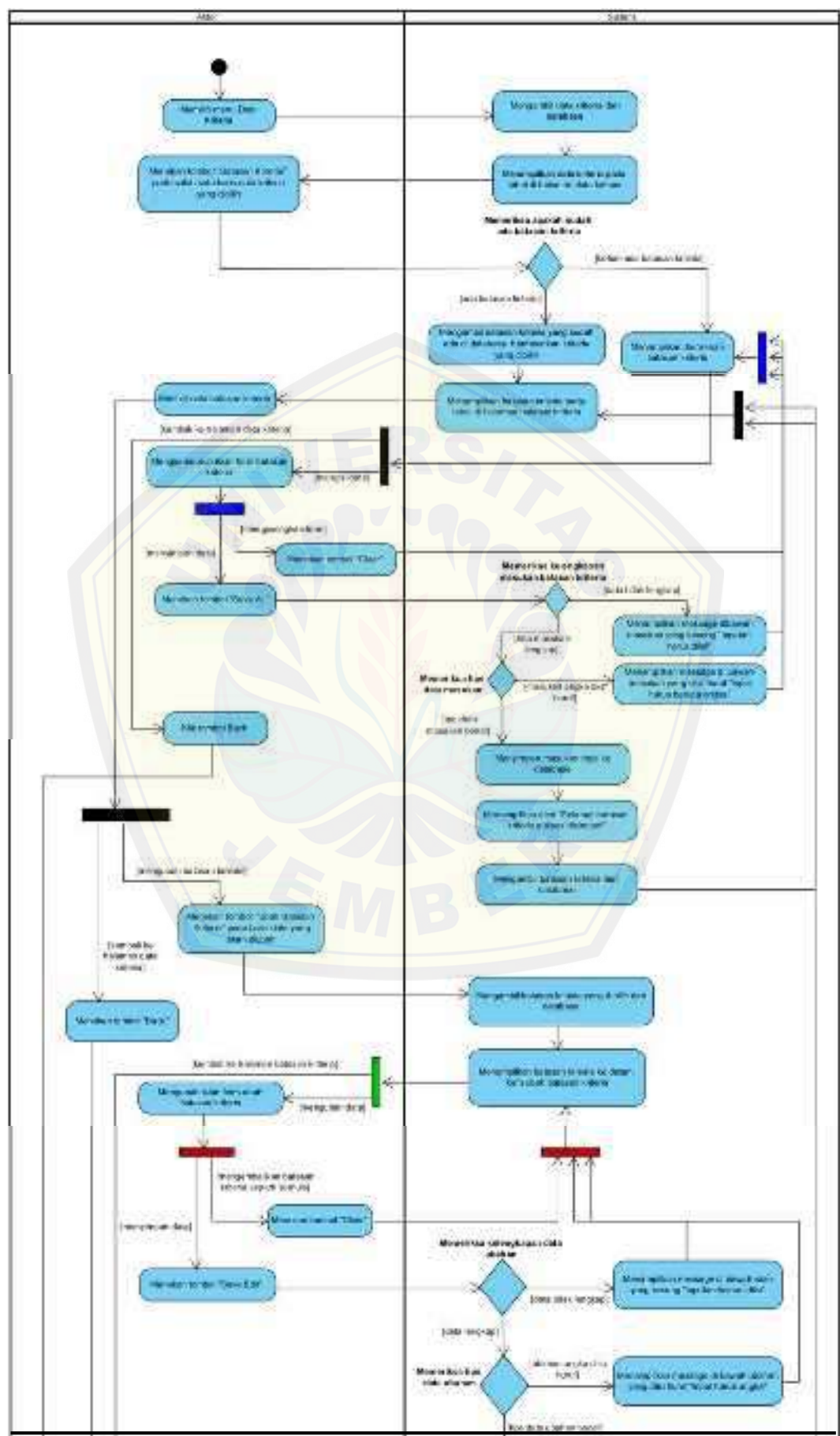


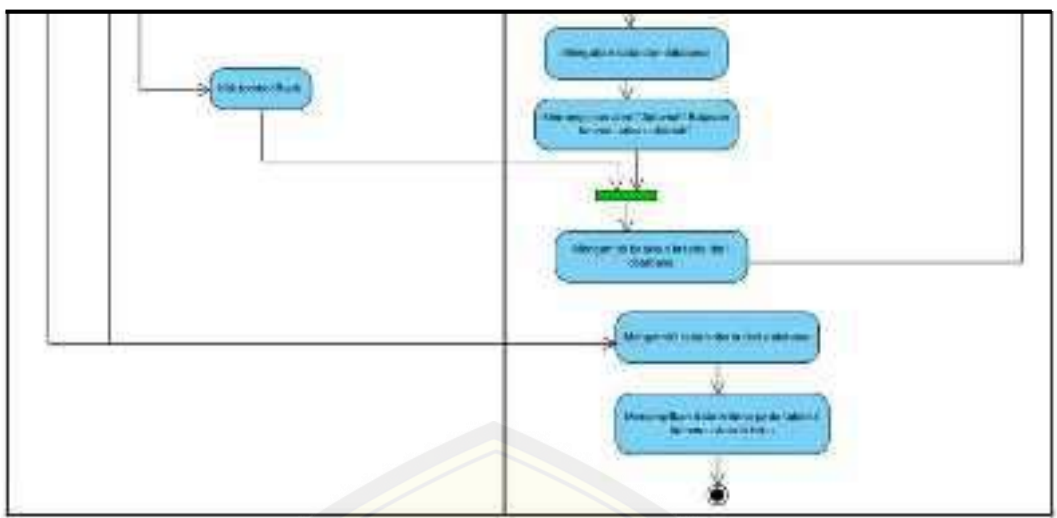


Gambar 7 Activity Diagram Mengelola Data Kriteria

B.8 Activity Diagram Mengelola Batasan Kriteria

Activity diagram mengelola batasan kriteria dapat dilakukan oleh administrator. Activity diagram mengelola batasan kriteria menjelaskan tentang administrator menggunakan Group Decision Support System (GDSS) penentuan lokasi penempatan ATM untuk melihat, menambah dan mengubah batasan kriteria berdasarkan data-data kriteria yang sudah dimasukkan. Activity diagram mengelola batasan kriteria dapat dilihat pada Gambar 8.

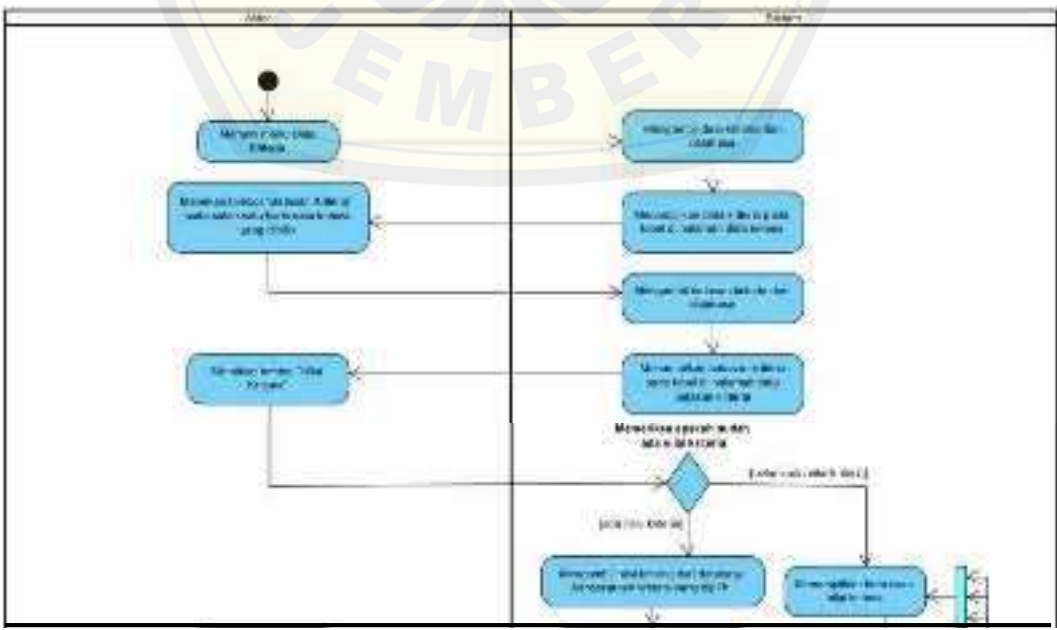


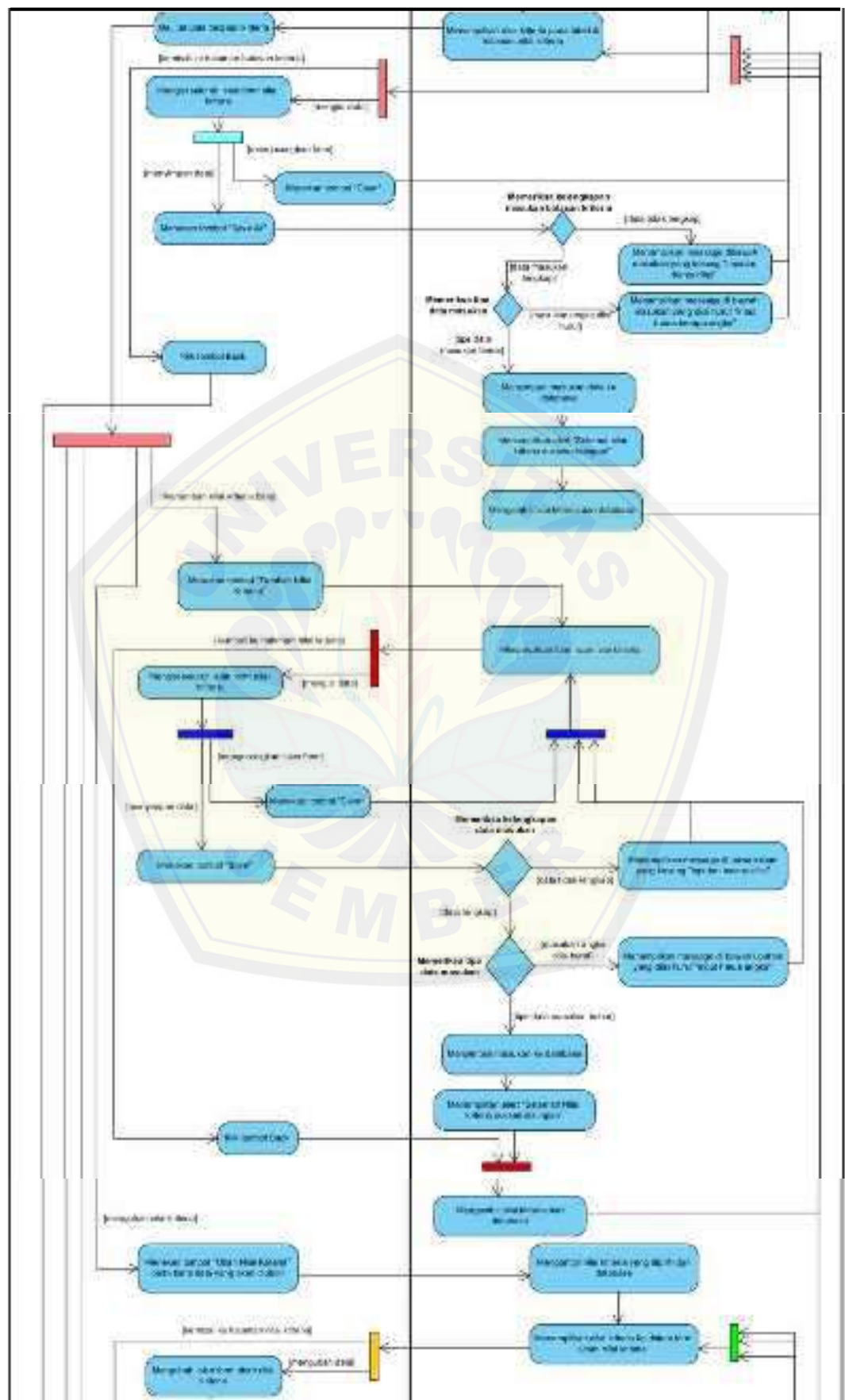


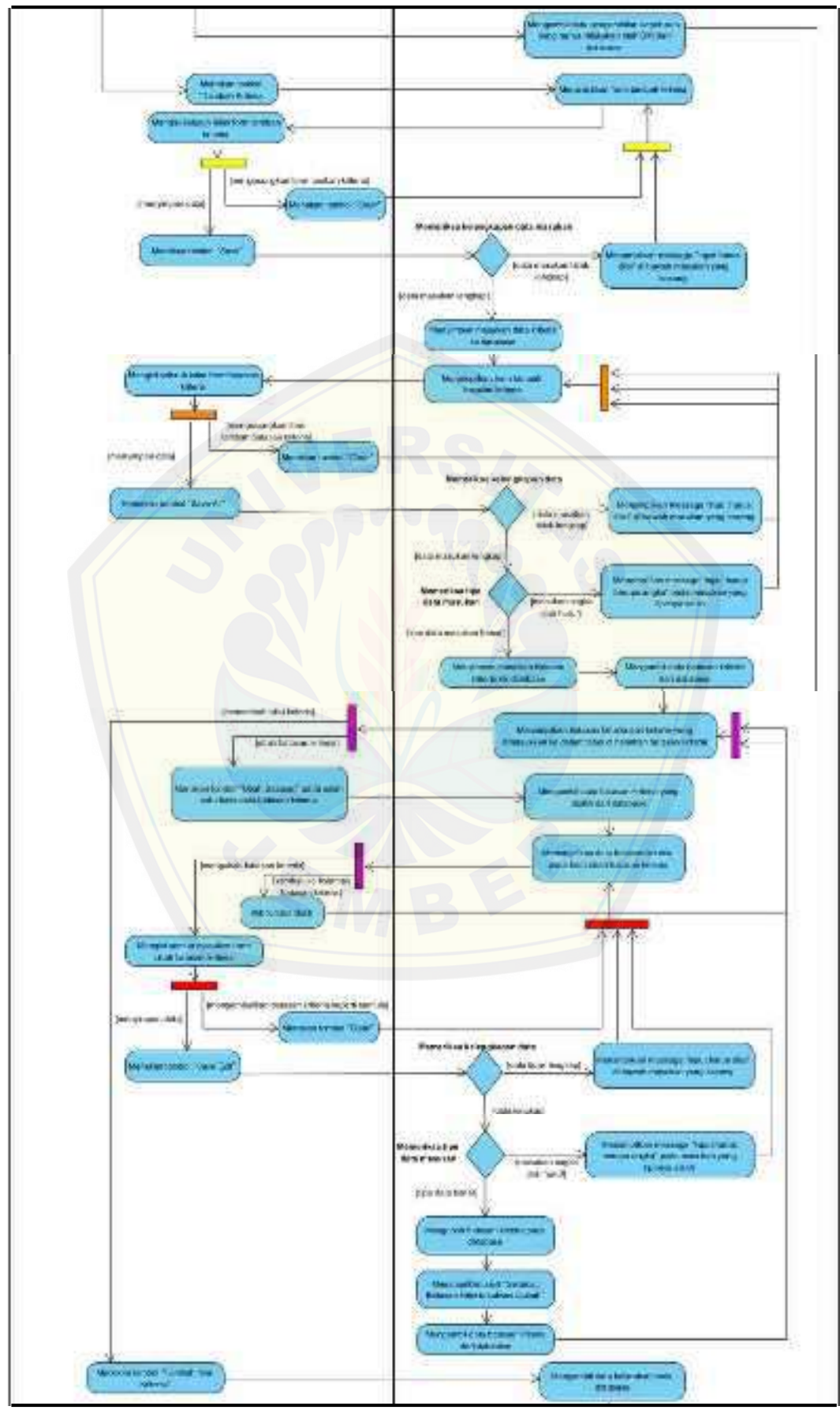
Gambar 8 Activity Diagram Mengelola Batasan Kriteria

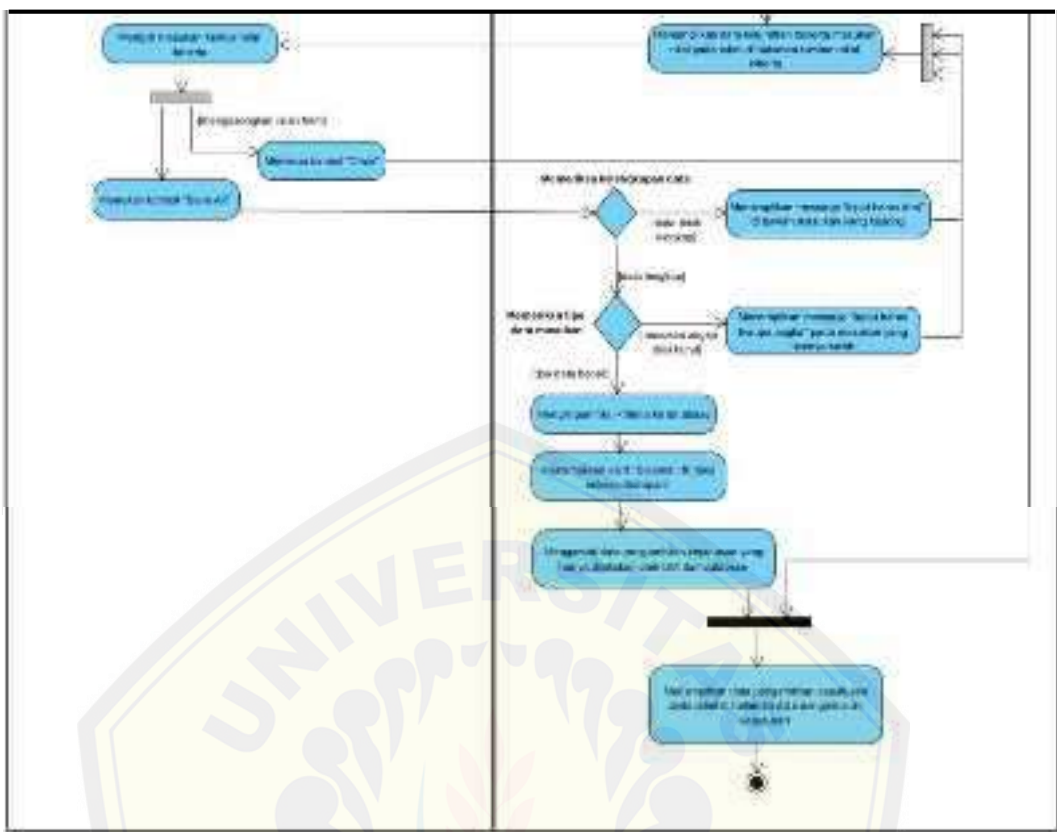
B.9 Activity Diagram Mengelola Nilai Kriteria

Activity diagram mengelola nilai kriteria dapat dilakukan oleh *administrator*. Activity diagram mengelola nilai kriteria menjelaskan tentang *administrator* menggunakan *Group Decision Support System (GDSS)* penentuan lokasi penempatan ATM untuk melihat, menambah, mengubah dan menghapus nilai kriteria berdasarkan data kriteria dan batasan kriteria yang sudah dimasukkan. Activity diagram mengelola nilai kriteria dapat dilihat pada Gambar 9.





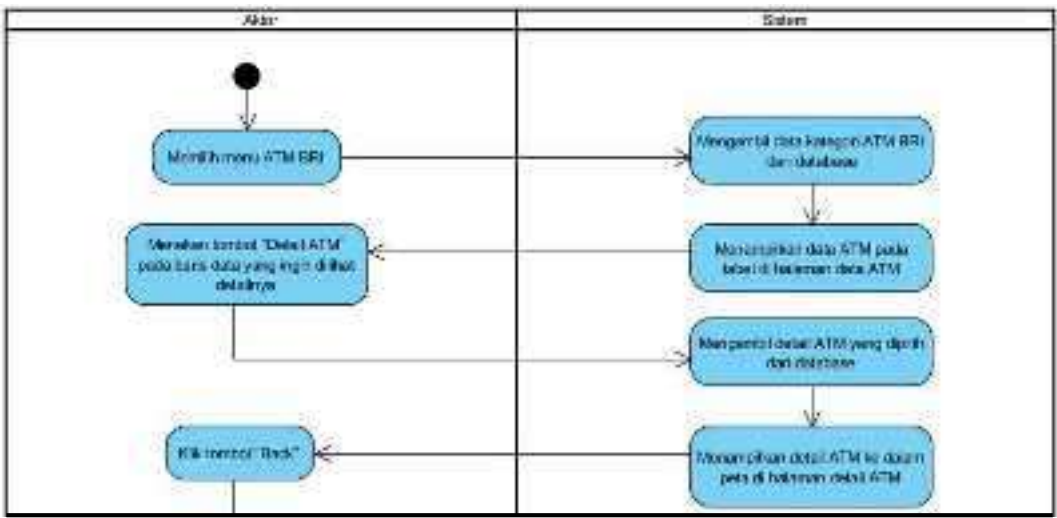


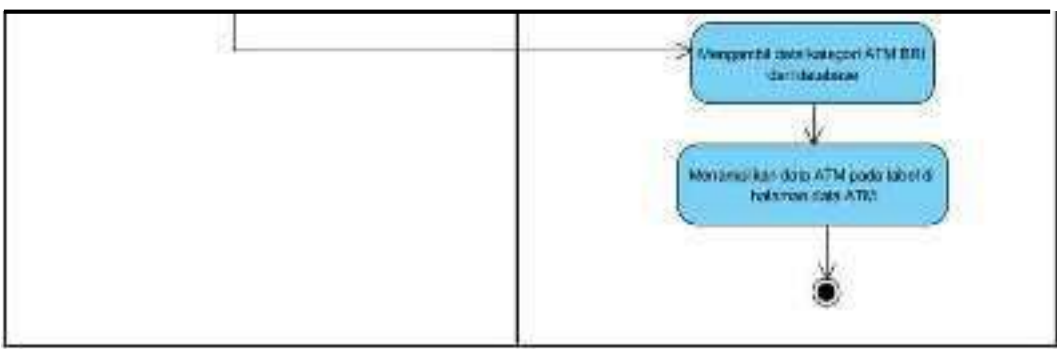


Gambar 13 Activity Diagram Memroses Pengambilan Keputusan

B.14 Activity Diagram Melihat Data ATM

Activity diagram melihat data ATM dapat dilakukan oleh user. Activity diagram melihat data ATM menjelaskan tentang user menggunakan Group Decision Support System (GDSS) penentuan lokasi penempatan ATM untuk melihat data ATM BRI yang ada di Kabupaten Jember. Activity diagram melihat data ATM dapat dilihat pada Gambar 14.

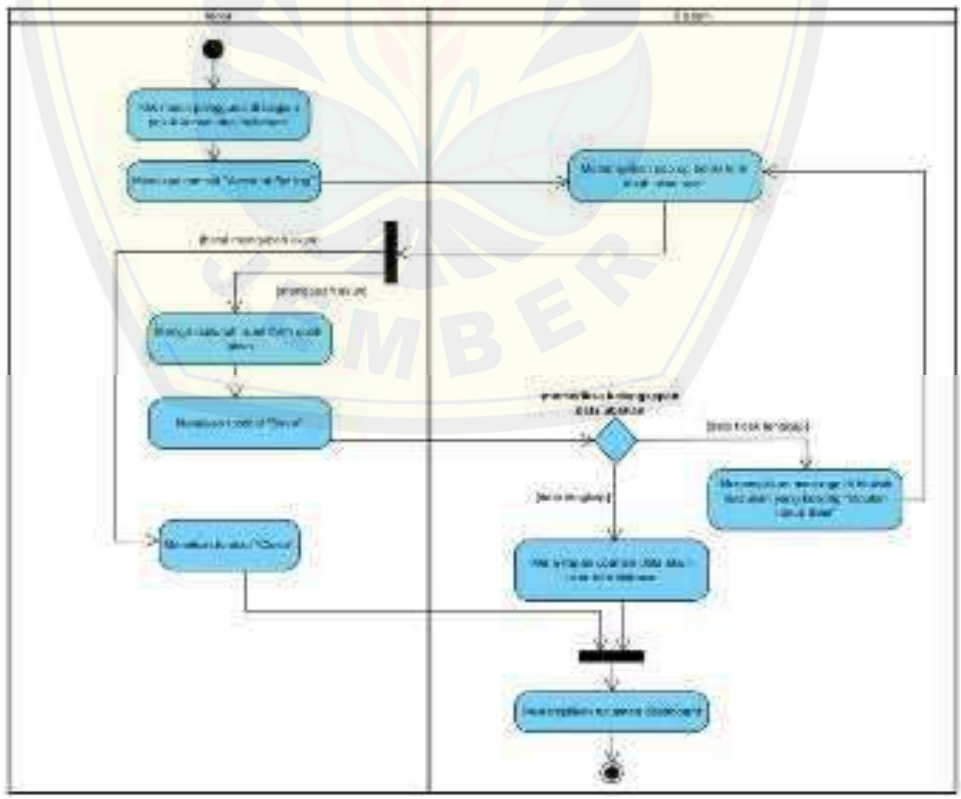




Gambar 14 Activity Diagram Melihat Data ATM

B.15 Activity Diagram Mengubah Akun User

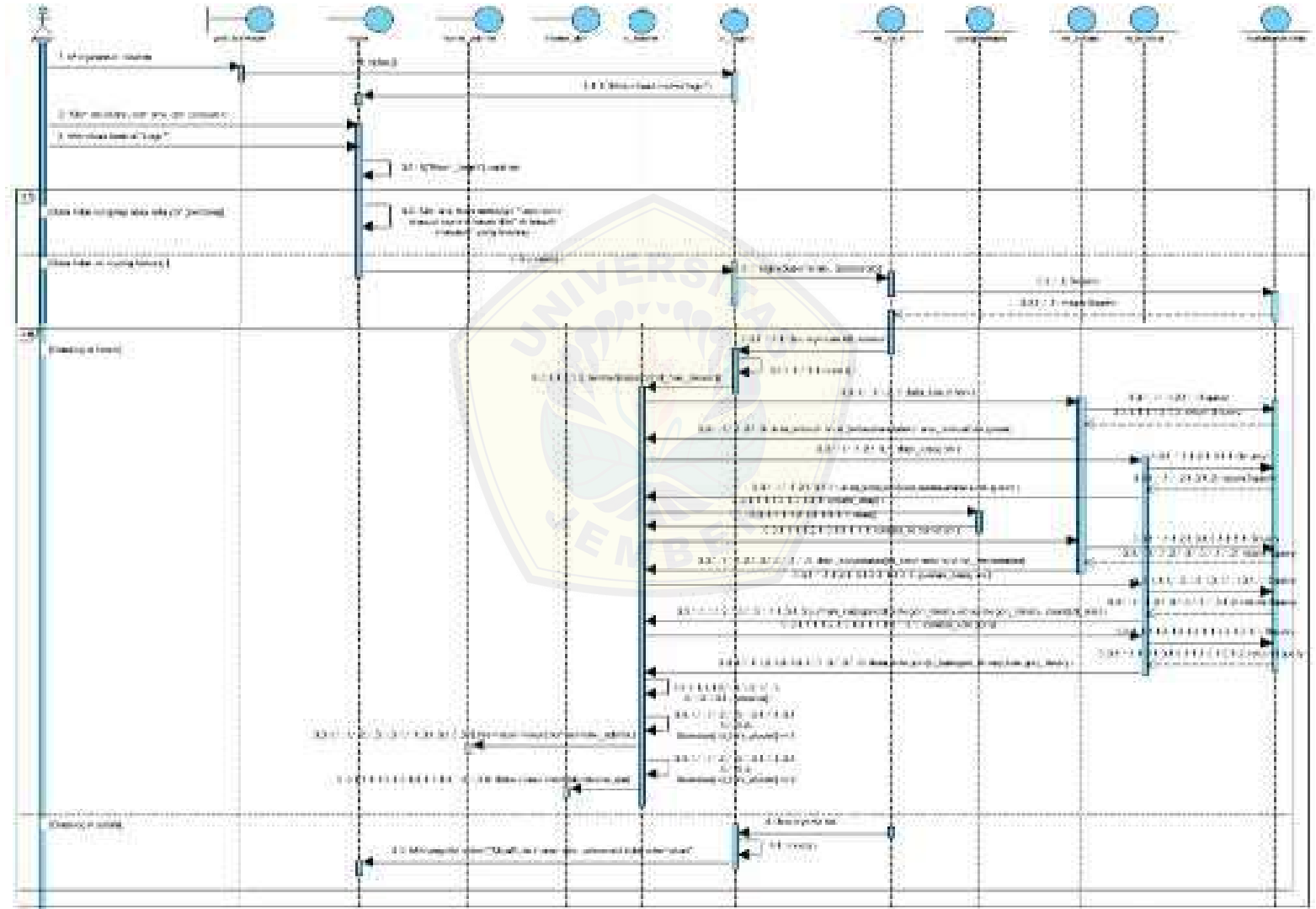
Activity diagram mengubah akun user dapat dilakukan oleh administrator dan decision maker. Activity diagram mengubah akun user menjelaskan tentang administrator dan decision maker menggunakan Group Decision Support System (GDSS) penentuan lokasi penempatan ATM untuk mengubah data akun user. Activity diagram mengubah akun user dapat dilihat pada Gambar 15.

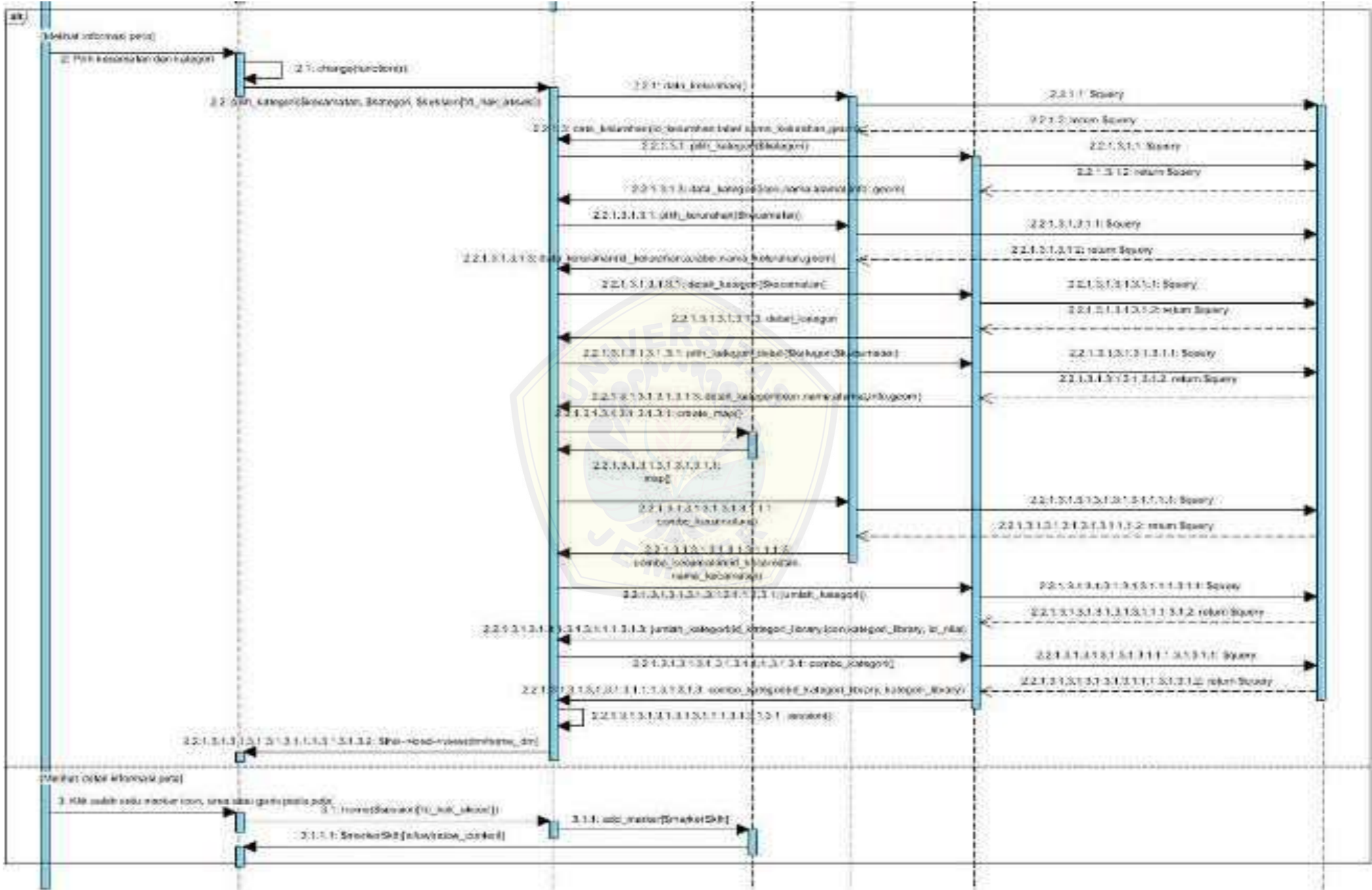


Gambar 15 Activity Diagram Mengubah Akun User

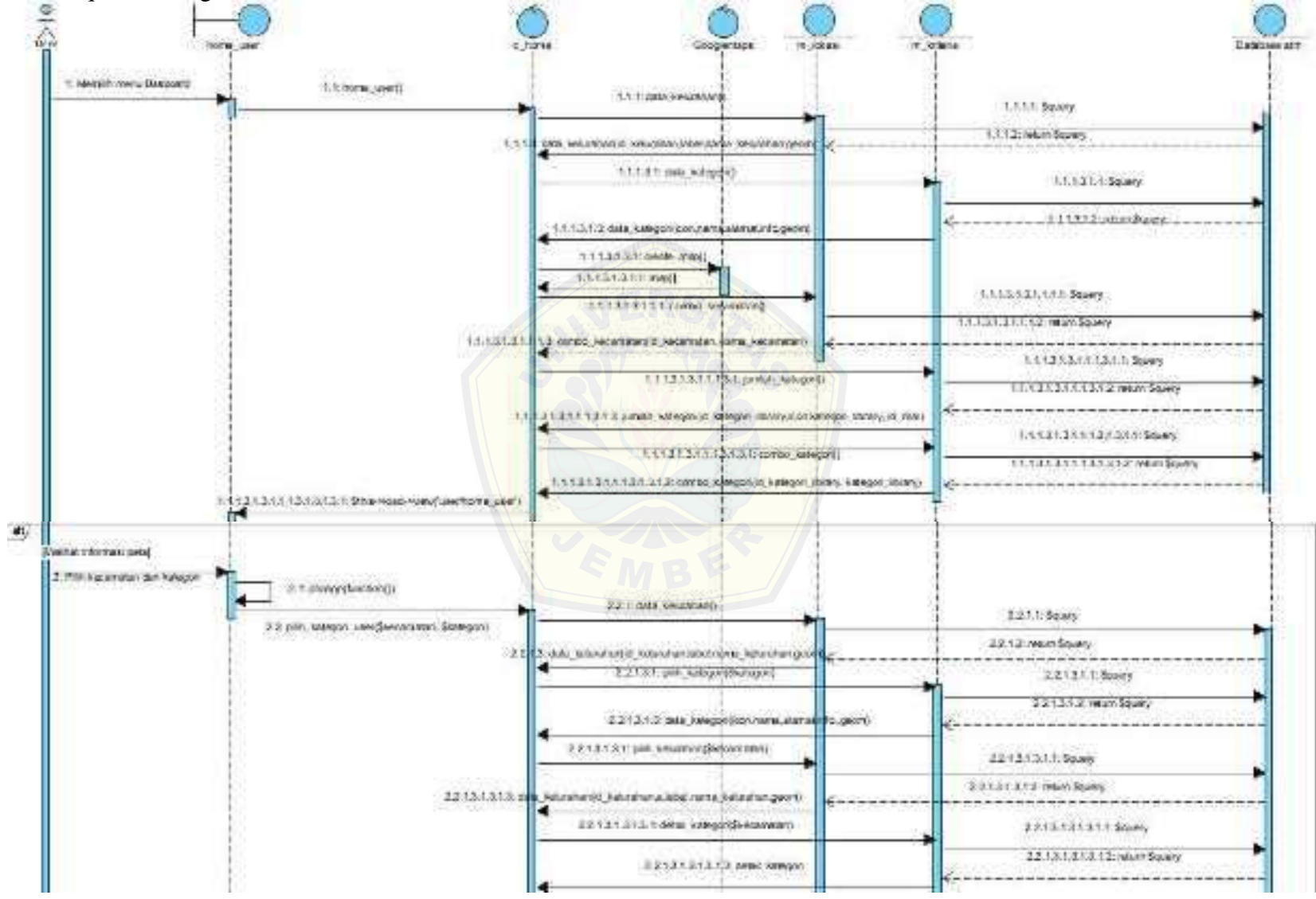
C. Sequence Diagram

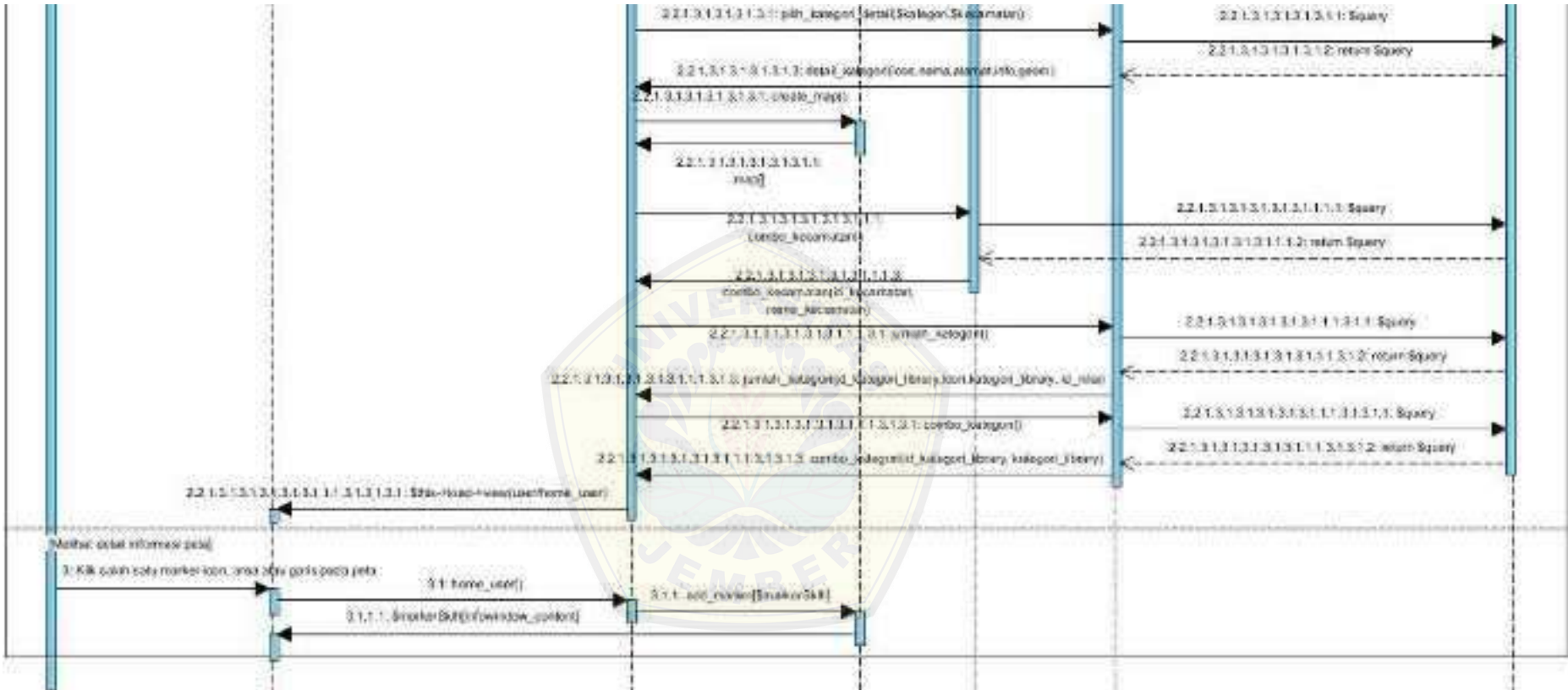
C.1 Sequence Diagram Login

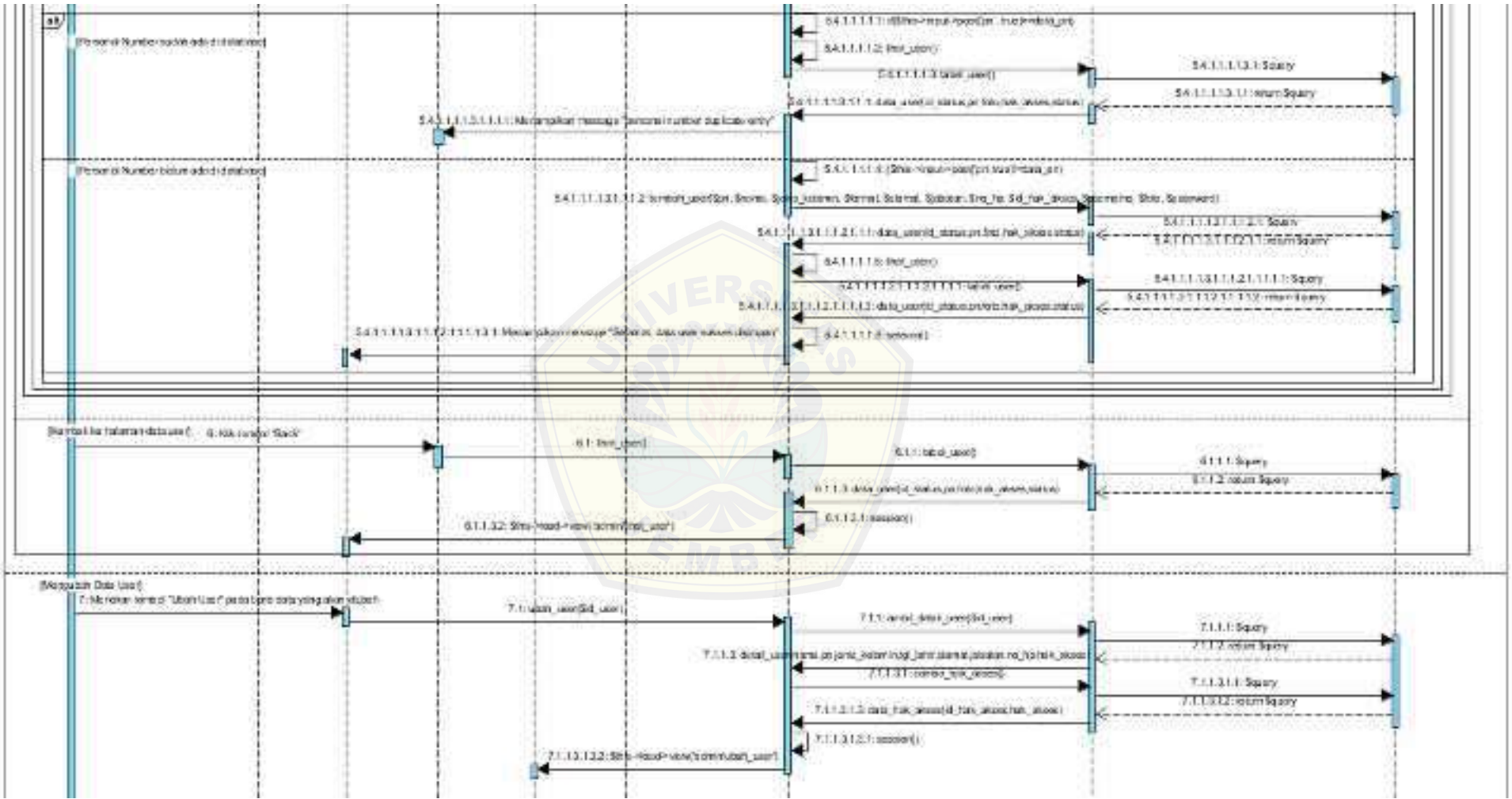


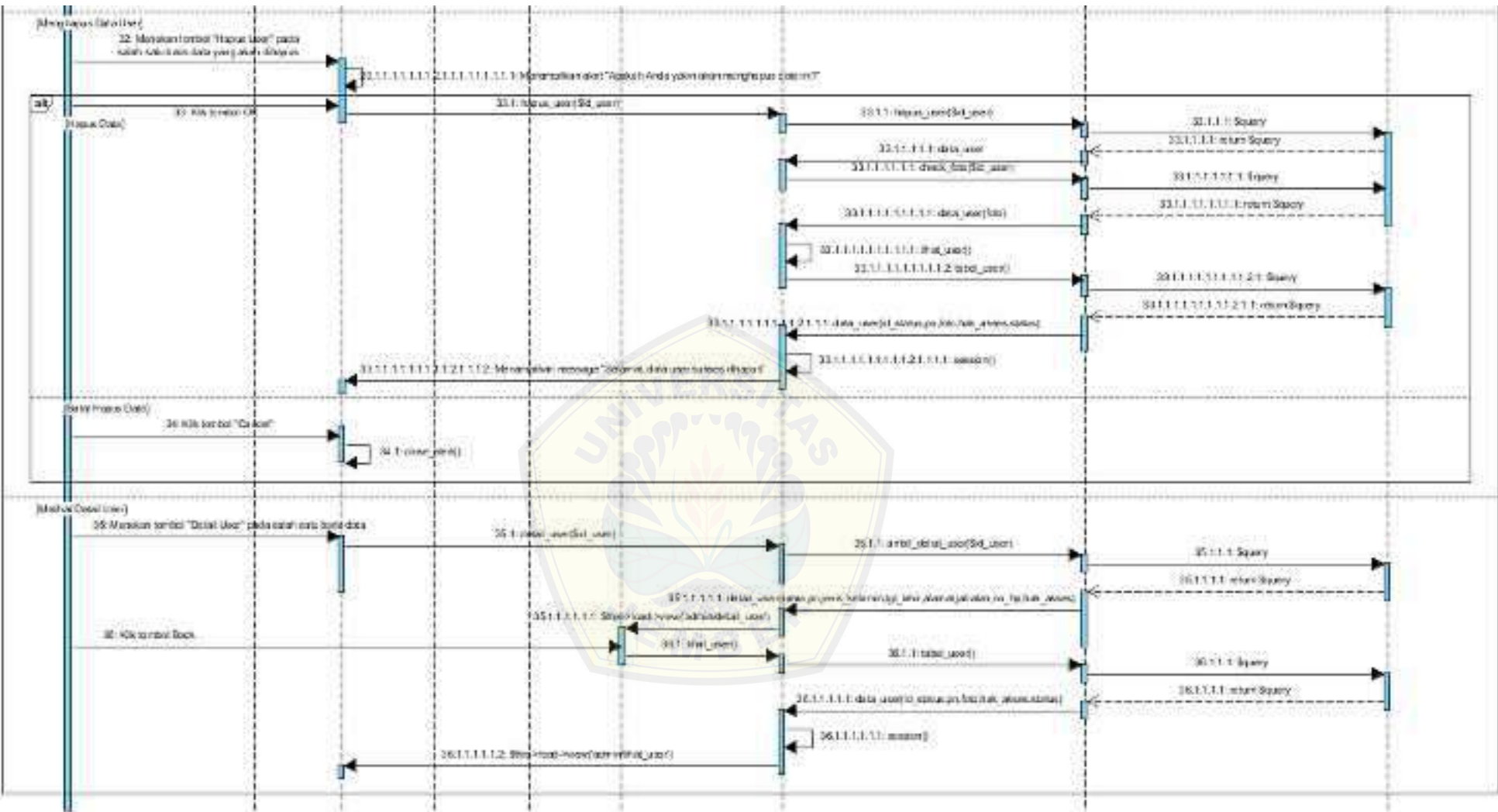


C.5 Sequence Diagram Melihat Informasi Peta Nasabah

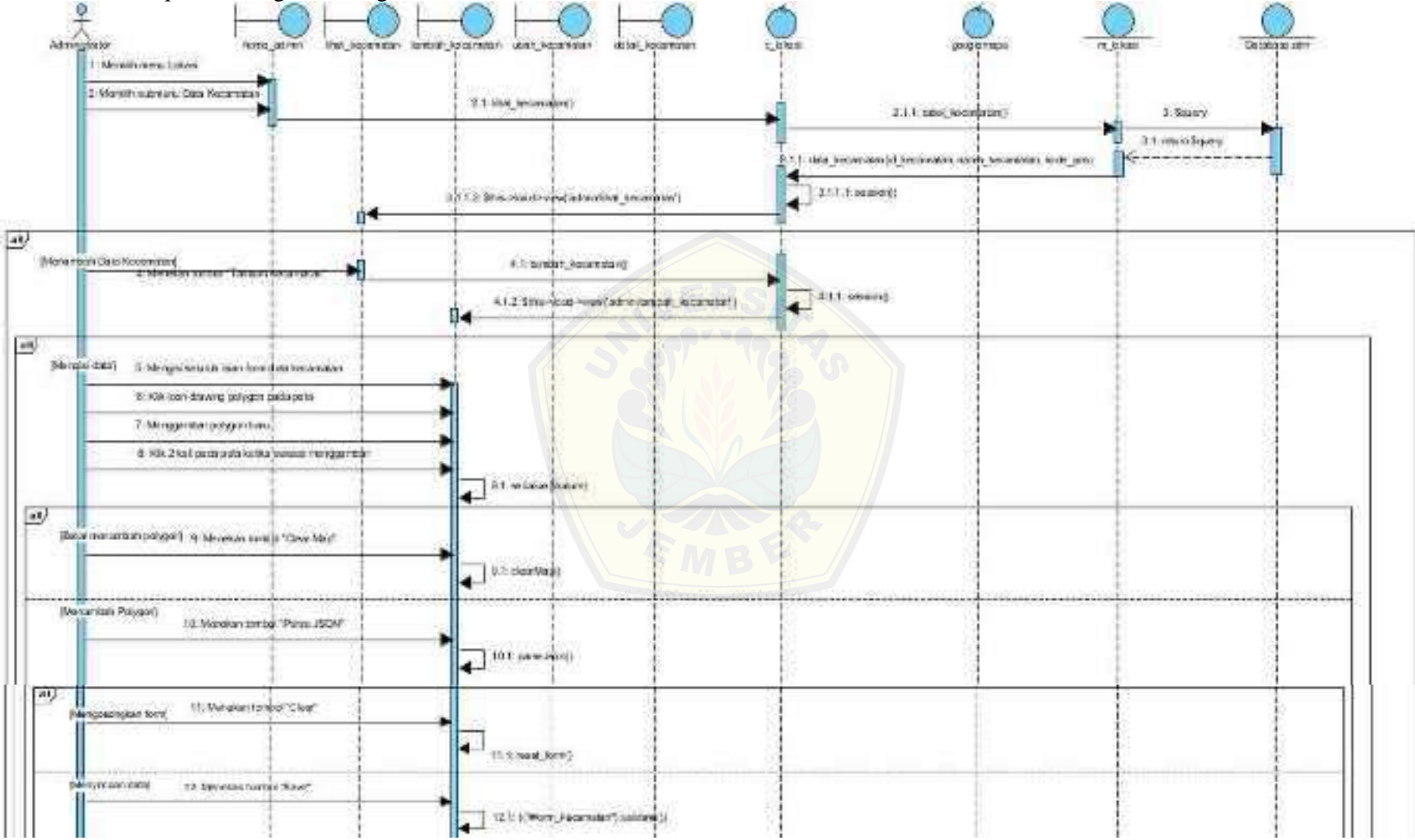


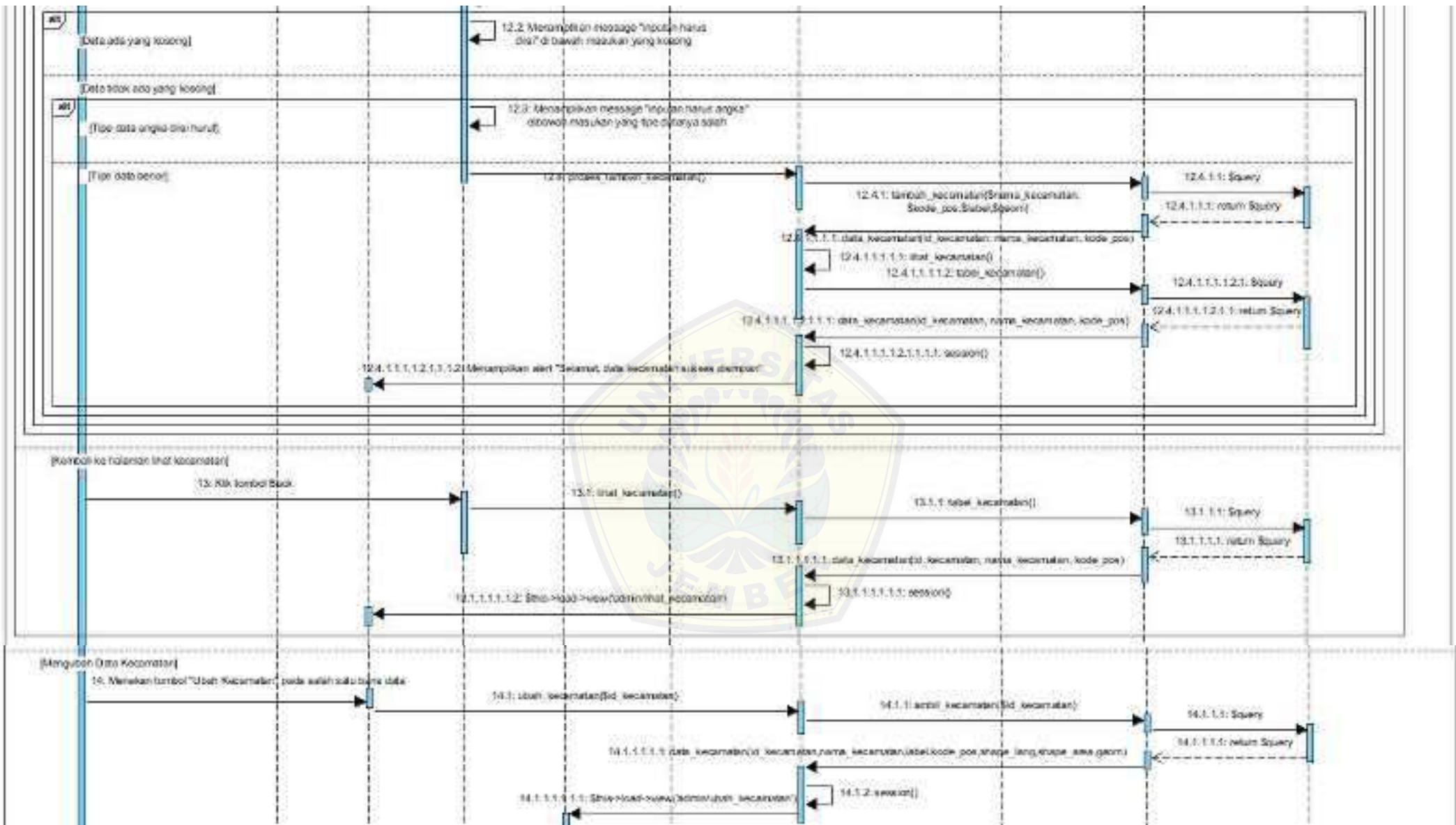


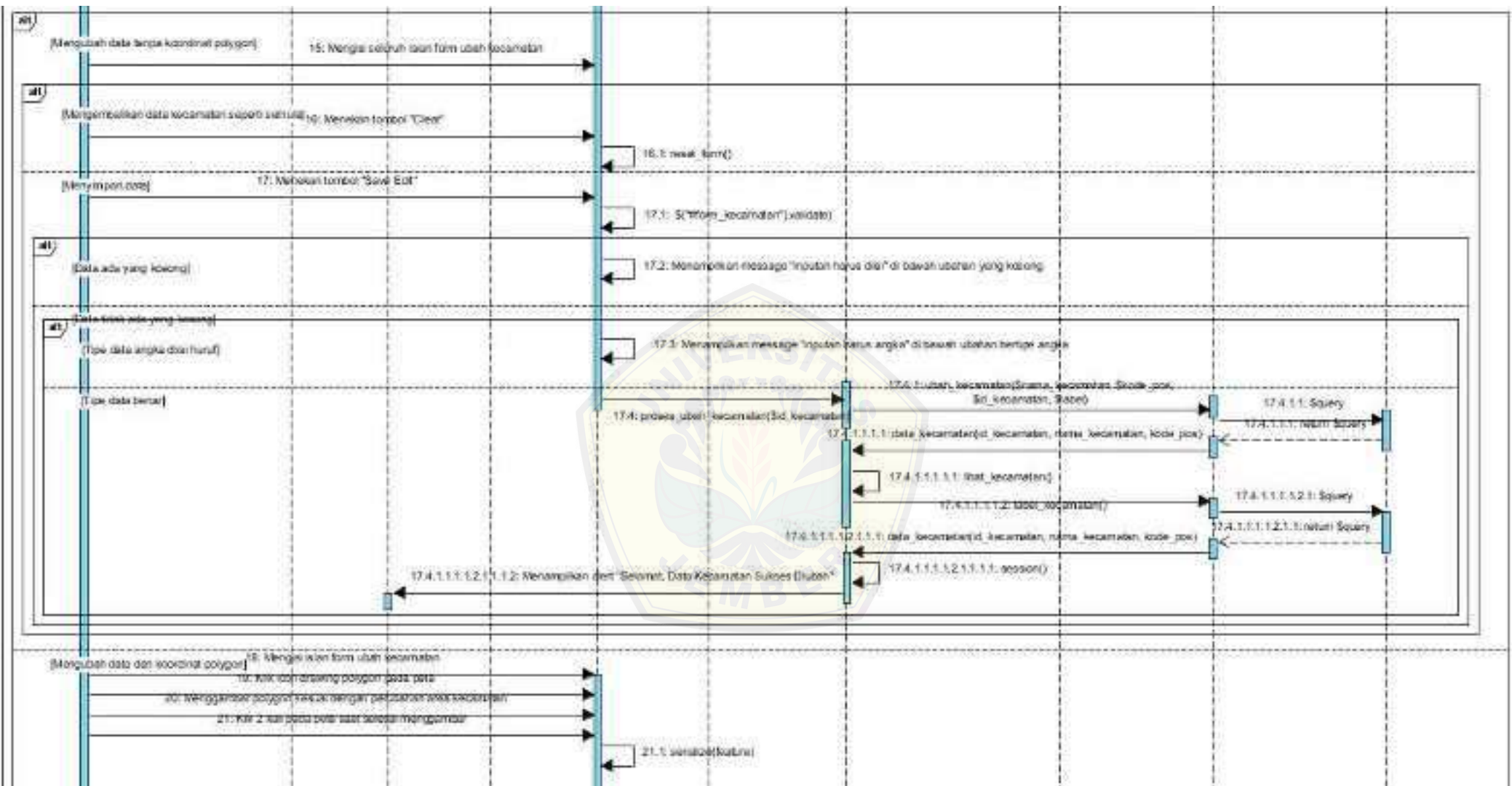


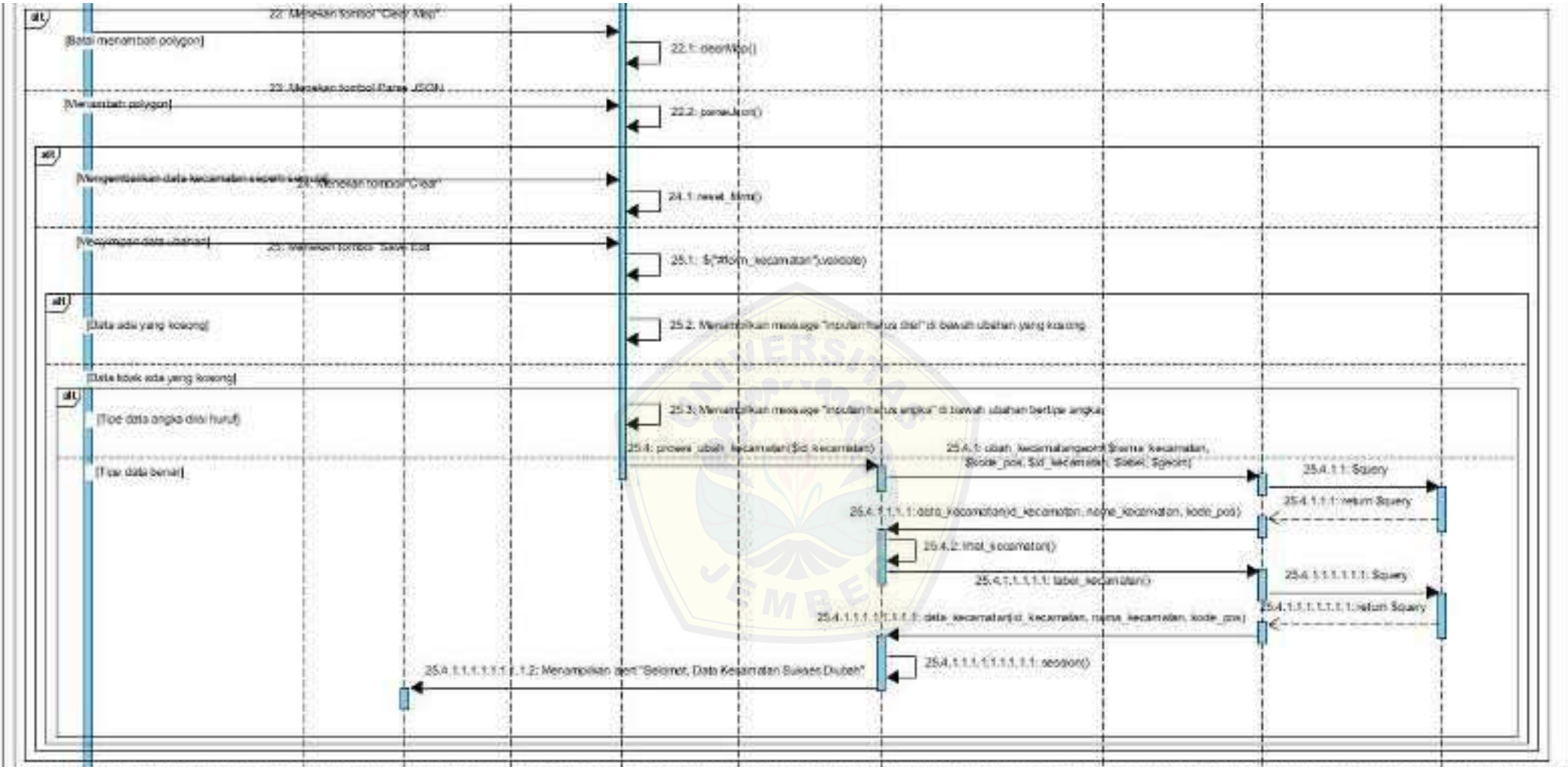


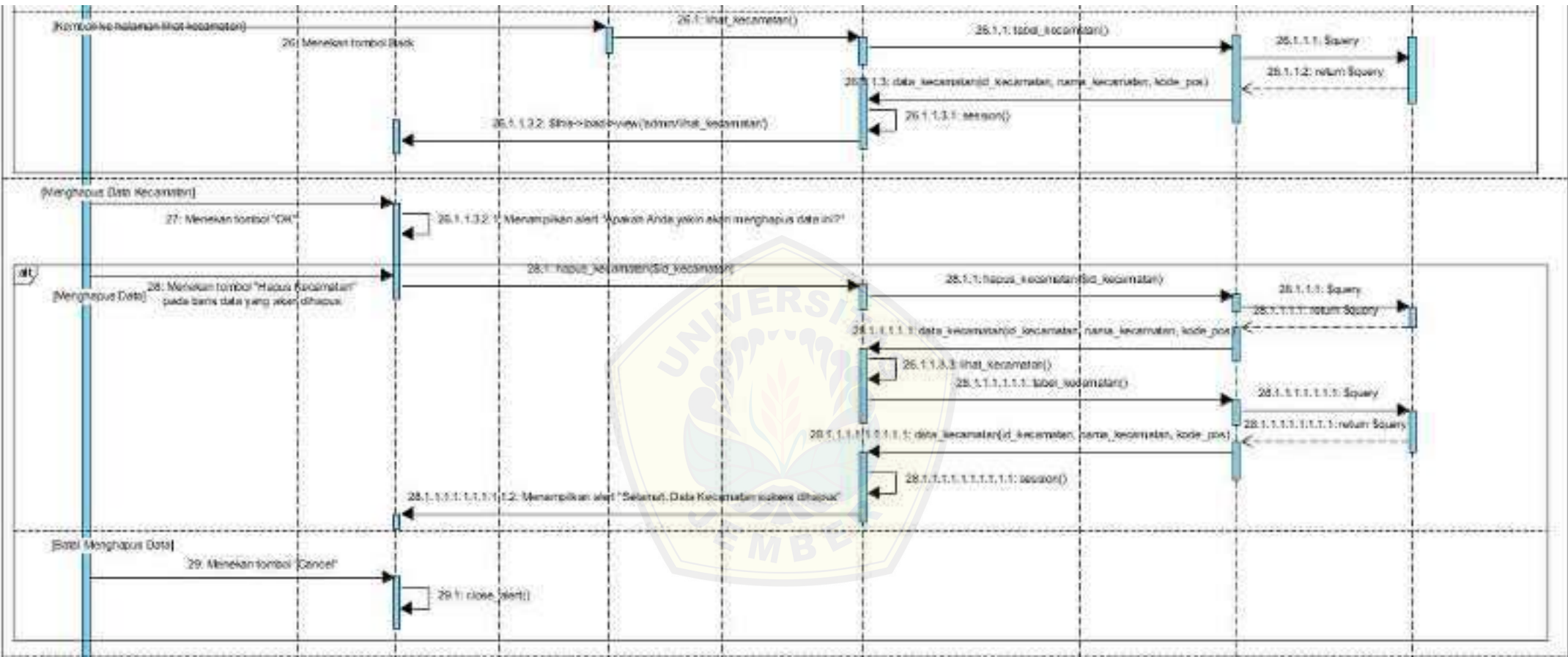
C.7 Sequence Diagram Mengelola Data Kecamatan

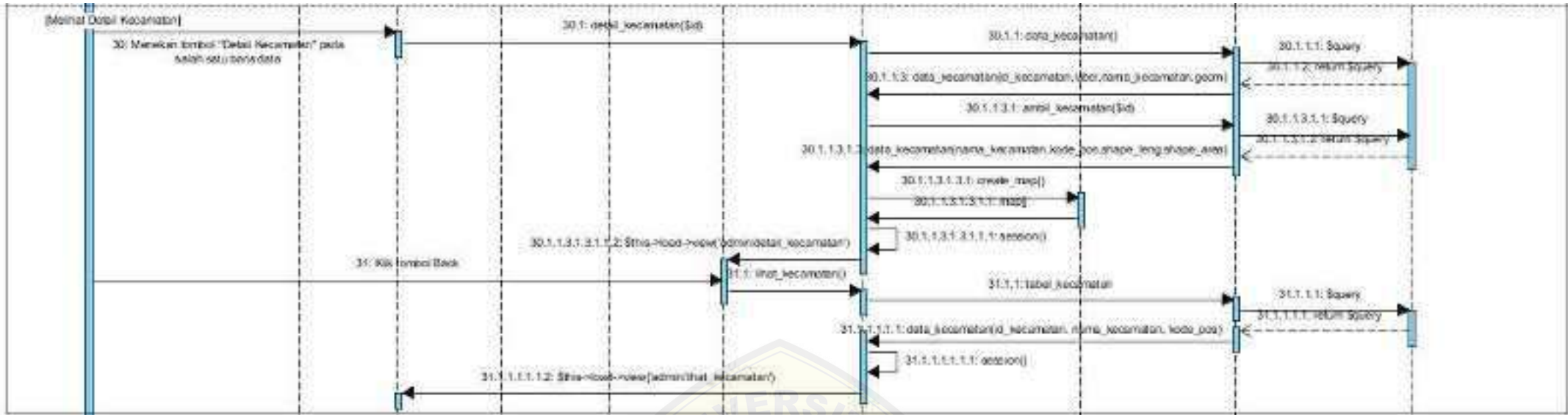




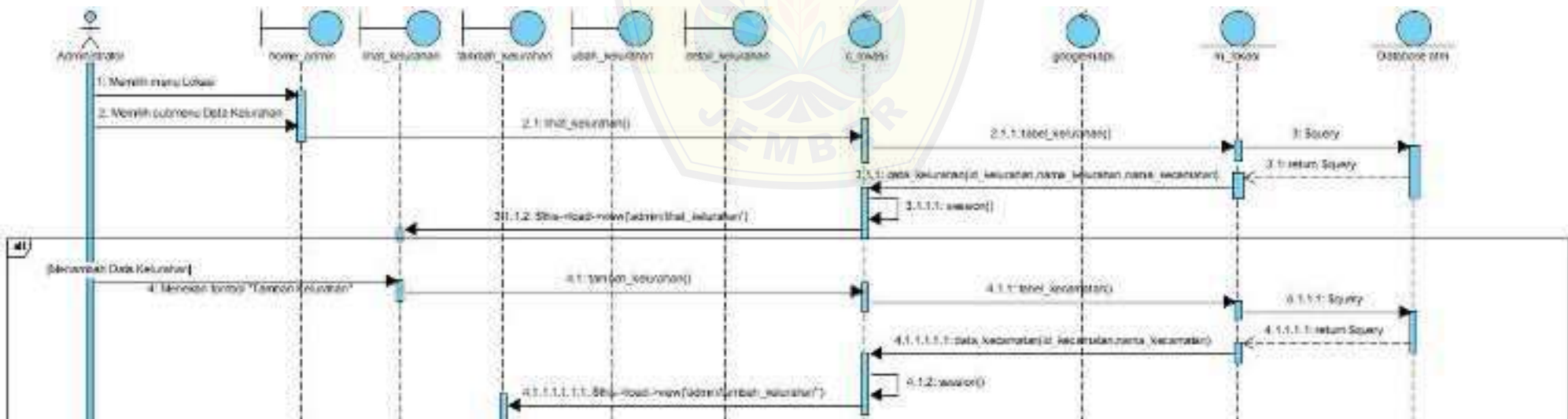


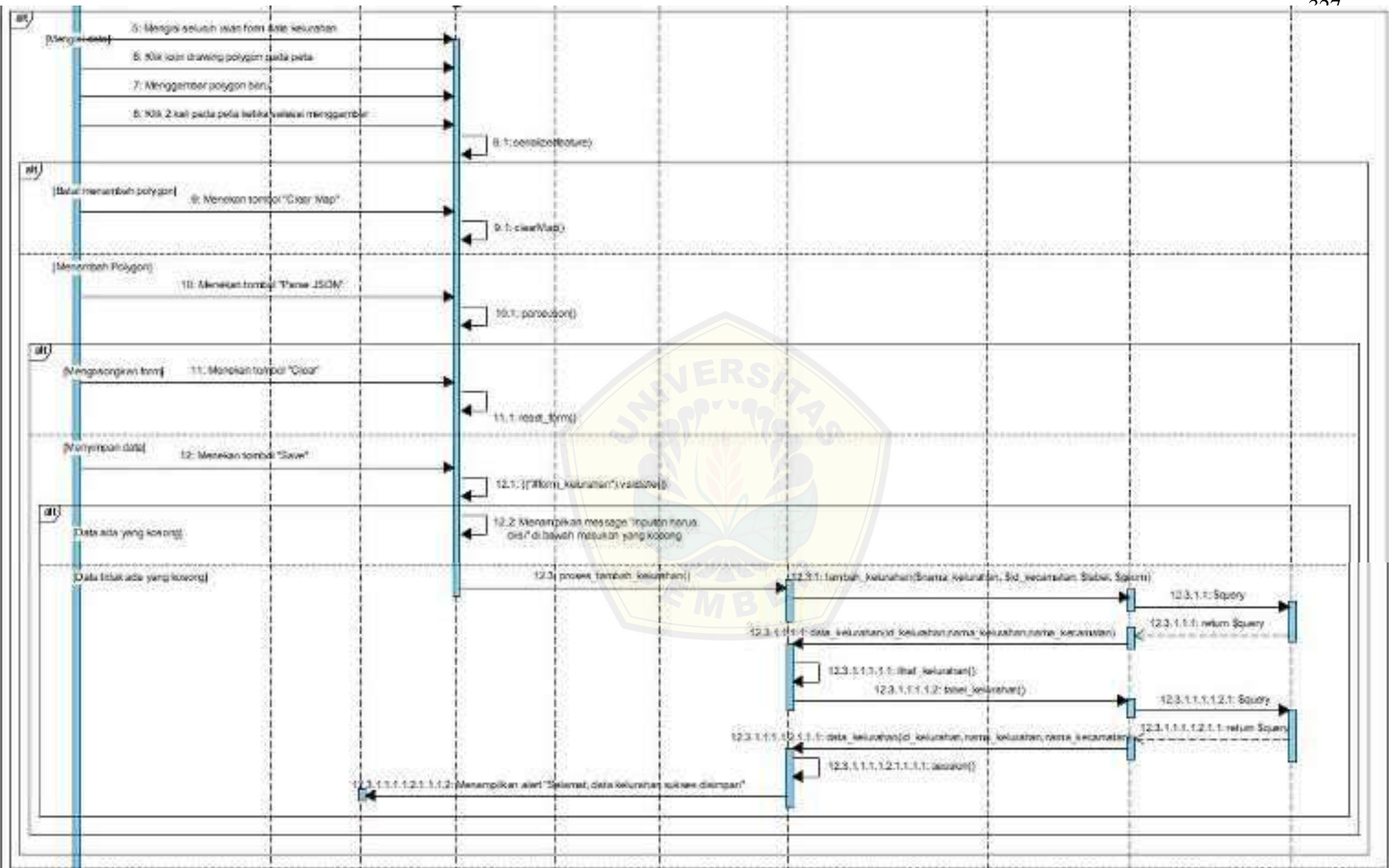


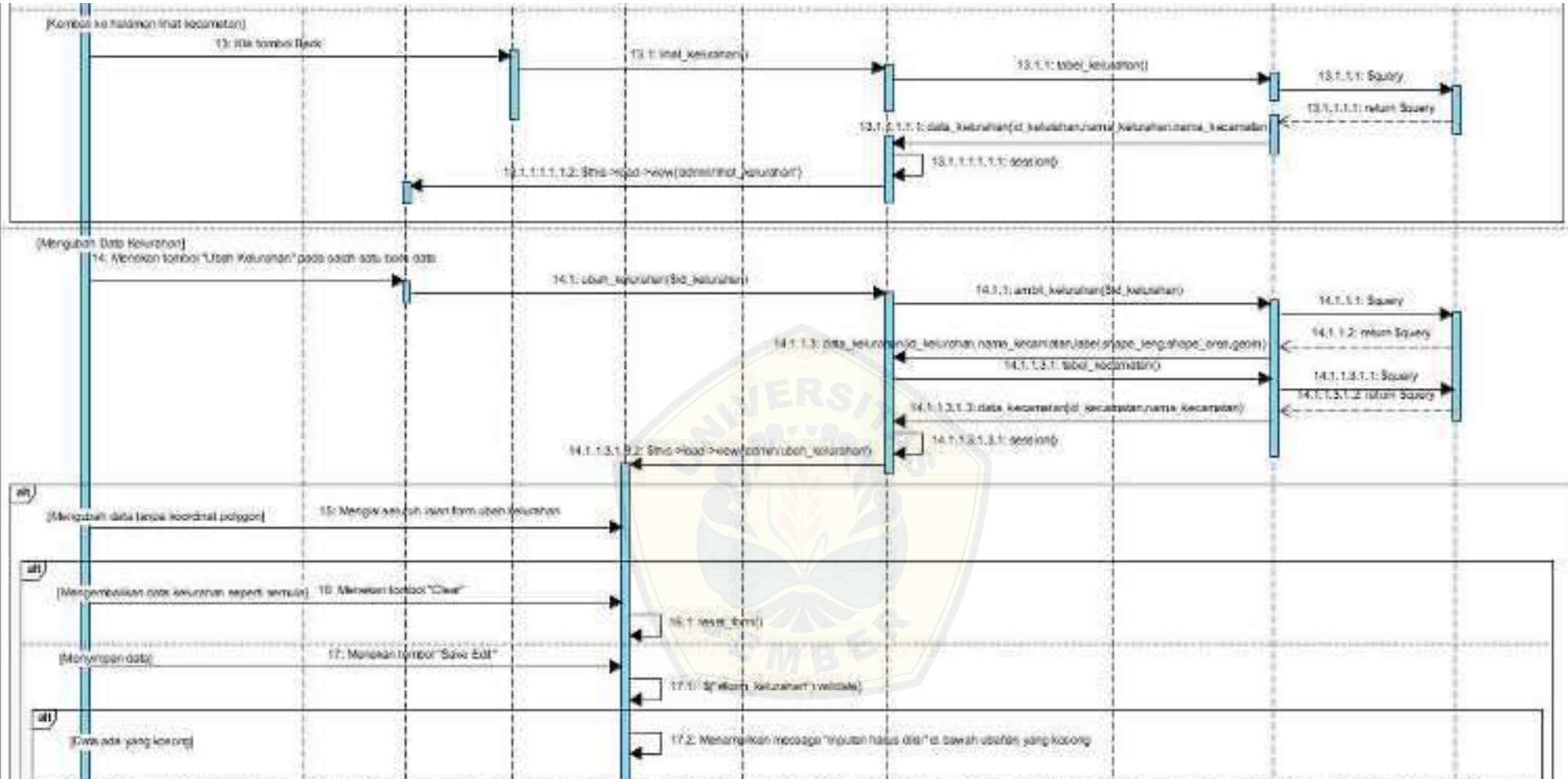


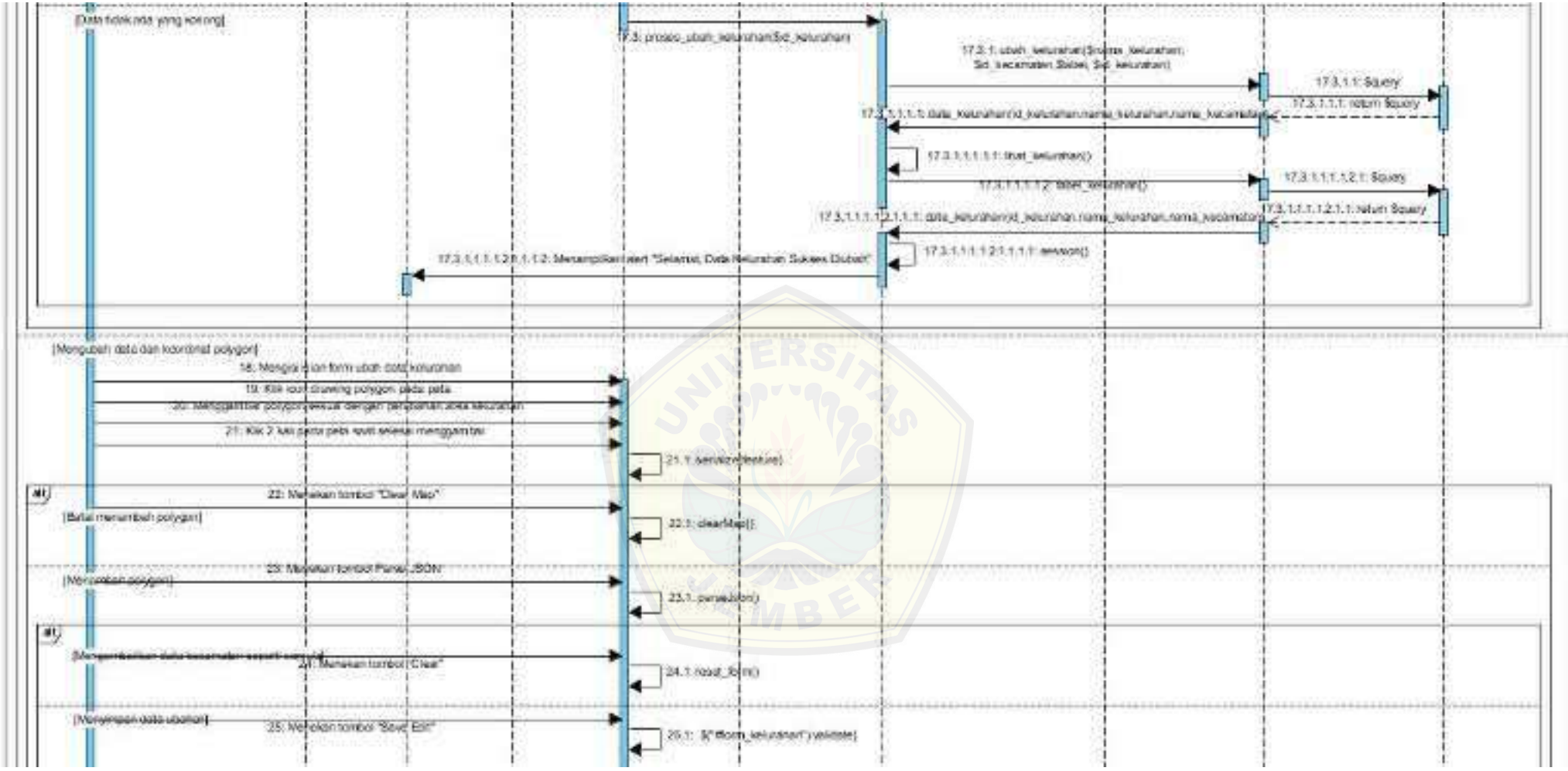


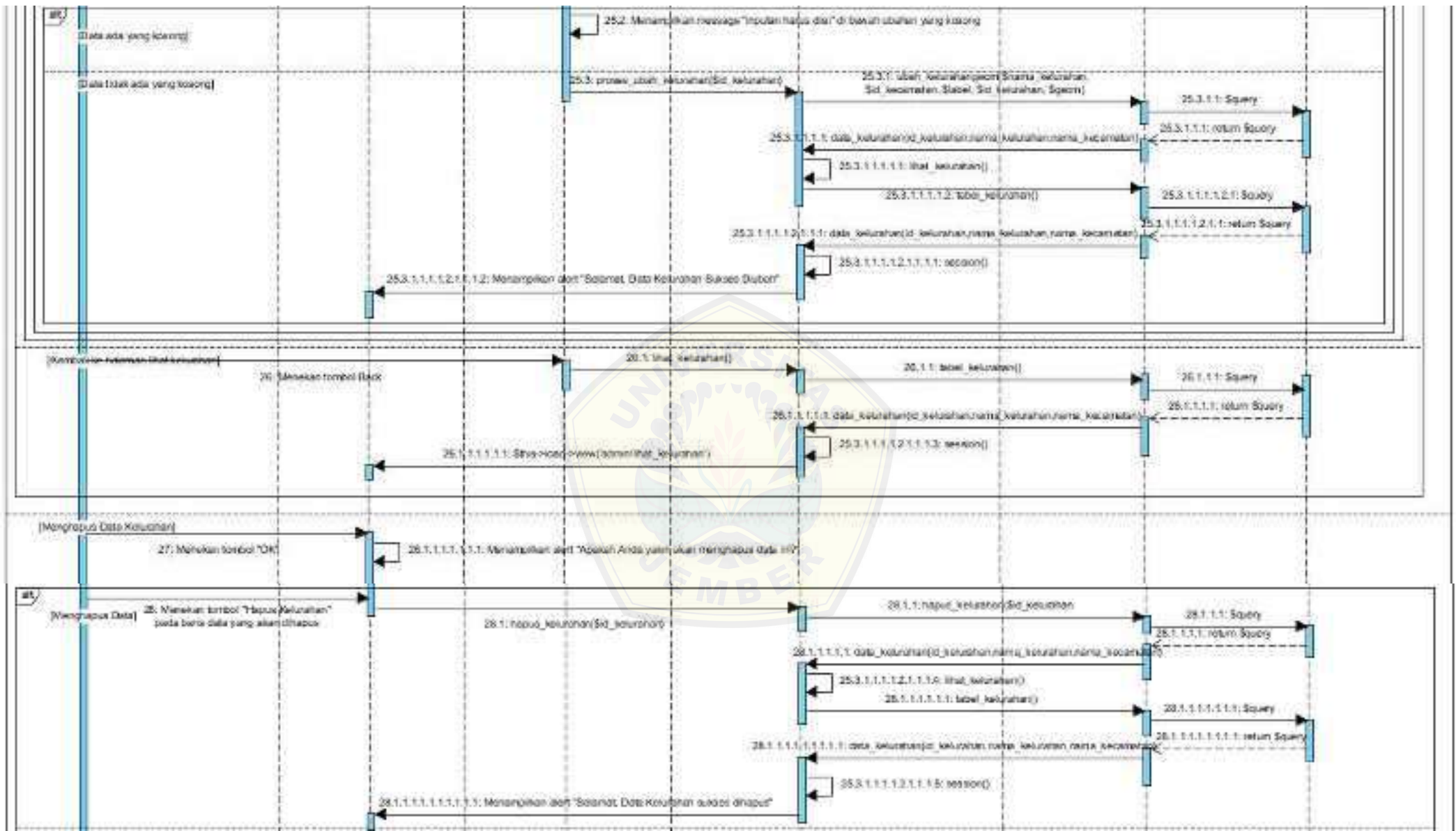
C.8 Sequence Diagram Mengelola Data Kelurahan

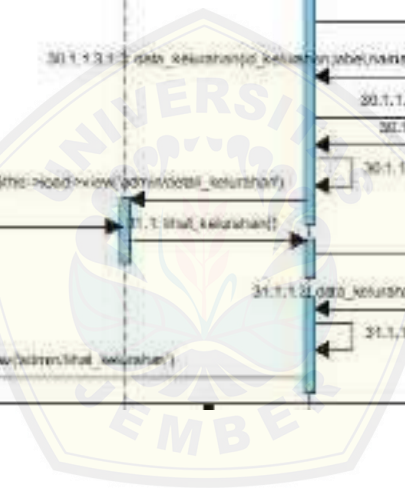
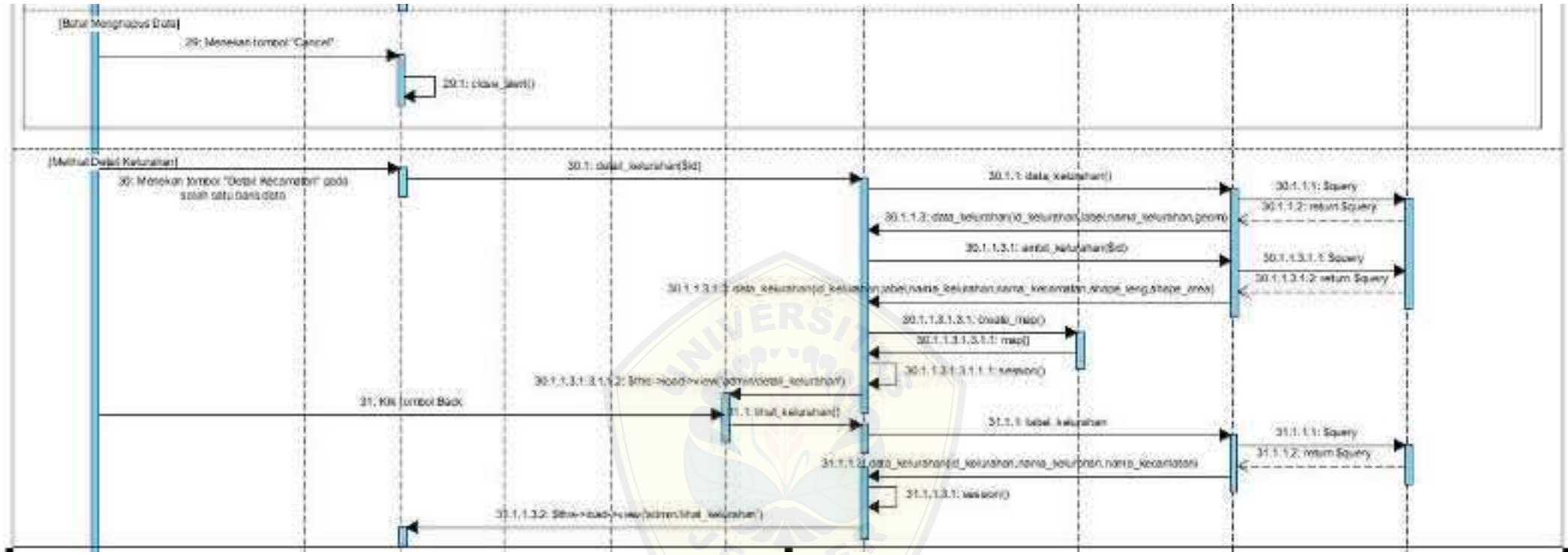




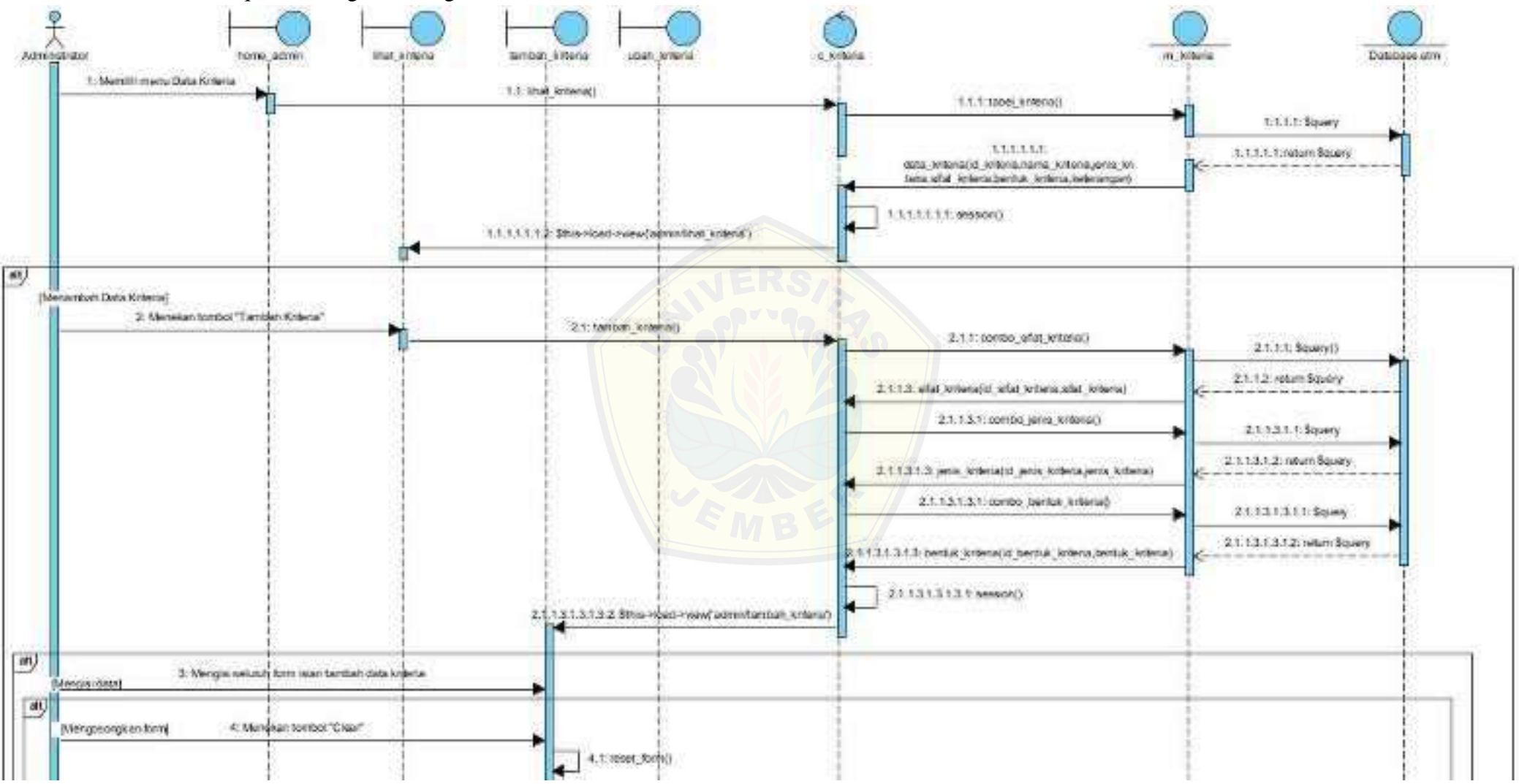


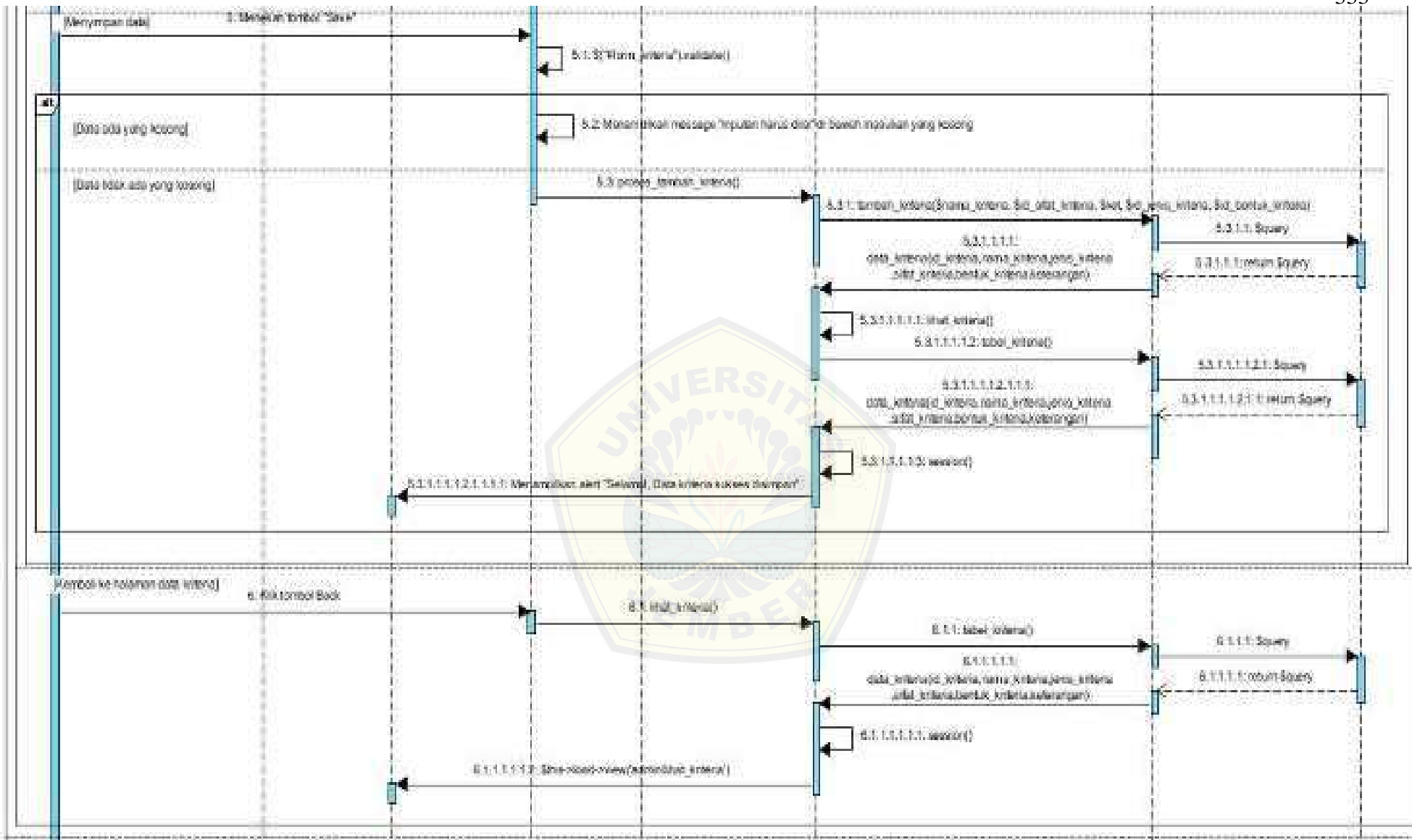


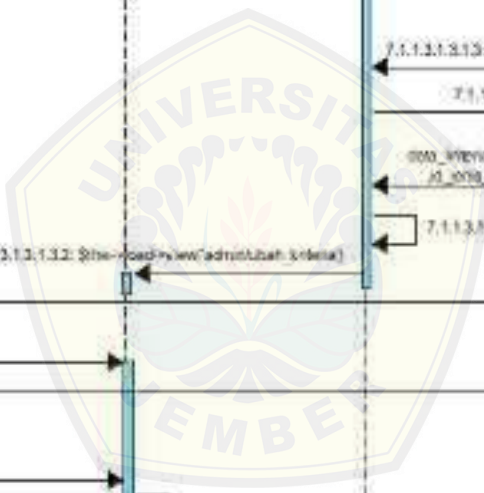
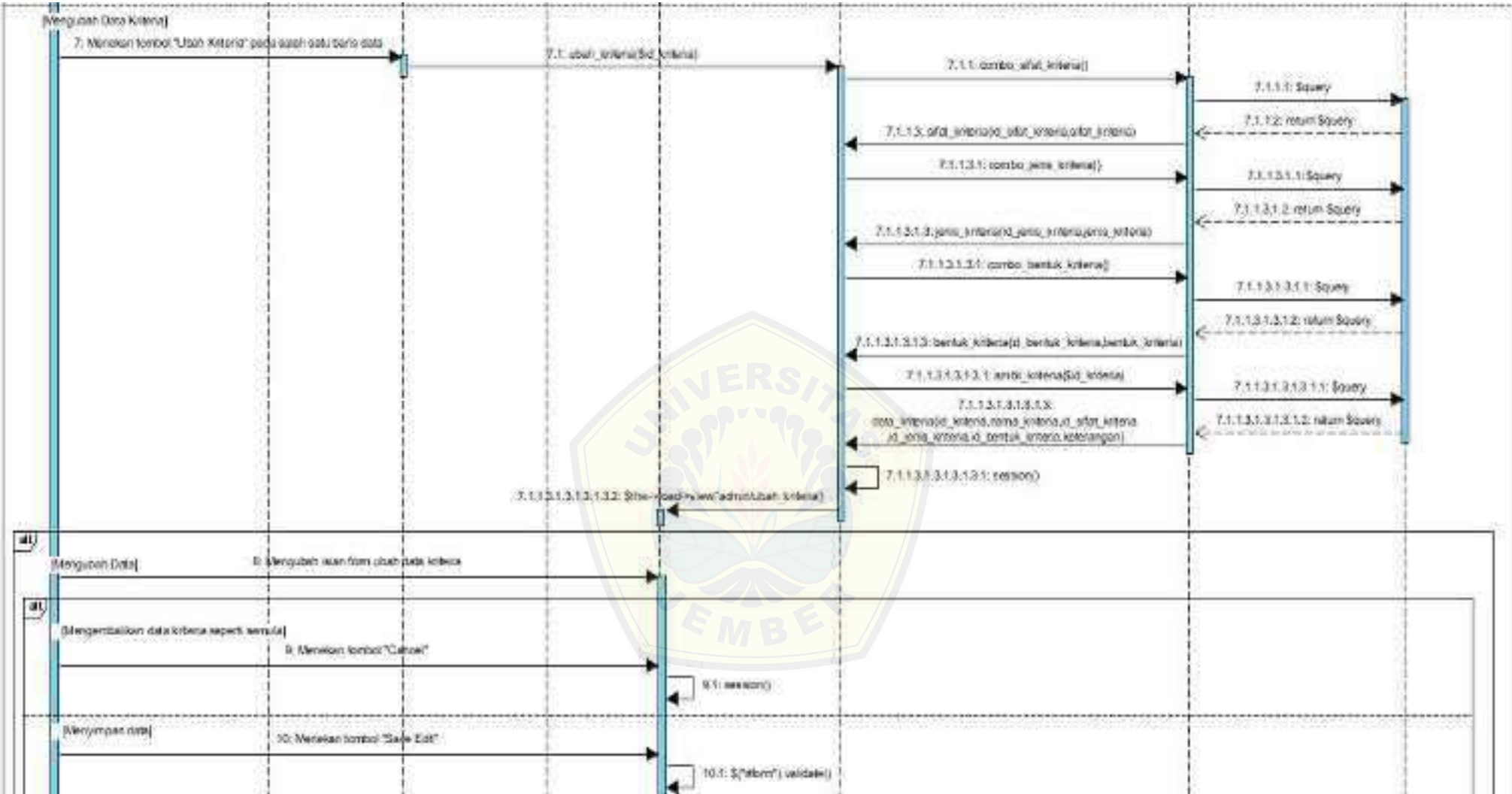


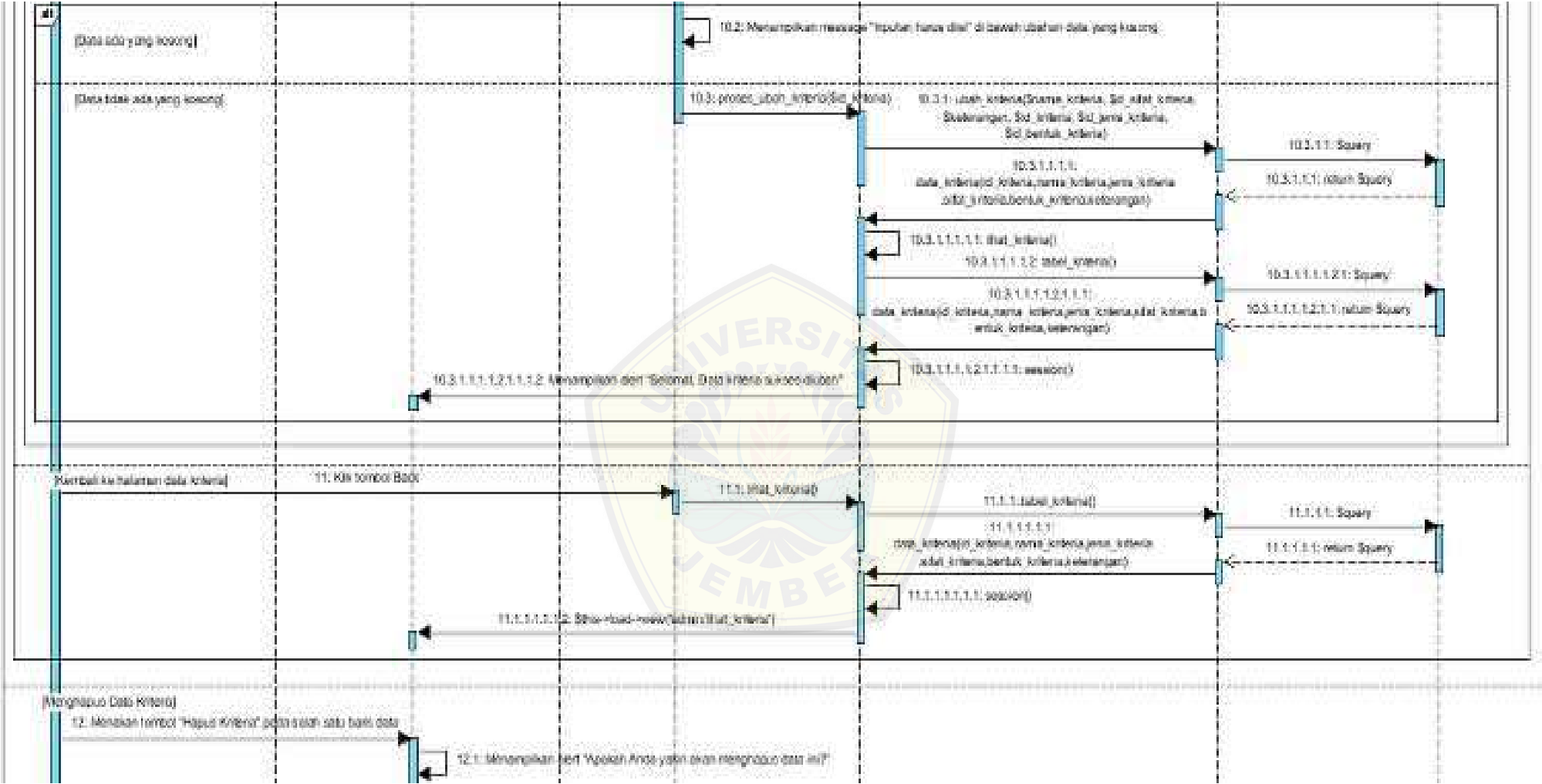


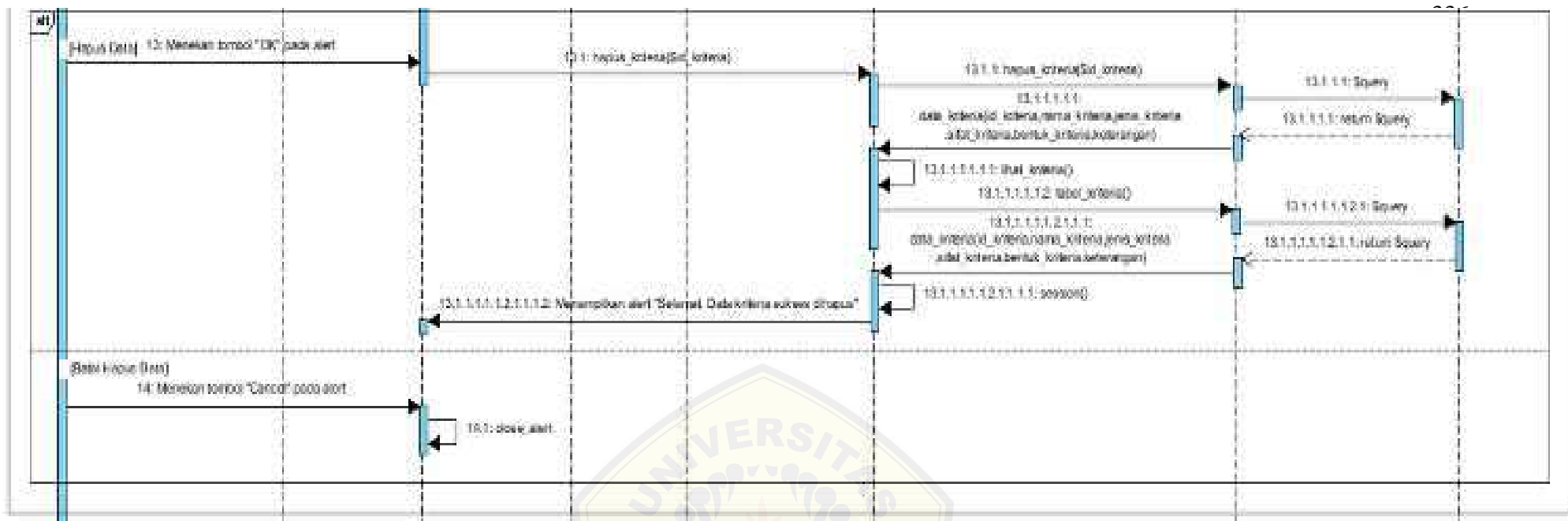
C.9 Sequence Diagram Mengelola Data Kriteria



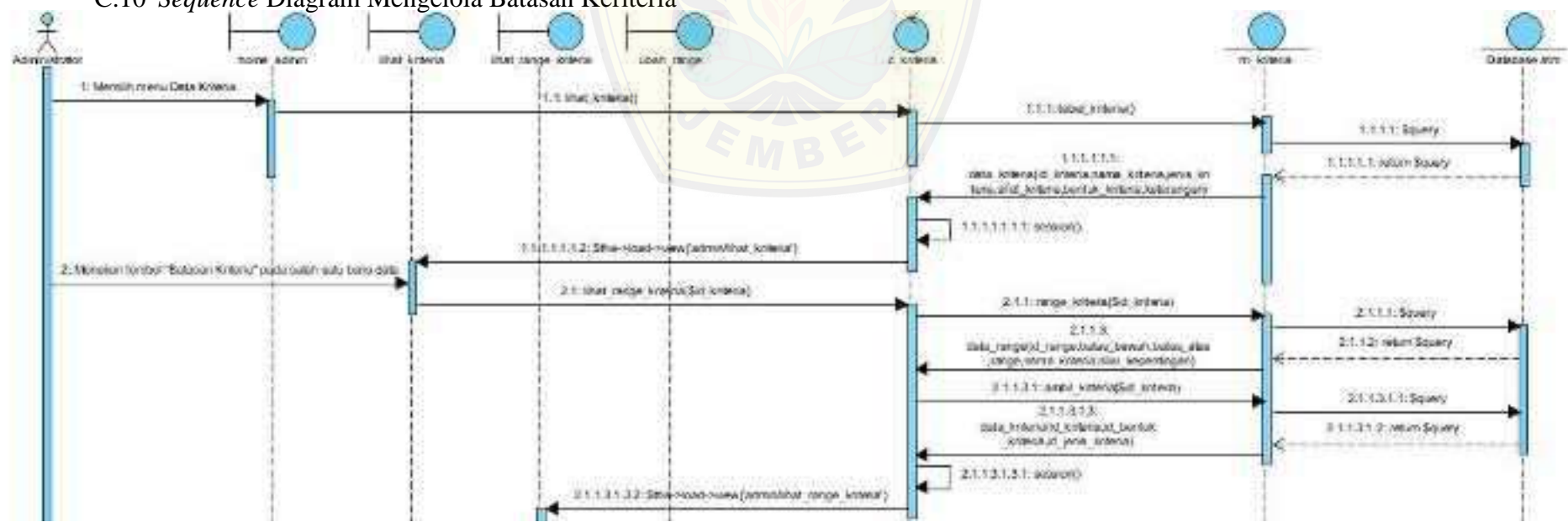


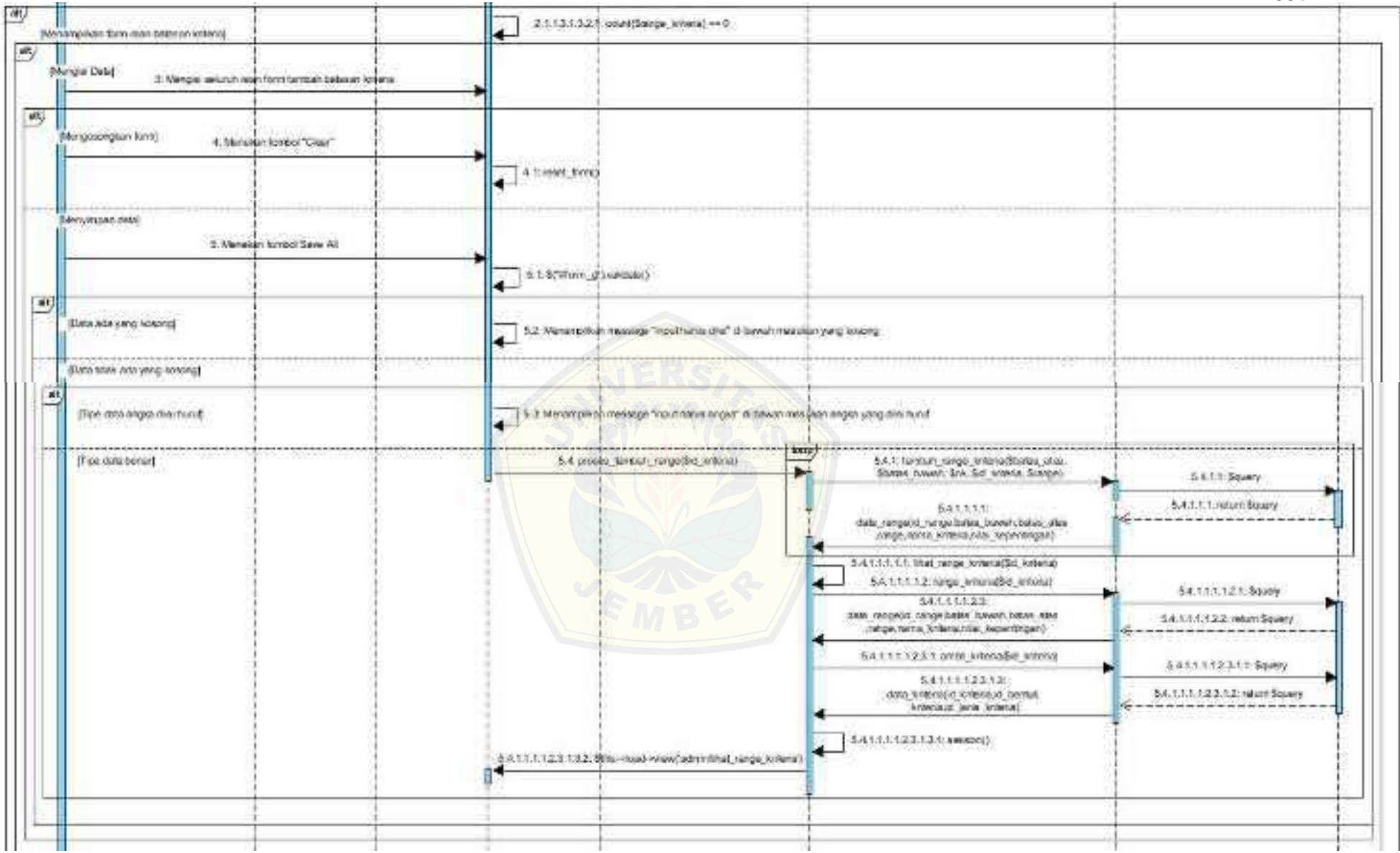


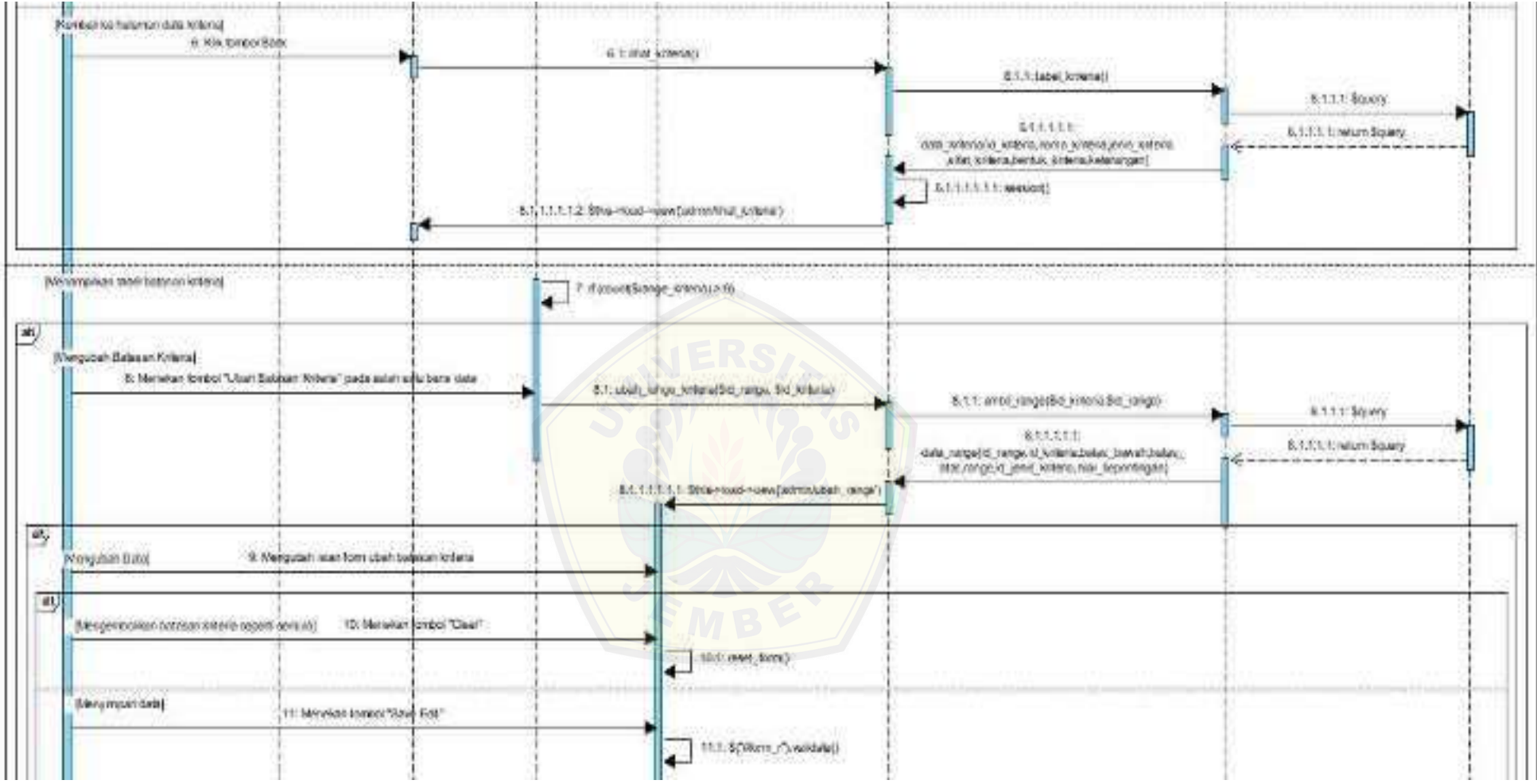


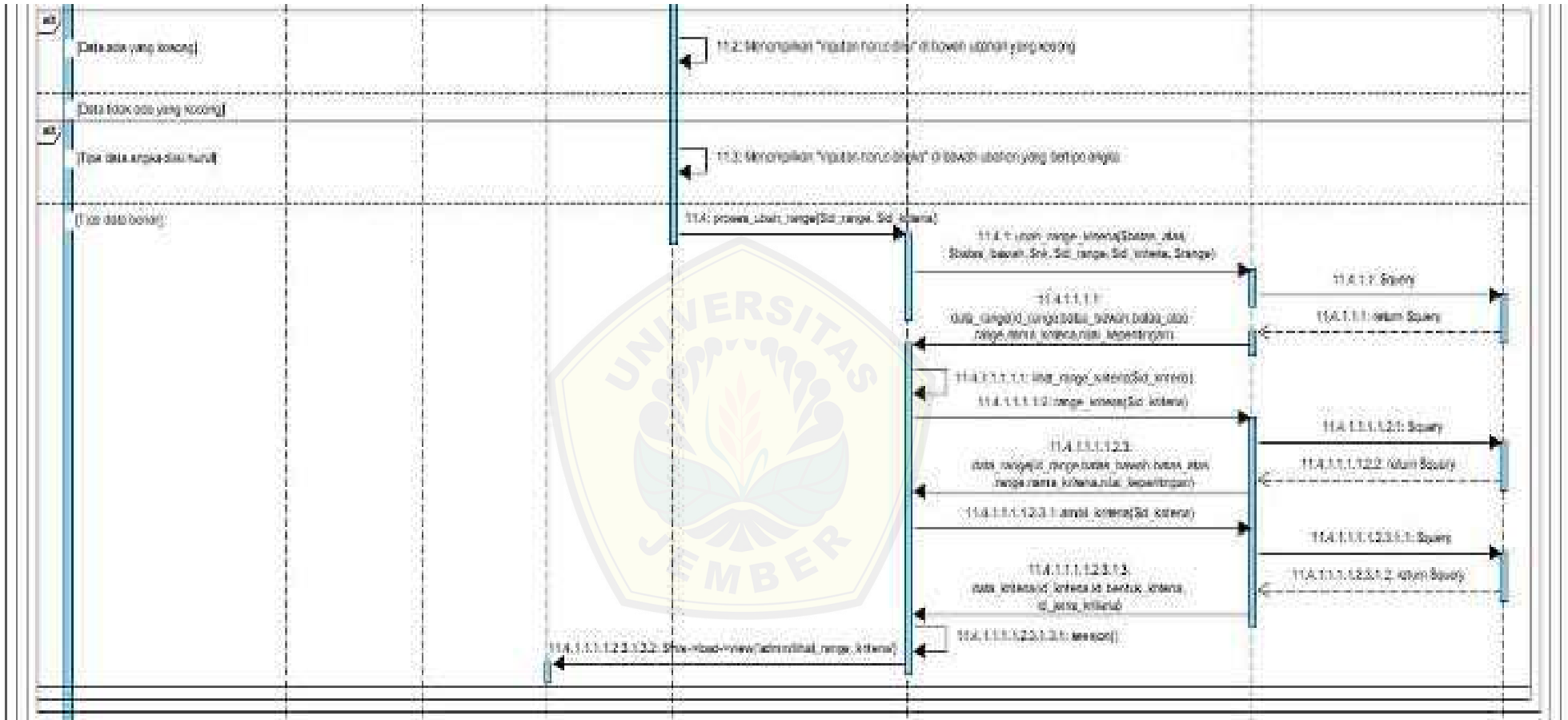


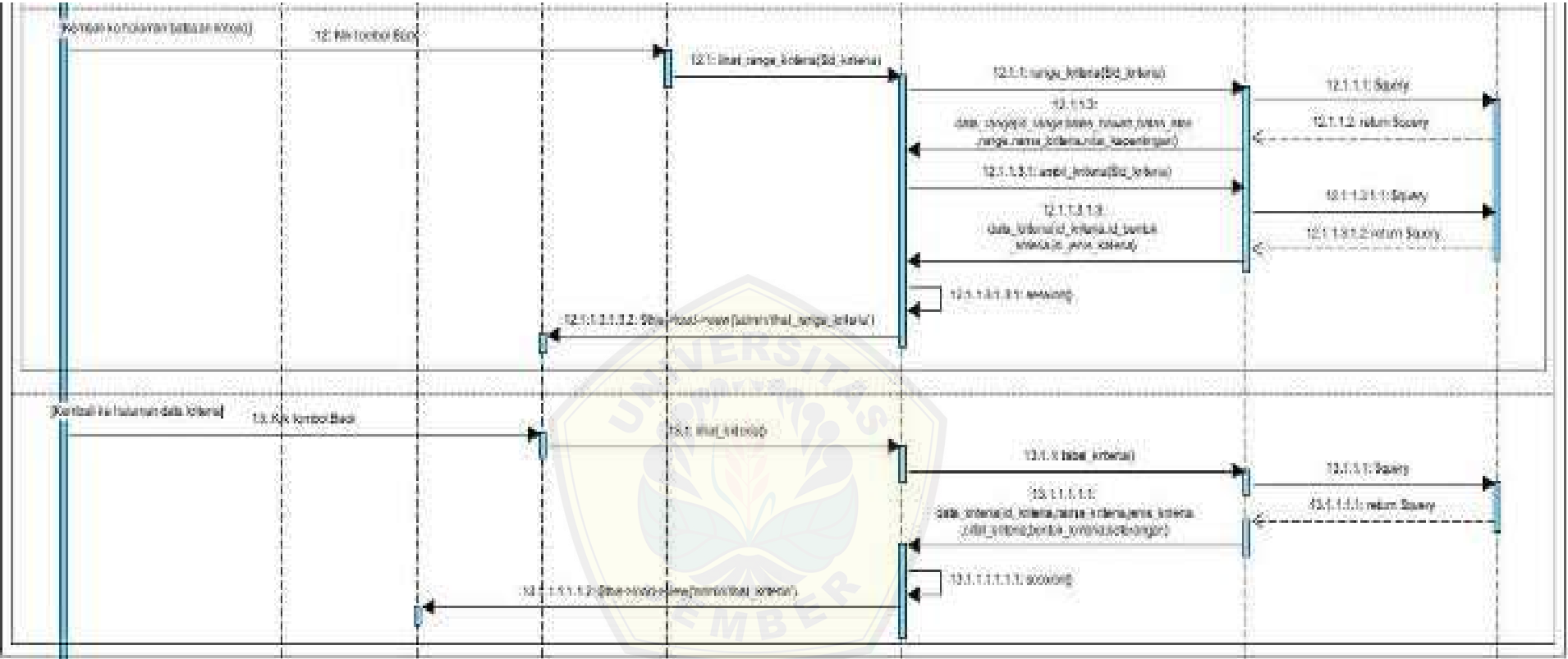
C.10 Sequence Diagram Mengelola Batasan Kriteria

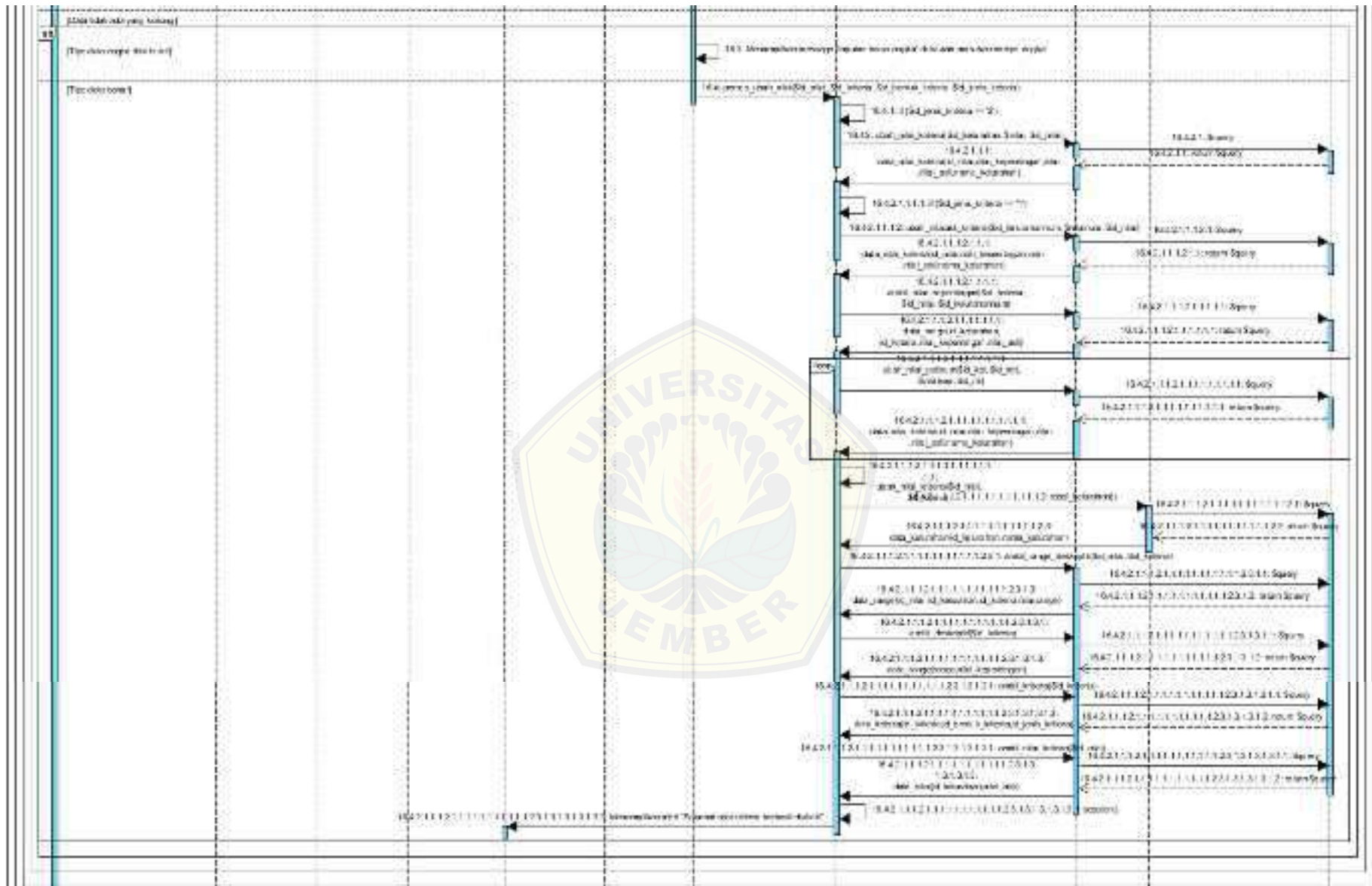


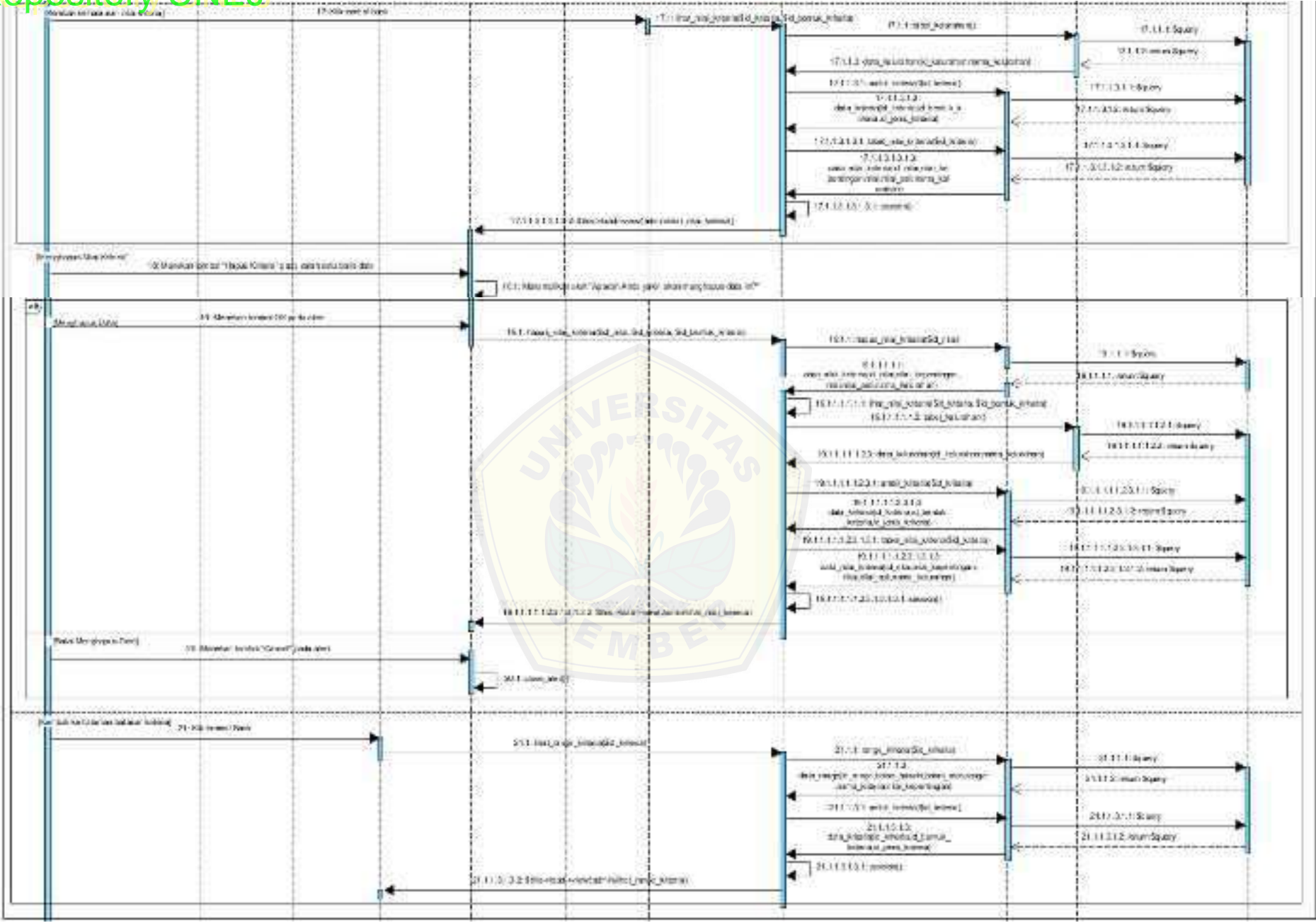


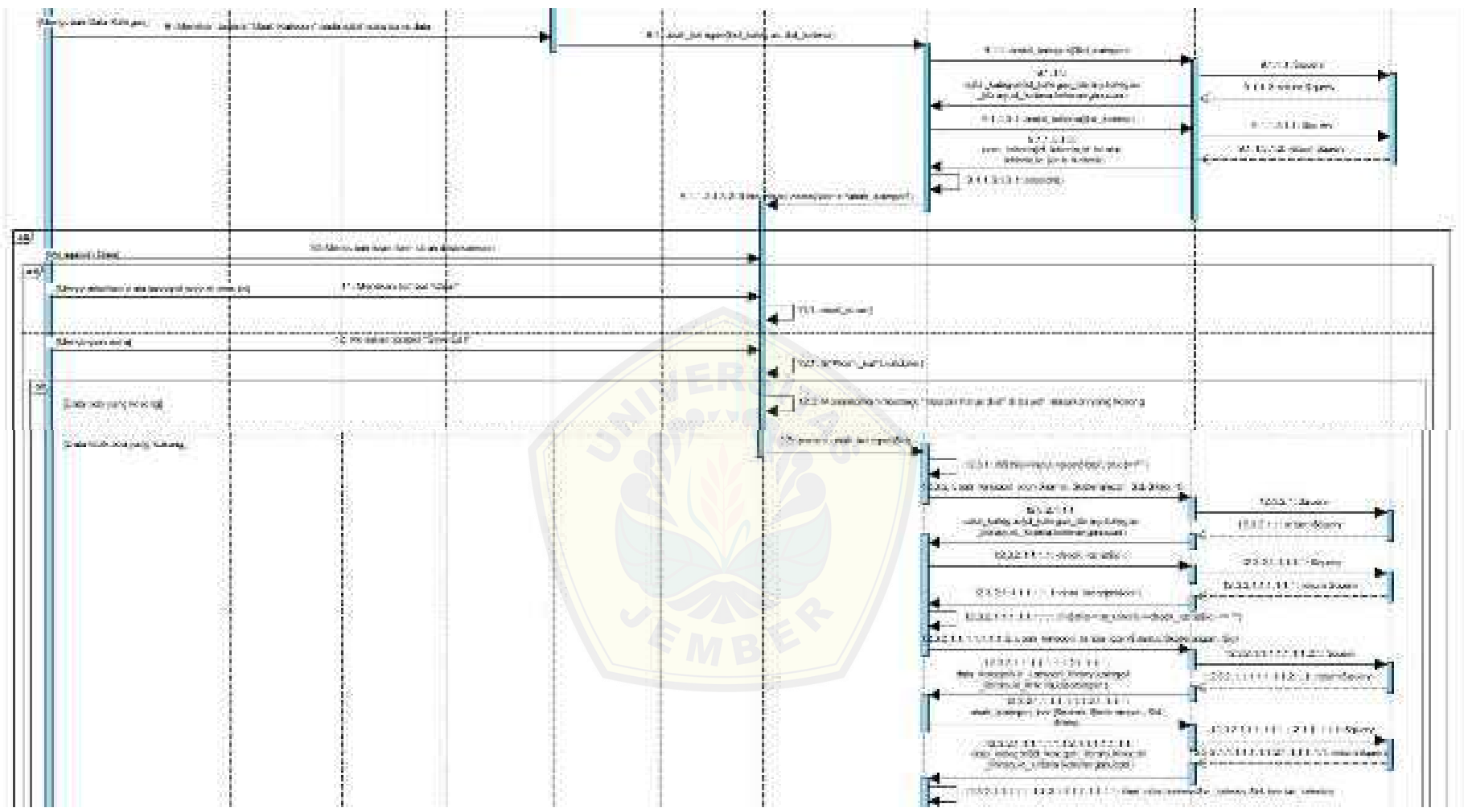


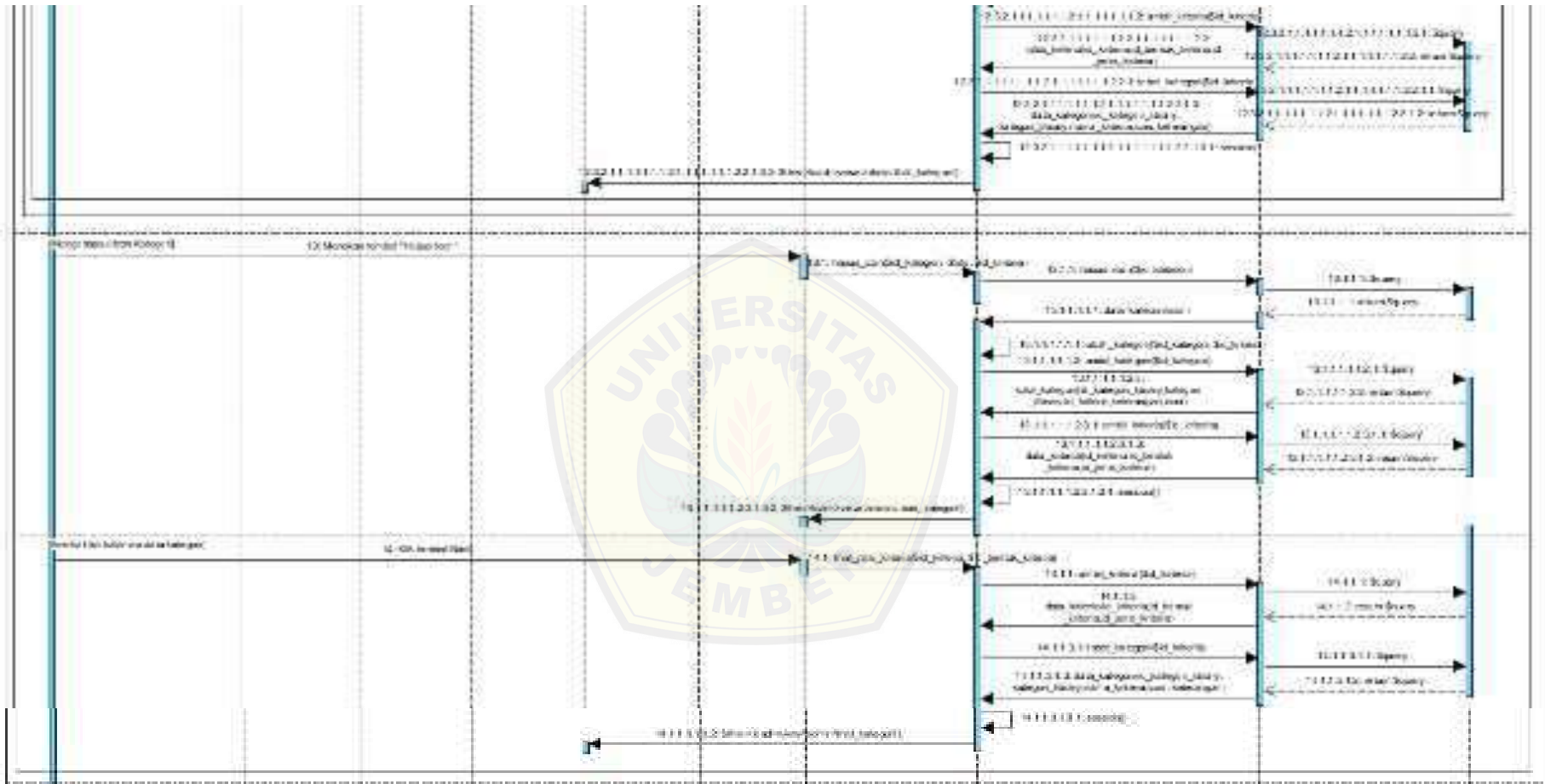


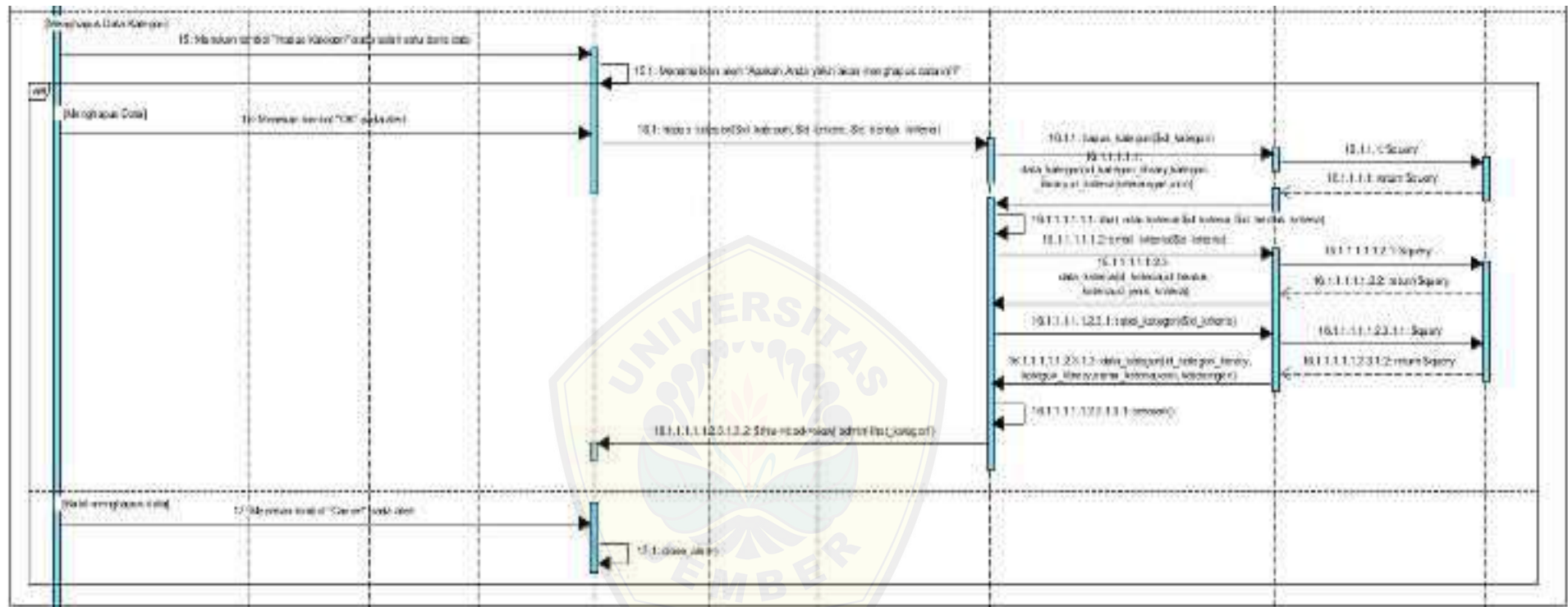


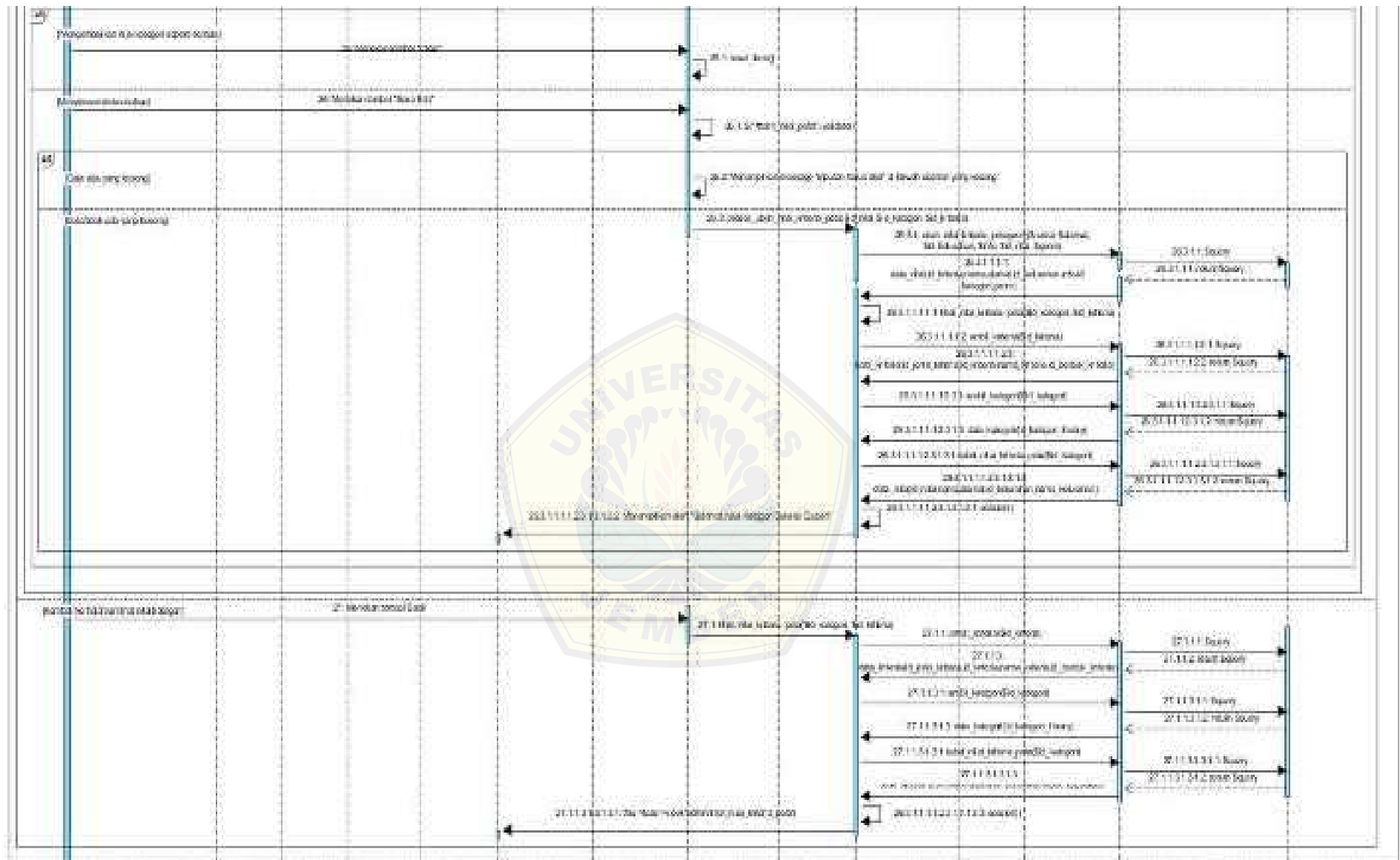


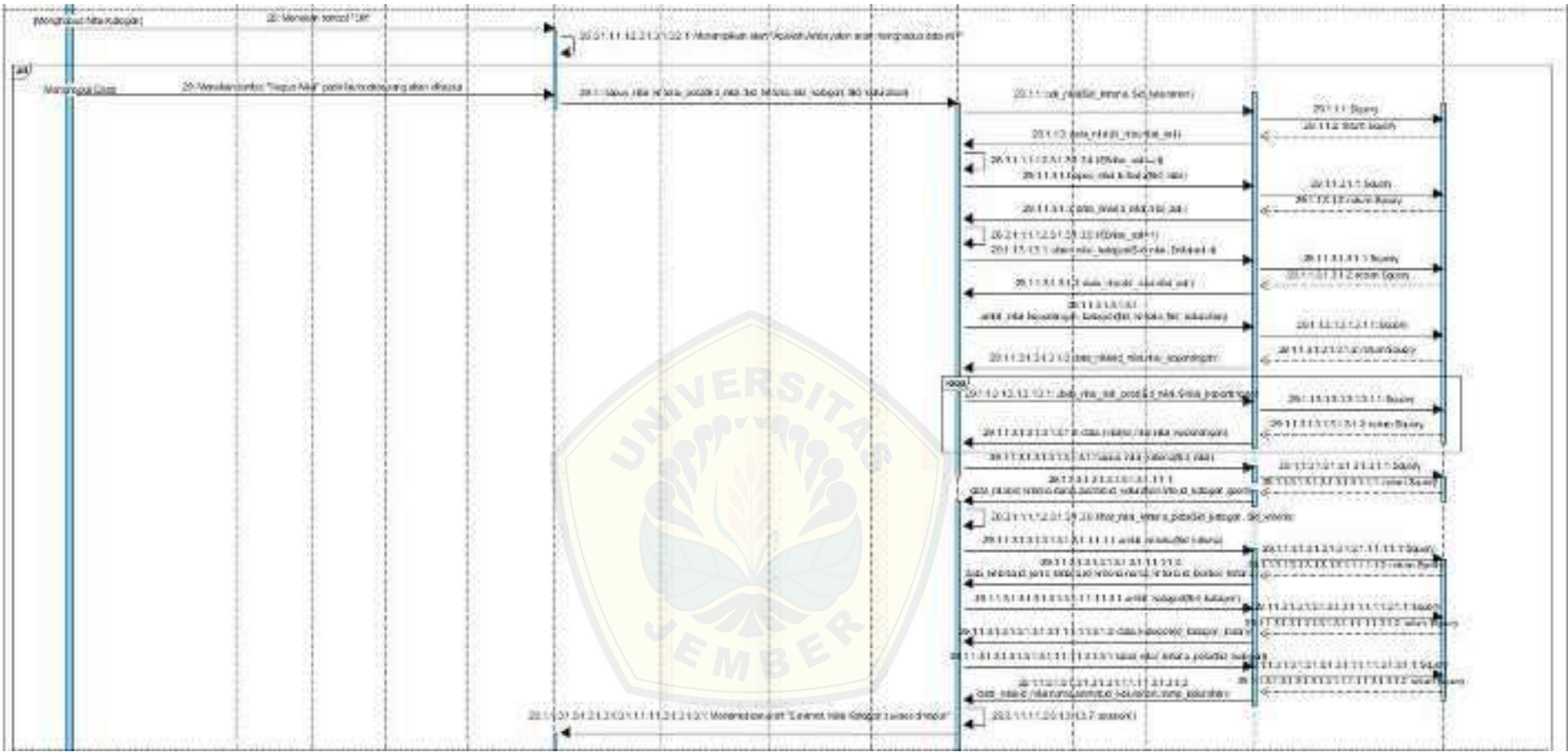




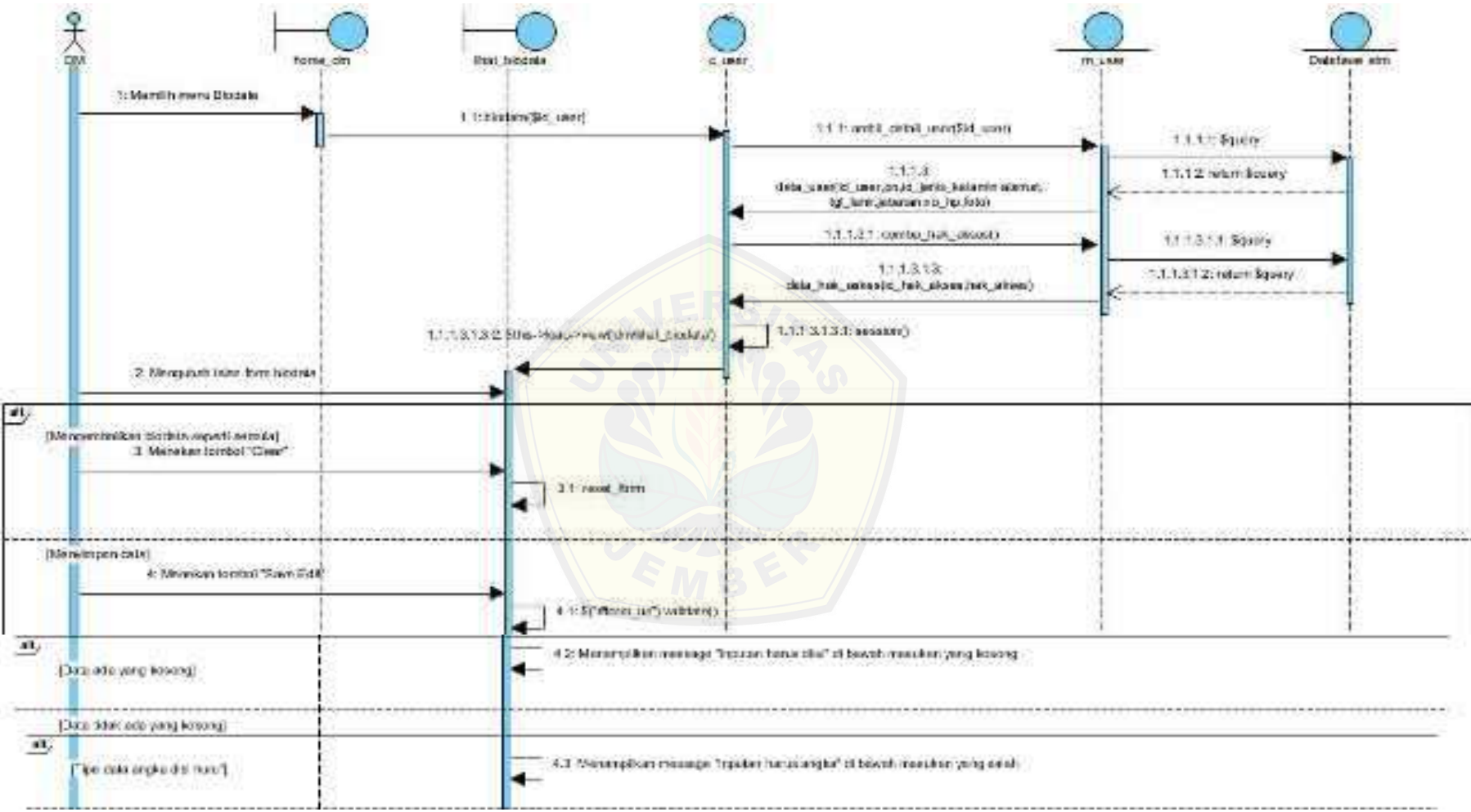


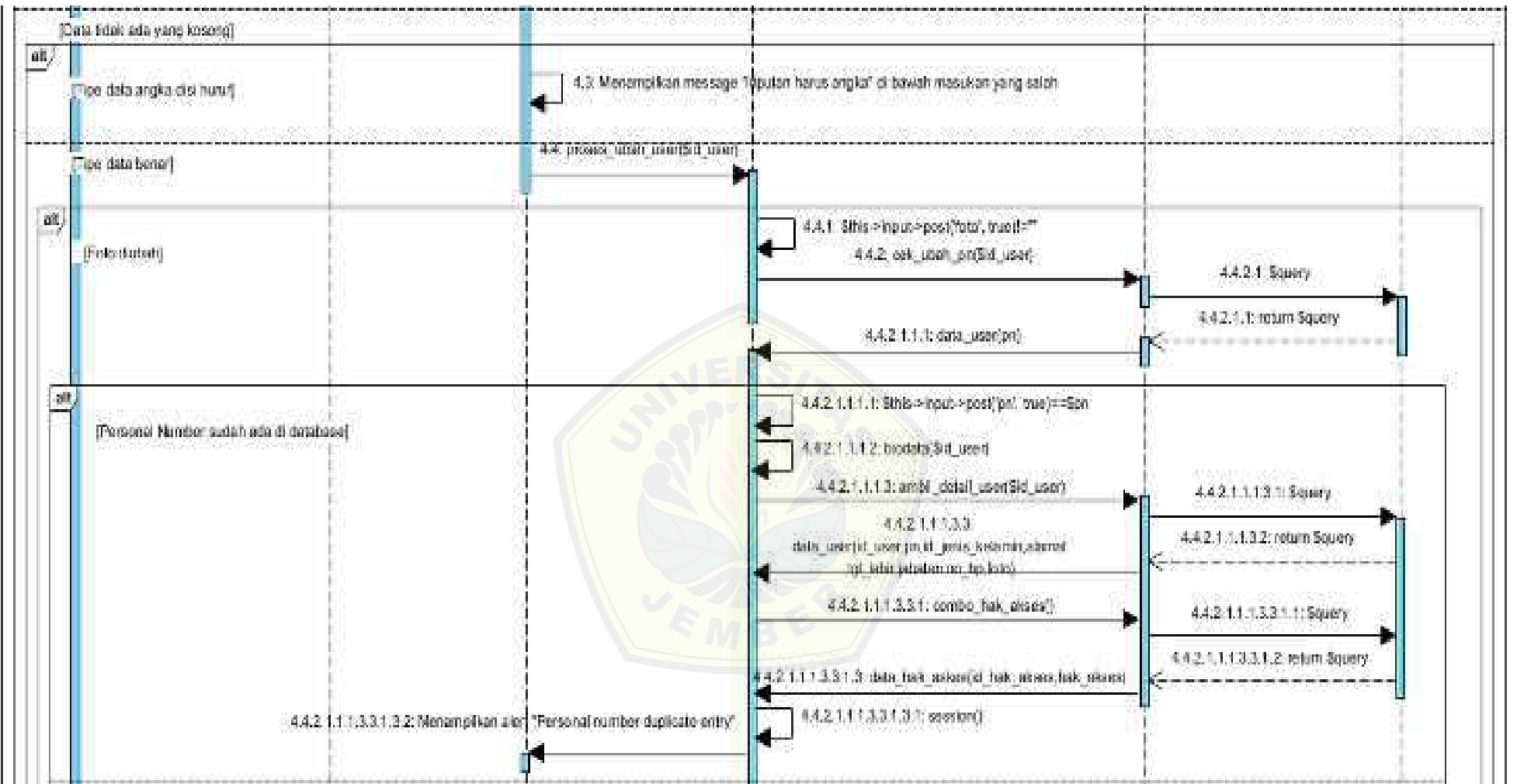


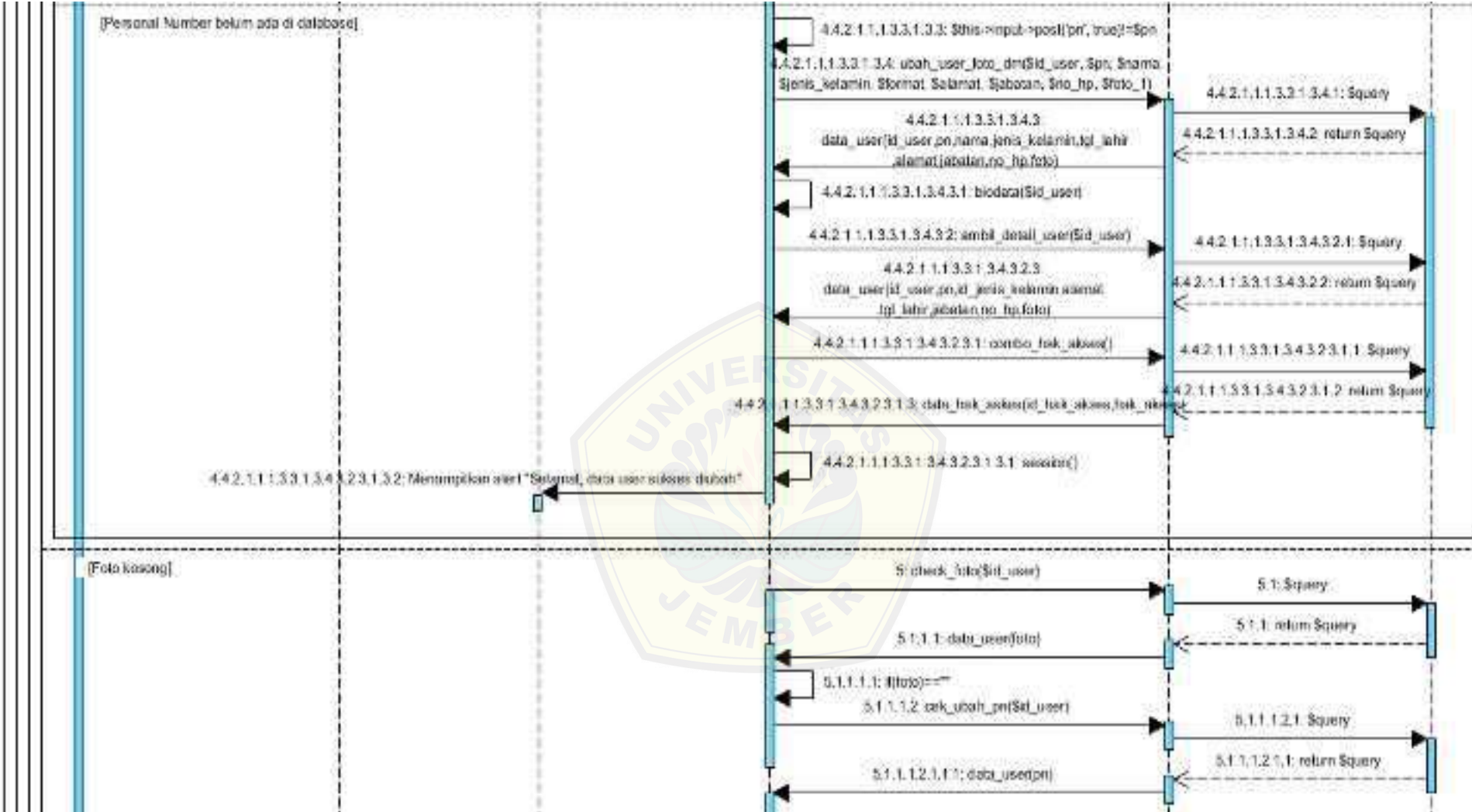


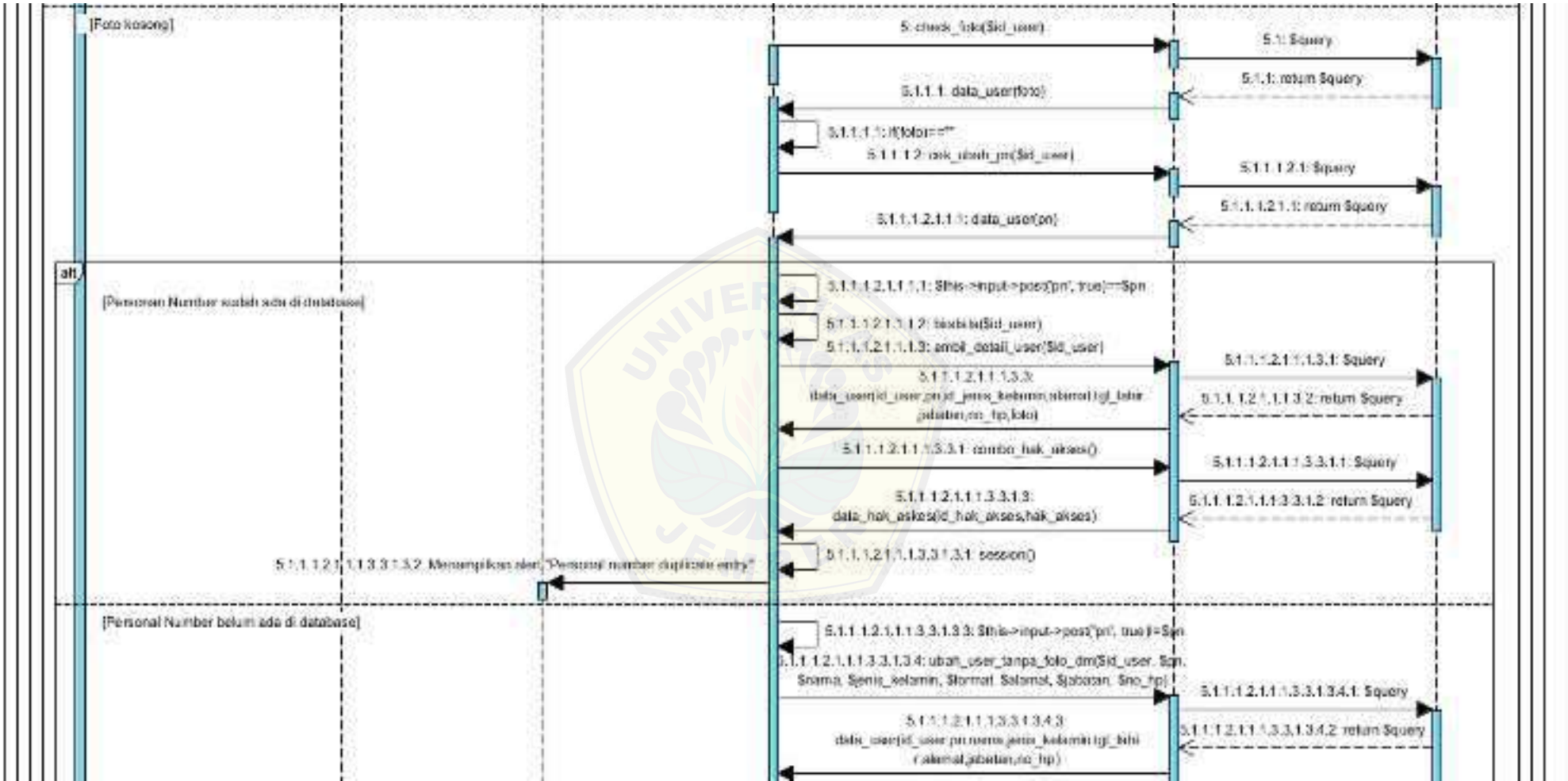


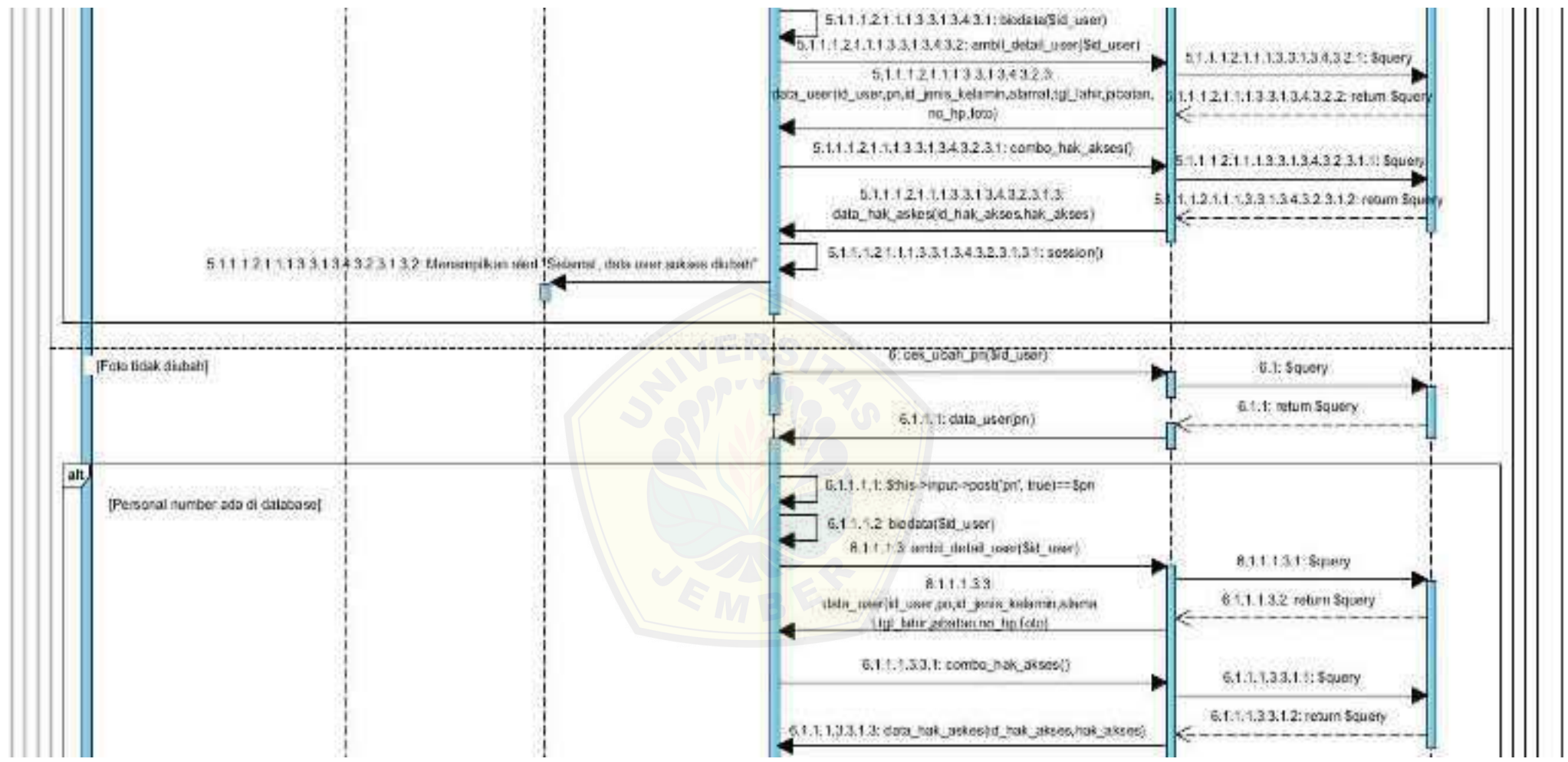
C.14 Sequence Diagram Mengelola Biodata

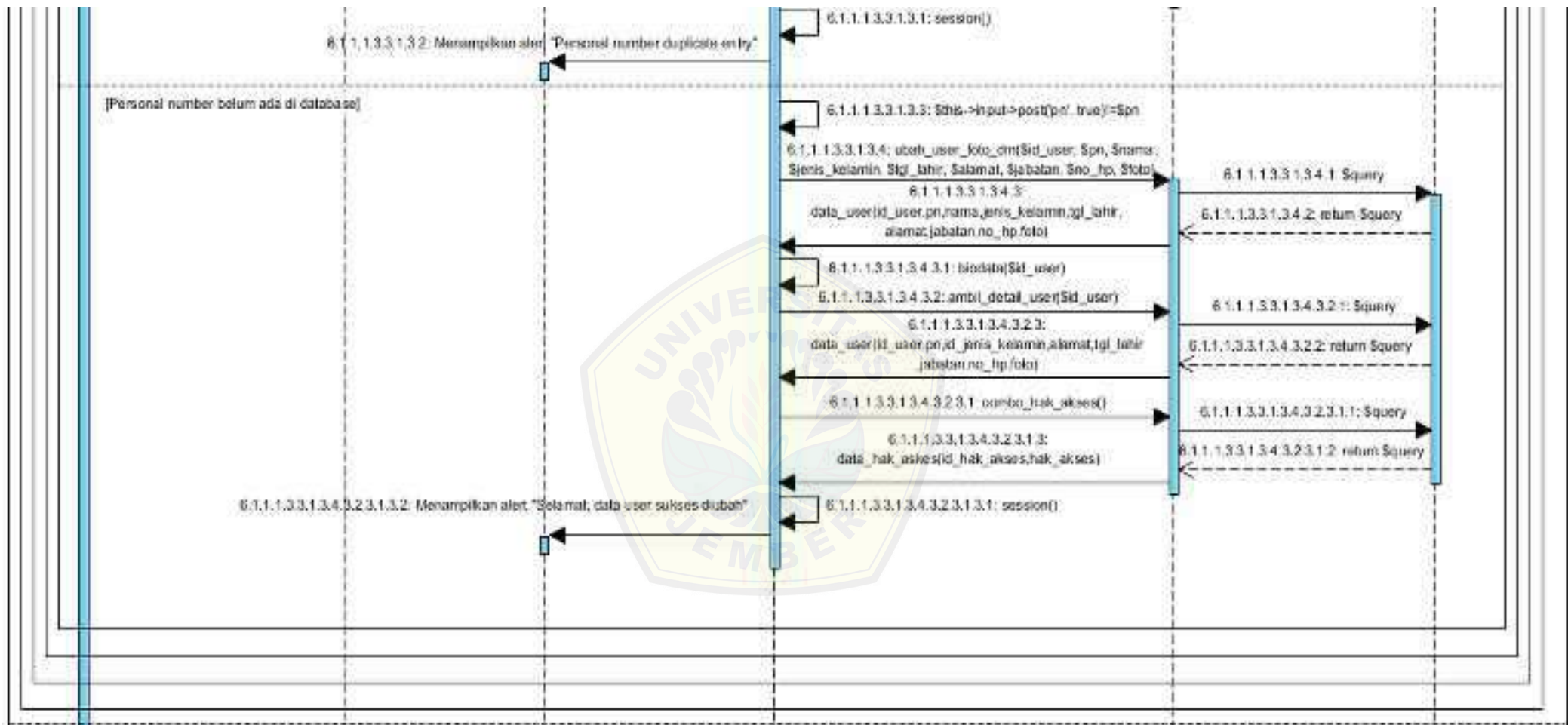


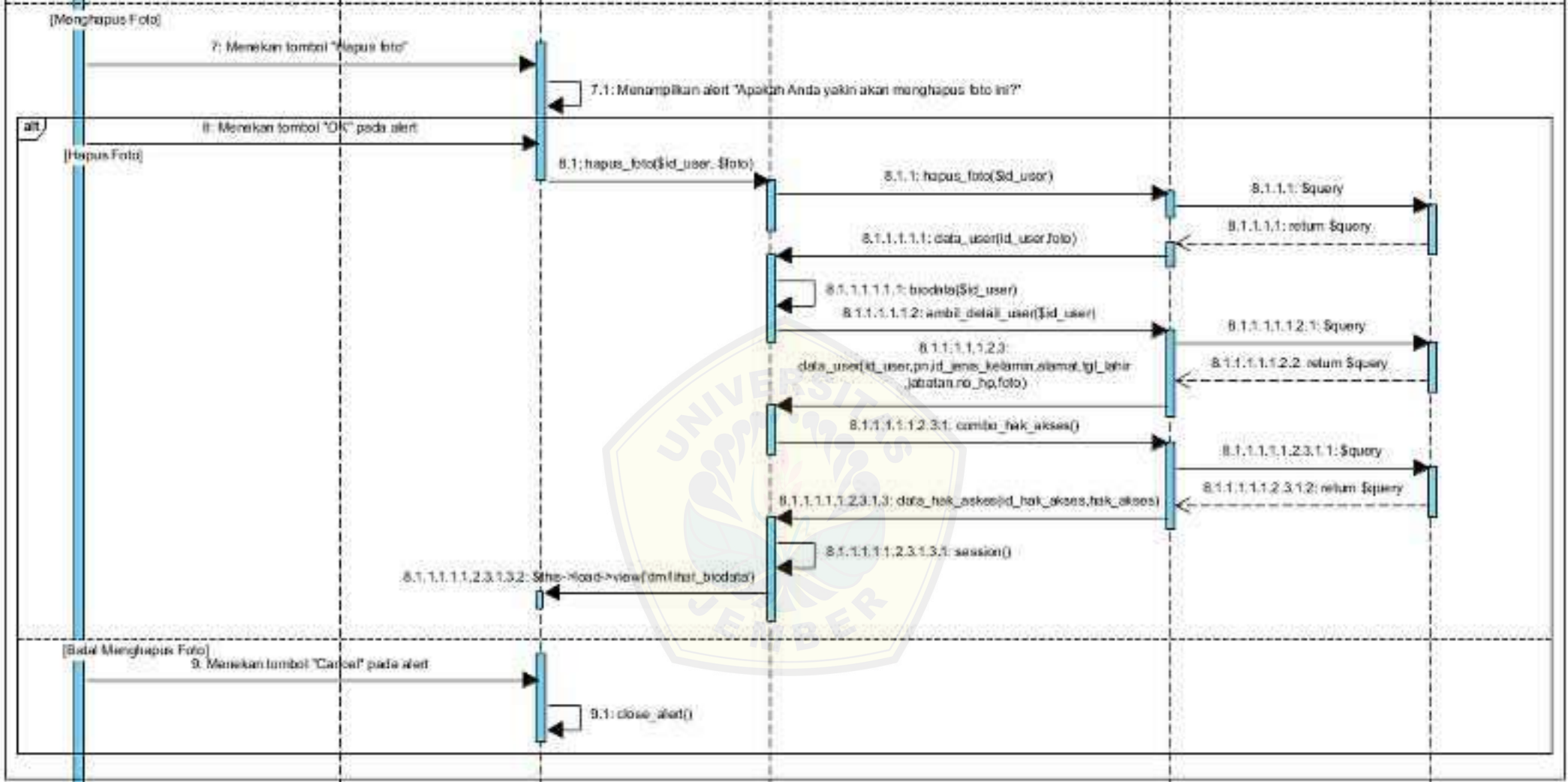




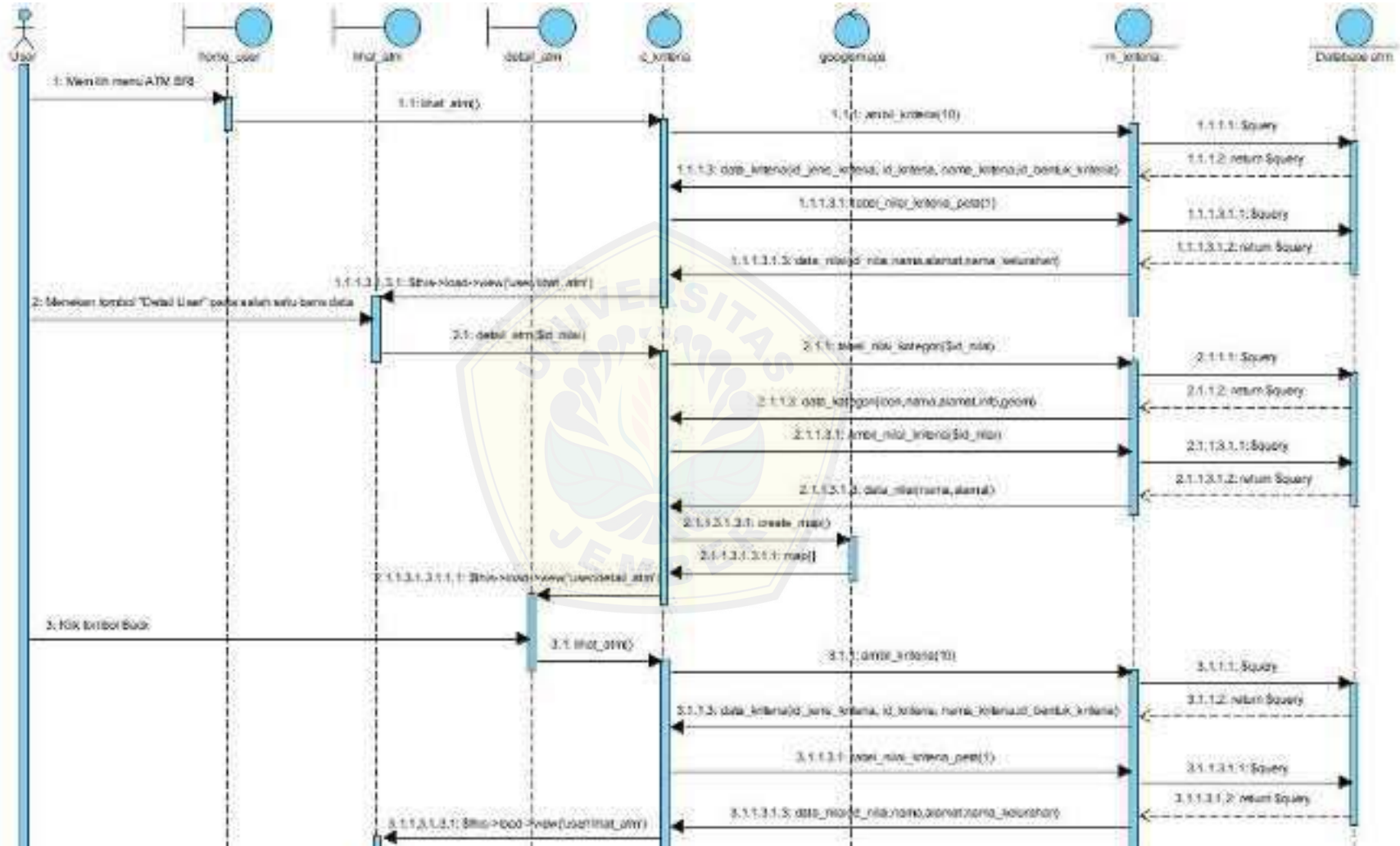




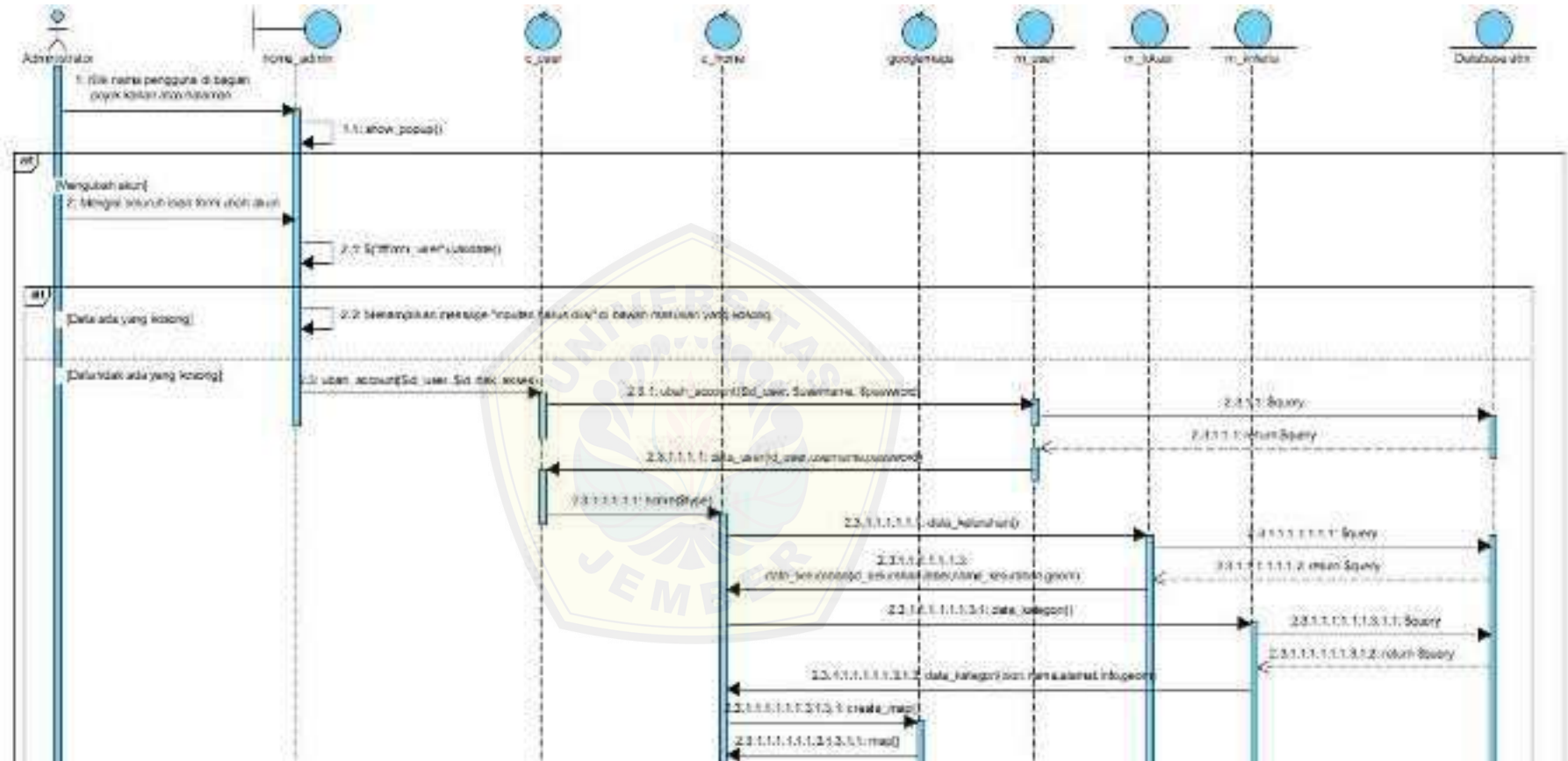




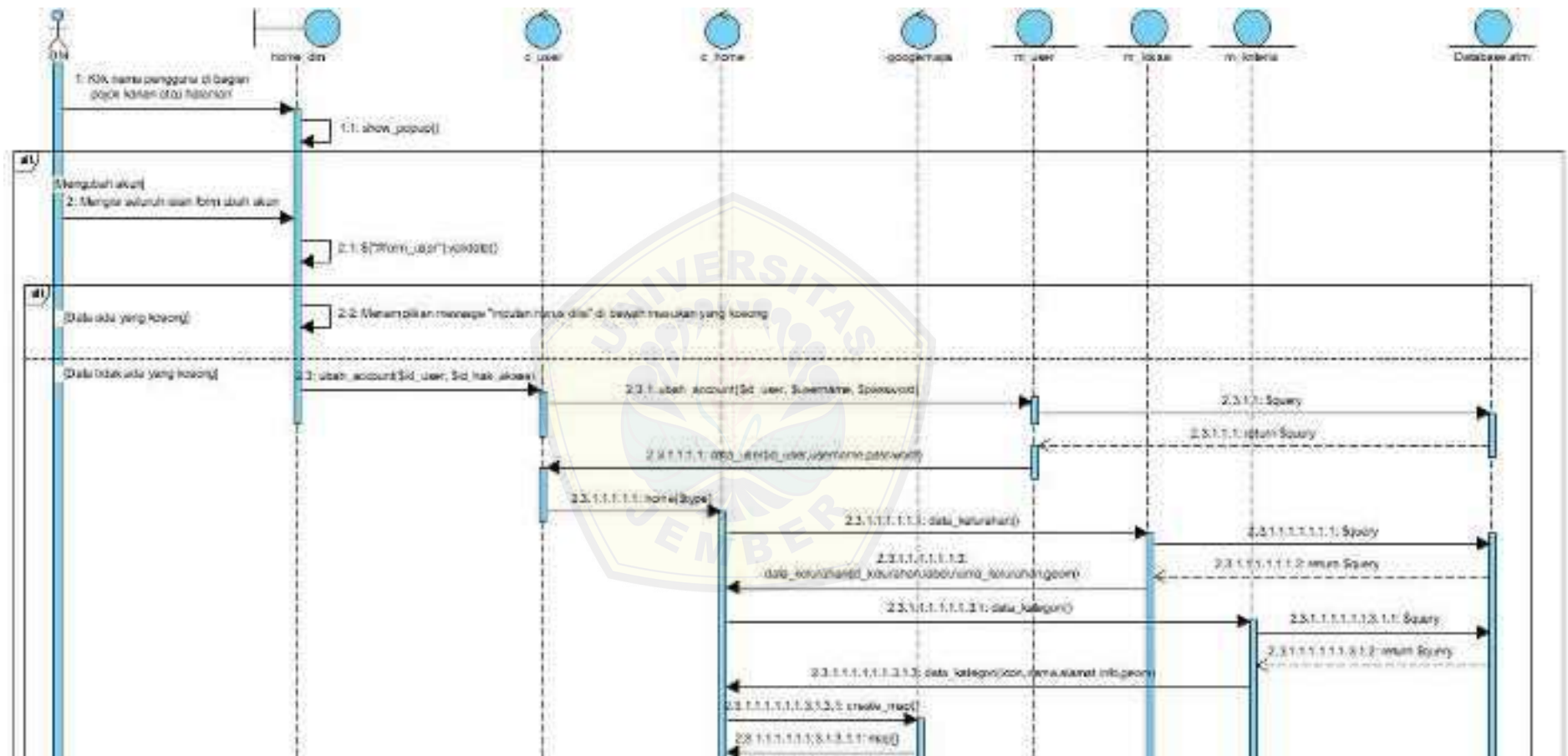
C.15 Sequence Diagram Melihat Data ATM

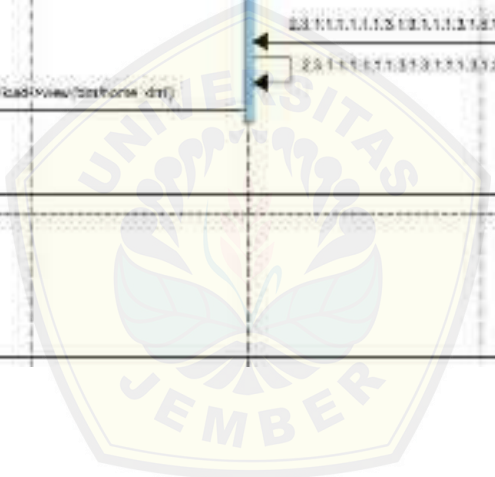
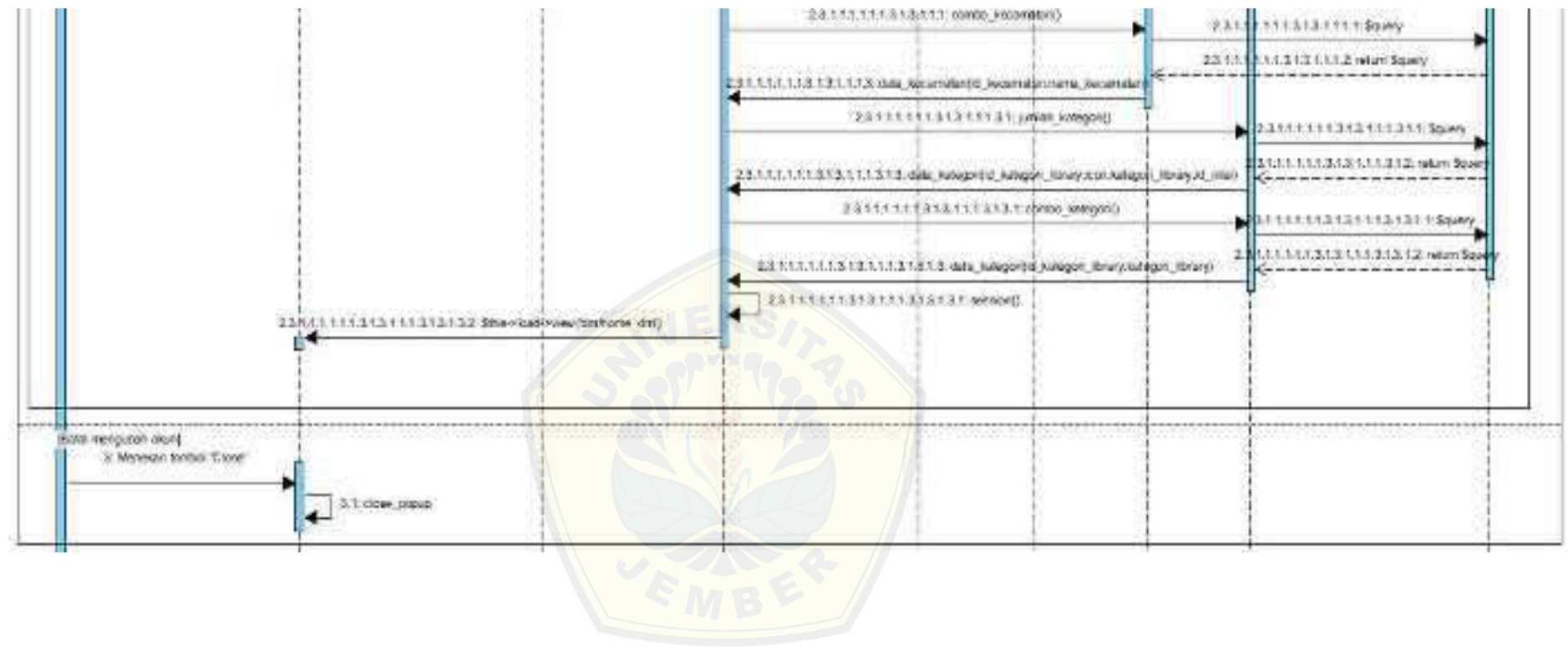


C.16 Sequence Diagram Mengubah Akun Admin



C.17 Sequence Diagram Mengubah Akun Pengambil Keputusan





D. Kode Program

D.1 Kode Program Pada *Package Controllers*

1. *Class c_dss*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_dss extends CI_Controller {

    function __construct() {
        parent::__construct();
    }

    //=====DM=====
    public function lihat_analisa_dm($id) {
        $data1['tabel_transaksi'] = $this->m_dss->tabel_transaksi($id);
        $data = $this->session();
        $data['content'] = $this->load->view('dm/lihat_analisa', $data1, true);
        $this->load->view('layout/template', $data);
    }

    public function lihat_proses1($id_transaksi, $id_user) {
        $data1['tabel_kriteria'] = $this->m_kriteria->tabel_kriteria();
        $data1['data_ambil'] = $this->m_kriteria->ambil_jumlah_kriteria();
        $data1['detail'] = $this->m_dss->ambil_detail_transaksi($id_transaksi, $id_user);
        $data = $this->session();
        $data['content'] = $this->load->view('dm/proses1', $data1, true);
        $this->load->view('layout/template', $data);
    }

    public function ubah_proses1($id_transaksi, $id_user) {
        $query = $this->m_dss->hapus_analisa_dm($id_transaksi, $id_user);
        if ($query) {
            $data1['tabel_kriteria'] = $this->m_kriteria->tabel_kriteria();
            $data1['data_ambil'] = $this->m_kriteria->ambil_jumlah_kriteria();
            $data1['detail'] = $this->m_dss->ambil_detail_transaksi($id_transaksi, $id_user);
            $data = $this->session();
            $data['content'] = $this->load->view('dm/proses1', $data1, true);
            $this->load->view('layout/template', $data);
        }
    }

    public function tambah_kriteria_dm($id_transaksi) {
        $data1['option'] = $this->m_kriteria->combo_sifat_kriteria();
        $data1['jenis'] = $this->m_kriteria->combo_jenis_kriteria();
        $data1['bentuk'] = $this->m_kriteria->combo_bentuk_kriteria();
        $data1['tabel_transaksi'] = $this->m_dss->ambil_analisa_dss($id_transaksi);
        $data = $this->session();
        $data['content'] = $this->load->view('dm/tambah_kriteria', $data1, true);
        $this->load->view('layout/template', $data);
    }

    public function proses_tambah_kriteria_dm($id_transaksi) {
        $nama_kriteria = $this->input->post('nama', true);
        $sid_sifat_kriteria = $this->input->post('sifat_kriteria', true);
        $sid_jenis_kriteria = $this->input->post('jenis_kriteria', true);
        $sid_bentuk_kriteria = '2';
    }
}

```

```

$ket = $this->input->post('ket', true);
$query = $this->m_kriteria->tambah_kriteria($nama_kriteria, $sid_sifat_kriteria, $ket, $sid_jenis_kriteria,
$sid_bentuk_kriteria);
if ($query) {
    $sid_kriteria = $this->m_dss->ambil_idakhir_kriteria();
    $sid_akhir = $sid_kriteria->id_kriteria;
    redirect("c_dss/lihat_range/$sid_akhir/$sid_transaksi");
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Kriteria Gagal di Simpan");
    redirect("c_dss/tambah_kriteria/$sid_transaksi");
}
}

public function lihat_range($sid_kriteria, $sid_transaksi) {
    $data1['range_kriteria'] = $this->m_kriteria->range_kriteria($sid_kriteria);
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($sid_kriteria);
    //$data1['jenis'] = $this->m_kriteria->ambil_jenis_kriteria($sid_kriteria);
    $data1['tabel_transaksi'] = $this->m_dss->ambil_analisa_dss($sid_transaksi);
    $data = $this->session();
    $data['content'] = $this->load->view('dm/lihat_range', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_tambah_range($sid_kriteria, $sid_transaksi) {
    $no = $this->input->post('no', true);
    for ($a = 1; $a <= $no; $a++) {
        $batas_atas = $this->input->post('batas_atas' . $a, true);
        $batas_bawah = $this->input->post('batas_bawah' . $a, true);
        $range = $this->input->post('range' . $a, true);
        $nk = $this->input->post('nk' . $a, true);
        $query = $this->m_kriteria->tambah_range_kriteria($batas_atas, $batas_bawah, $nk, $sid_kriteria,
$range);
    }
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Range Kriteria Sukses di Simpan");
        redirect("c_dss/lihat_range/$sid_kriteria/$sid_transaksi");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Range Kriteria Gagal di Simpan");
        redirect("c_dss/lihat_range/$sid_kriteria/$sid_transaksi");
    }
}

public function ubah_range($sid_range, $sid_kriteria, $sid_transaksi) {
    $data1['data_ambil'] = $this->m_kriteria->ambil_range($sid_kriteria, $sid_range);
    $data1['jenis'] = $this->m_kriteria->ambil_jenis_kriteria($sid_kriteria);
    $data1['tabel_transaksi'] = $this->m_dss->ambil_analisa_dss($sid_transaksi);
    $data = $this->session();
    $data['content'] = $this->load->view('dm/ubah_range', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_ubah_range($sid_range, $sid_kriteria, $sid_transaksi) {
    $batas_atas = $this->input->post('batas_atas', true);
    $batas_bawah = $this->input->post('batas_bawah', true);
    $range = $this->input->post('range', true);
    $nk = $this->input->post('nk', true);
    $query = $this->m_kriteria->ubah_range_kriteria($batas_atas, $batas_bawah, $nk, $sid_range, $sid_kriteria,
$range);
    if ($query) {

```

```

$this->session->set_userdata('operation', "sukses");
$this->session->set_userdata('message', "Data Range Kriteria Sukses di Ubah");
redirect("c_dss/lihat_range/$id_kriteria/$id_transaksi");
} else {
$this->session->set_userdata('operation', "gagal");
$this->session->set_userdata('message', "Data Range Kriteria gagal di Ubah");
redirect("c_dss/lihat_range/$id_kriteria/$id_transaksi");
}
}

public function hapus_range($id_range, $id_kriteria, $id_transaksi) {
$query = $this->m_kriteria->hapus_range($id_range);
if ($query) {
$this->session->set_userdata('operation', "sukses");
$this->session->set_userdata('message', "Data Range Kriteria Sukses di Hapus");
redirect("c_dss/lihat_range/$id_kriteria/$id_transaksi");
} else {
$this->session->set_userdata('operation', "gagal");
$this->session->set_userdata('message', "Data Range Kriteria gagal di Hapus");
redirect("c_dss/lihat_range/$id_kriteria/$id_transaksi");
}
}

public function tambah_nilai_kriteria($id_kriteria, $id_transaksi) {
$data1['kelurahan'] = $this->m_lokasi->tabel_kelurahan();
$data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
$data1['tabel_transaksi'] = $this->m_dss->ambil_analisa_dss($id_transaksi);
$data1['nilai'] = $this->m_kriteria->tabel_nilai_kriteria($id_kriteria);
$data = $this->session();
$data['content'] = $this->load->view('dm/tambah_nilai_kriteria', $data1, true);
$this->load->view('layout/template', $data);
}

public function proses_tambah_nilai($id_kriteria, $id_transaksi, $id_user, $id_jenis_kriteria) {
$no = $this->input->post('no', true);
if ($id_jenis_kriteria == '2') {
for ($a = 1; $a < $no; $a++) {
$id_kelurahan = $this->input->post('id_kelurahan' . $a, true);
$nilai = $this->input->post('nilai' . $a, true);
$query = $this->m_kriteria->tambah_nilai_kriteria($id_kelurahan, $id_kriteria, $nilai);
}
if ($query) {
$this->session->set_userdata('operation', "sukses");
$this->session->set_userdata('message', "Kriteria Sukses di Simpan");
redirect("c_dss/lihat_proses1/$id_transaksi/$id_user");
} else {
$this->session->set_userdata('operation', "gagal");
$this->session->set_userdata('message', "Kriteria Gagal di Simpan");
redirect("c_dss/lihat_proses1/$id_transaksi/$id_user");
}
} else {
for ($a = 1; $a < $no; $a++) {
$id_kelurahan = $this->input->post('id_kelurahan' . $a, true);
$niliasli = $this->input->post('niliasli' . $a, true);
$query = $this->m_kriteria->tambah_nilai_kriteriaasli($id_kelurahan, $id_kriteria, $niliasli);
}
if ($query) {
$nilai = $this->m_kriteria->ambil_nilai_kepentingansemua($id_kriteria);
foreach ($nilai as $row) {
$simpan = $row->nilai_kepentingan;
$id_kelurahan = $row->id_kelurahan;
$id_kriteria = $row->id_kriteria;
}
}
}
}

```

```

        $query2 = $this->m_kriteria->ubah_nilai_asli($id_kelurahan, $id_kriteria, $simpan);
    }
    if ($query2) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Kriteria Sukses di Simpan");
        redirect("c_dss/lihat_proses1/$id_transaksi/$id_user");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Kriteria Gagal di Simpan");
        redirect("c_dss/lihat_proses1/$id_transaksi/$id_user");
    }
}
}
}

public function lihat_hasil_dss($id_transaksi, $id_user) {
    $this->load->library('googlemaps');
    $config['center'] = '-8.239383846027533, 113.6823472357595';
    $config['zoom'] = 10;
    $this->googlemaps->initialize($config);
    $kelurahan = $this->m_dss->tabel_lihat_dss($id_transaksi, $id_user);
    $kat = $this->m_dss->data_atm($id_transaksi);
    foreach ($kelurahan as $value) {
        $marker2 = array();
        $polygon = str_replace("MULTIPOLYGON", "", $value->lat);
        $polygon = str_replace("(", "", $polygon);
        $polygon = str_replace(")", "", $polygon);
        $polygon = str_replace(",", ";", $polygon);
        $polygon = str_replace(" ", "", $polygon);
        $data = explode(";", $polygon);
        $label = $value->label;
        $nama = $value->nama_kelurahan;
        $ranking = $value->ranking;
        if ($ranking == '1') {
            $marker2['fillColor'] = '#FF0000';
        }
        if ($ranking == '2') {
            $marker2['fillColor'] = '#008000';
        }
        if ($ranking == '3') {
            $marker2['fillColor'] = '#1E90FF';
        }
        if ($ranking == '4') {
            $marker2['fillColor'] = '#FFFF00';
        }
        if ($ranking == '5') {
            $marker2['fillColor'] = '#A0522D';
        }
        if ($ranking == '6') {
            $marker2['fillColor'] = '#8B008B';
        }
        if ($ranking == '7') {
            $marker2['fillColor'] = '#FF8C00';
        }
        if ($ranking == '8') {
            $marker2['fillColor'] = '#FF1493';
        }
        if ($ranking == '9') {
            $marker2['fillColor'] = '#000000';
        }
        if ($ranking == '10') {
            $marker2['fillColor'] = '#00FFFF';
        }
    }
}

```

```

    }
    if ($ranking == '11') {
        $marker2['fillColor'] = '#ADFF2F';
    }
    if ($ranking == '12') {
        $marker2['fillColor'] = '#F5F5F5';
    }
    }

    for ($i = 0; $i < count($data); $i++) {
        $koordinat = explode(",", $data[$i]);
        $marker2['points'][] = $koordinat[1] . ", " . $koordinat[0];
        $marker2['onclick'] = "
            var infowindow = new google.maps.InfoWindow(
                {
                    size: new google.maps.Size(150,50)
                });
            infowindow.setContent('<p>Ranking $ranking.</p><p>$nama</p>');
            if (event) {
                point = event.latLng;
            }
            infowindow.setPosition(point);
            infowindow.open(map);";
    }
    $this->googlemaps->add_polygon($marker2);
}
foreach ($kat as $value) {
    $markerSkIh = array();
    $icon = $value->icon;
    $markerSkIh['icon'] = "http://localhost/dss_atm/temp_upload/$icon";
    $markerSkIh['position'] = "$value->lat, $value->long";
    $markerSkIh['infowindow_content'] = "<p>$value->nama</p><p>$value->alamat</p><p>$value->info</p>";
    $this->googlemaps->add_marker($markerSkIh);
}
$data1['map'] = $this->googlemaps->create_map();
$data1['tabel_dss'] = $this->m_dss->tabel_lihat_dss($id_transaksi, $id_user);
$data1['terbaik'] = $this->m_dss->ambil_ranking_terbaik($id_transaksi, $id_user);
$data1['user'] = $this->m_user->ambil_user($id_user);
$data = $this->session();
$data['content'] = $this->load->view('dm/lihat_hasil_dss', $data1, true);
$this->load->view('layout/template', $data);
}

public function lihat_hasil_gdss($id_transaksi, $id_hak_akses) {
    $this->load->library('googlemaps');
    $config['center'] = '-8.239383846027533, 113.6823472357595';
    $config['zoom'] = 10;
    $this->googlemaps->initialize($config);
    $kelurahan = $this->m_dss->peta_gdss($id_transaksi);
    $kat = $this->m_dss->data_atm($id_transaksi);
    foreach ($kelurahan as $value) {
        $marker2 = array();
        $polygon = str_replace("MULTIPOLYGON", "", $value->lat);
        $polygon = str_replace("(", "", $polygon);
        $polygon = str_replace(")", "", $polygon);
        $polygon = str_replace(",", ";", $polygon);
        $polygon = str_replace(" ", "", $polygon);
        $data = explode(";", $polygon);
        $label = $value->label;
        $nama = $value->nama_kelurahan;
        $ranking = $value->ranking;
    }
}

```



```

if ($ranking == '1') {
    $marker2['fillColor'] = '#FF0000';
}
if ($ranking == '2') {
    $marker2['fillColor'] = '#008000';
}
if ($ranking == '3') {
    $marker2['fillColor'] = '#1E90FF';
}
if ($ranking == '4') {
    $marker2['fillColor'] = '#FFFF00';
}
if ($ranking == '5') {
    $marker2['fillColor'] = '#A0522D';
}
if ($ranking == '6') {
    $marker2['fillColor'] = '#8B008B';
}
if ($ranking == '7') {
    $marker2['fillColor'] = '#FF8C00';
}
if ($ranking == '8') {
    $marker2['fillColor'] = '#FF1493';
}
if ($ranking == '9') {
    $marker2['fillColor'] = '#000000';
}
if ($ranking == '10') {
    $marker2['fillColor'] = '#00FFFF';
}
if ($ranking == '11') {
    $marker2['fillColor'] = '#ADFF2F';
}
if ($ranking == '12') {
    $marker2['fillColor'] = '#F5F5F5';
}
for ($i = 0; $i < count($data); $i++) {
    $koordinat = explode(",", $data[$i]);
    $marker2['points'][] = $koordinat[1] . "," . $koordinat[0];
    $marker2['onclick'] = "
        var infowindow = new google.maps.InfoWindow(
            {
                size: new google.maps.Size(150,50)
            });
        infowindow.setContent('<p>Ranking $ranking.</p><p>$nama</p>');
        if (event) {
            point = event.latLng;
        }
        infowindow.setPosition(point);
        infowindow.open(map);";
}
$this->googlemaps->add_polygon($marker2);
}
foreach ($kat as $value) {
    $markerSkh = array();
    $icon = $value->icon;
    $markerSkh['icon'] = "http://localhost/dss_atm/temp_upload/$icon";
    $markerSkh['position'] = "$value->lat, $value->long";
    $markerSkh['infowindow_content'] = "<p>$value->nama</p><p>$value->alamat</p><p>$value->info</p>";
    $this->googlemaps->add_marker($markerSkh);
}

```

```

}
$data1['map'] = $this->googlemaps->create_map();
$data1['user'] = $this->m_dss->ambil_vikor_user($id_transaksi);
$data1['metrik'] = $this->m_dss->ambil_vikor($id_transaksi);
$data1['tabel_gdss'] = $this->m_dss->tabel_lihat_gdss($id_transaksi);
$data1['terbaik'] = $this->m_dss->ambil_ranking_terbaik_gdss($id_transaksi);
$data = $this->session();
if ($id_hak_akses == '2') {
    $data['content'] = $this->load->view('dm/lihat_hasil_gdss', $data1, true);
} else {
    $data['content'] = $this->load->view('admin/lihat_hasil_gdss', $data1, true);
}
$this->load->view('layout/template', $data);
}

//=====MULAI
METODE=====
public function proses1_tambah_bobot_awal($id_transaksi, $id_user) {
    $no = $this->input->post('no', true);
    $simpan = 0;
    for ($a = 1; $a < $no; $a++) {
        $cek = $this->input->post('check' . $a, true);
        if ($cek != null) {
            $bobot_awal = $this->input->post('bobot' . $a, true);
            $simpan = $simpan + $bobot_awal;
        }
    }
    if ($simpan == 1) {
        for ($a = 1; $a < $no; $a++) {
            $cek = $this->input->post('check' . $a, true);
            if ($cek != null) {
                $id_kriteria = $this->input->post('id_kriteria' . $a, true);
                $bobot_awal = $this->input->post('bobot' . $a, true);
                $query = $this->m_dss->tambah_bobot_awal($id_transaksi, $id_user, $id_kriteria, $bobot_awal);
            }
        }
        if ($query) {
            redirect("c_dss/metode_entropy/$id_transaksi/$id_user");
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Gagal Menambah Bobot Awal");
            redirect("c_dss/lihat_proses1/$id_transaksi/$id_user");
        }
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Jumlah bobot awal harus sama dengan 1");
        redirect("c_dss/lihat_proses1/$id_transaksi/$id_user");
    }
}

//METODE ENTROPY
public function metode_entropy($id_transaksi, $id_user) {
    $metrik = array();
    $nilai = array();
    $temp = array();
    $nilaimax = array();
    $nilaimin = array();
    $normalisasi = array();
    $jum = array();
    $jumlen = array();
    $jumentropy = 0;
    $jumlamda = 0;
}

```

```

$probabilitas = array();
$lenprobabilitas = array();
$metrikentropy = array();
$lamda = array();
$lamdabobot = array();
$entropy = array();
$ambildata = $this->m_dss->metrik($id_transaksi, $id_user);
$kelurahan = $this->m_dss->ambil_kelurahan($id_transaksi, $id_user);
$kriteria = $this->m_dss->ambil_kriteria($id_transaksi, $id_user);
$jumkriteria = count($kriteria);
$jumdesa = count($kelurahan);
$b = 0;
foreach ($kriteria as $row) {
    $bobotawal[$b] = $row['bobot_awal'];
    $sifat[$b] = $row['id_sifat_kriteria'];
    $id_krit[$b] = $row['id_kriteria'];
    $b++;
}
//=====menyimpan nilai kepentingan ke metrik 2D=====
$kel = "";
$dcs = -1;
$krit = 0;
foreach ($ambildata as $row) {
    if ($kel != $row['nama_kelurahan']) {
        $krit = 0;
        $dcs++;
        $kel = $row['nama_kelurahan'];
        $nilai[$dcs][$krit] = $row['nilai_kepentingan'];
        $krit++;
    } else {
        $nilai[$dcs][$krit] = $row['nilai_kepentingan'];
        $krit++;
    }
}
//=====1. MEMBUAT METRIK KRITERIA=====
for ($j = 0; $j < $jumkriteria; $j++) {
    for ($i = 0; $i < $jumdesa; $i++) {
        $metrik[$i][$j] = $nilai[$i][$j];
    }
}
//mencari nilai maksimal per kriteria
for ($j = 0; $j < $jumkriteria; $j++) {
    for ($i = 0; $i < $jumdesa; $i++) {
        $temp[$i] = $metrik[$i][$j];
    }
    $nilaimax [$j] = max($temp);
    unset($temp);
}
//mencari nilai minimal per kriteria
for ($j = 0; $j < $jumkriteria; $j++) {
    for ($i = 0; $i < $jumdesa; $i++) {
        $temp[$i] = $metrik[$i][$j];
    }
    $nilaimin [$j] = min($temp);
    unset($temp);
}
//=====2. MEMBUAT METRIK TERNORMALISASI=====
for ($j = 0; $j < $jumkriteria; $j++) {
    for ($i = 0; $i < $jumdesa; $i++) {
        if ($sifat[$j] == '1') {
            $normalisasi[$i][$j] = $metrik[$i][$j] / $nilaimax [$j]; //benefit
        } else {

```

```

        $normalisasi[$i][$j] = $nilaimin [$j] / $metrik[$i][$j]; //cost
    }
}
}
//=====METODE ENTROPY=====
//mencari jumlah nilai ternormalisasi per kriteria
for ($j = 0; $j < $jumkriteria; $j++) {
    $simpan = 0;
    for ($i = 0; $i < $jumdesa; $i++) {
        $simpan = $simpan + $normalisasi[$i][$j];
    }
    $jum[$j] = $simpan;
}
//=====3. MEMBUAT METRIK PROBABILITAS=====
for ($j = 0; $j < $jumkriteria; $j++) {
    for ($i = 0; $i < $jumdesa; $i++) {
        $probabilitas[$i][$j] = $normalisasi[$i][$j] / $jum[$j];
    }
}
//nilai probabilitas * LN(nilai probabilitas )
for ($j = 0; $j < $jumkriteria; $j++) {
    for ($i = 0; $i < $jumdesa; $i++) {
        $lenprobabilitas[$i][$j] = $probabilitas[$i][$j] * log(($probabilitas[$i][$j]), M_E);
    }
}
//jumlah len probabilitas per kriteria
for ($j = 0; $j < $jumkriteria; $j++) {
    $simpan = 0;
    for ($i = 0; $i < $jumdesa; $i++) {
        $simpan = $simpan + $lenprobabilitas[$i][$j];
    }
    $jumlen[$j] = $simpan;
}
//=====METRIK ENTROPY PER KRITERIA=====
for ($j = 0; $j < $jumkriteria; $j++) {
    $metrikentropy[$j] = ((-1) / log($jumdesa, M_E)) * $jumlen[$j];
}
//=====LAMDA ENTROPY PER KRITERIA=====
//jumlah nilai entropy
for ($j = 0; $j < $jumkriteria; $j++) {
    $jumentropy = $jumentropy + $metrikentropy[$j];
}
//hitung lamda
for ($j = 0; $j < $jumkriteria; $j++) {
    if(($jumkriteria - $jumentropy)==0){
        $lamda[$j]=0;
    }
    else{
        $lamda[$j] = (1 / ($jumkriteria - $jumentropy)) * (1 - $metrikentropy[$j]);
    }
}
//=====IAMDA KALI BOBOT AWAL PER KRITERIA=====
for ($j = 0; $j < $jumkriteria; $j++) {
    $lamdabobot[$j] = $lamda[$j] * $bobotawal[$j];
}
//=====JUMLAH IAMDA KALI BOBOT AWAL PER
KRITERIA=====
for ($j = 0; $j < $jumkriteria; $j++) {
    $jumlamda = $jumlamda + $lamdabobot[$j];
}
//=====HASIL BOBOT ENTROPY PER KRITERIA=====
for ($j = 0; $j < $jumkriteria; $j++) {
    if ($lamdabobot[$j] == 0) {

```

```

        $entropy[$j] = 0;
    } else {
        $entropy[$j] = $lamdabobot[$j] / $jumlamda;
    }
}
//SIMPAN ENTROPY
for ($j = 0; $j < $jumkriteria; $j++) {
    $bobot_entropy = $entropy[$j];
    $id_kriteria = $id_krit[$j];
    $query = $this->m_dss->simpan_bobot_entropy($id_transaksi, $id_user, $id_kriteria, $bobot_entropy);
}
if ($query) {
    $data1['normalisasi'] = $normalisasi;
    $data1['probabilitas'] = $probabilitas;
    $data1['lenprobabilitas'] = $lenprobabilitas;
    $data1['metrikentropy'] = $metrikentropy;
    $data1['lamda'] = $lamda;
    $data1['lamdabobot'] = $lamdabobot;
    $data1['entropy'] = $entropy;
    $data1['jumlen'] = $jumlen;
    $data1['kriteria'] = $this->m_dss->ambil_kriteria($id_transaksi, $id_user);
    $data1['metrik'] = $this->m_dss->metrik($id_transaksi, $id_user);
    $data1['kelurahan'] = $this->m_dss->ambil_kelurahan($id_transaksi, $id_user);
    $data1['detail'] = $this->m_dss->ambil_detail_transaksi($id_transaksi, $id_user);
    $data = $this->session();
    $data['content'] = $this->load->view('dm/metode_entropy', $data1, true);
    $this->load->view('layout/template', $data);
} else {
    redirect("c_dss/proses1_tambah_bobot_awal/$id_transaksi/$id_user");
}
}

public function metode_vikor($id_transaksi, $id_user) {
    $metrik = array();
    $nilai = array();
    $temp = array();
    $nilaimax = array();
    $nilaimin = array();
    $normalisasi = array();
    $idealpositif = array();
    $idealnegasif = array();
    $idealtebobot = array();
    $solusiteringgi = array();
    $solusiterendah = array();
    $rasio = array();
    $ambildata = $this->m_dss->metrik($id_transaksi, $id_user);
    $kelurahan = $this->m_dss->ambil_kelurahan($id_transaksi, $id_user);
    $kriteria = $this->m_dss->ambil_kriteria($id_transaksi, $id_user);
    $jumkriteria = count($kriteria);
    $jumdesa = count($kelurahan);
    //=====menyimpan nilai kepentingan ke metrik 2D=====
    $kel = "";
    $des = -1;
    $krit = 0;
    foreach ($ambildata as $row) {
        if ($kel != $row['nama_kelurahan']) {
            $krit = 0;
            $des++;
            $kel = $row['nama_kelurahan'];
            $nilai[$des][$krit] = $row['nilai_kepentingan'];
            $krit++;
        } else {

```

```

    $nilai[$des][$kрит] = $row['nilai_kepentingan'];
    $krit++;
  }
}
//=====1. MEMBUAT METRIK KRITERIA=====
for ($j = 0; $j < $jumkriteria; $j++) {
  for ($i = 0; $i < $jumdesa; $i++) {
    $metrik[$i][$j] = $nilai[$i][$j];
  }
}
//mencari nilai maksimal per kriteria
for ($j = 0; $j < $jumkriteria; $j++) {
  for ($i = 0; $i < $jumdesa; $i++) {
    $temp[$i] = $metrik[$i][$j];
  }
  $nilaimax [$j] = max($temp);
  unset($temp);
}
//mencari nilai minimal per kriteria
for ($j = 0; $j < $jumkriteria; $j++) {
  for ($i = 0; $i < $jumdesa; $i++) {
    $temp[$i] = $metrik[$i][$j];
  }
  $nilaimin [$j] = min($temp);
  unset($temp);
}
//=====2. MEMBUAT METRIK TERNORMALISASI=====
$b = 0;
foreach ($kriteria as $row) {
  $entropy[$b] = $row['bobot_entropy'];
  $sifat[$b] = $row['id_sifat_kriteria'];
  $b++;
}
for ($j = 0; $j < $jumkriteria; $j++) {
  for ($i = 0; $i < $jumdesa; $i++) {
    if ($sifat[$j] == '1') {
      $normalisasi[$i][$j] = $metrik[$i][$j] / $nilaimax [$j];
    } else {
      $normalisasi[$i][$j] = $nilaimin [$j] / $metrik[$i][$j];
    }
  }
}
//=====METODE VIKOR=====
//1.mencari nilai ideal positif dan negatif setiap kriteria dari metrik normalisasi
//ideal positif
for ($j = 0; $j < $jumkriteria; $j++) {
  for ($i = 0; $i < $jumdesa; $i++) {
    $temp[$i] = $normalisasi[$i][$j];
  }
  $idealpositif [$j] = max($temp);
  unset($temp);
}
//ideal negatif
for ($j = 0; $j < $jumkriteria; $j++) {
  for ($i = 0; $i < $jumdesa; $i++) {
    $temp[$i] = $normalisasi[$i][$j];
  }
  $idealnegasif [$j] = min($temp);
  unset($temp);
}
//2.mencari nilai solusi tertinggi dan terendah setiap alternatif
//nilai solusi ideal terbobot entropy

```

```

$bagi = array();
for ($j = 0; $j < $jumkriteria; $j++) {
  for ($i = 0; $i < $jumdesa; $i++) {
    $bagi[$j] = ($idealpositif [$j] - $idealnegasif [$j]);
    if ($bagi[$j] == '0') {
      $idealterbobot[$i][$j] = 0;
    } else {
      $idealterbobot[$i][$j] = $entropy[$j] * (($idealpositif [$j] - $normalisasi[$i][$j]) / $bagi[$j]);
    }
  }
}
//S = Nilai solusi tertinggi, jumlah ideal terbobot
for ($i = 0; $i < $jumdesa; $i++) {
  $simpan = 0;
  for ($j = 0; $j < $jumkriteria; $j++) {
    $simpan = $simpan + $idealterbobot[$i][$j];
  }
  $solusiteringgi[$i] = $simpan;
}
//R = Nilai solusi terendah, max ideal terbobot
for ($i = 0; $i < $jumdesa; $i++) {
  for ($j = 0; $j < $jumkriteria; $j++) {
    $temp[$j] = $idealterbobot[$i][$j];
  }
  $solusiterendah [$i] = max($temp);
  unset($temp);
}
//3. Menghitung Nilai Rasio (Q) setiap alternatif
//menghitung nilai Smax
for ($i = 0; $i < $jumdesa; $i++) {
  $temp[$i] = $solusiteringgi[$i];
}
$max = max($temp);
unset($temp);
//menghitung nilai Smin
for ($i = 0; $i < $jumdesa; $i++) {
  $temp[$i] = $solusiteringgi[$i];
}
$min = min($temp);
unset($temp);
//menghitung nilai Rmax
for ($i = 0; $i < $jumdesa; $i++) {
  $temp[$i] = $solusiterendah[$i];
}
$Rmax = max($temp);
unset($temp);
//menghitung nilai Rmin
for ($i = 0; $i < $jumdesa; $i++) {
  $temp[$i] = $solusiterendah[$i];
}
$Rmin = min($temp);
unset($temp);

$kurangs = $smax - $smin;
$kurangr = $Rmax - $Rmin;
//menghitung nilai Q per alternatif
$v = 0.5;
for ($i = 0; $i < $jumdesa; $i++) {
  if ($kurangs == '0') {
    $hasil1 = 0;
  } else {
    $hasil1 = ($v * (($solusiteringgi[$i] - $smin) / ($kurangs)));
  }
}

```

```

    }
    if ($kurangr == '0') {
        $hasil2 = 0;
    } else {
        $hasil2 = ((1 - $v) * (($solusiterendah[$i] - $rmin) / ($kurangr)));
    }
    $rasio[$i] = $hasil1 + $hasil2;
}
//4. Mengurutkan Nilai S,R,Q setiap alternatif decreasing order
$a = 0;
foreach ($kelurahan as $row) {
    $desa[$a] = $row['nama_kelurahan'];
    $a++;
}

//mengurutkan Nilai S
$nilaiq = array();
$dessaq = array();
$nilais = array();
$dessa = array();
$nilair = array();
$dessar = array();
$surut = array();
for ($i = 0; $i < $jumdesa; $i++) {
    $surut[$i] = array(
        'alternatif' => $desa[$i],
        'ns' => $solusitertinggi[$i]
    );
}
$b = array();
$c = array();
foreach ($surut as $key => $value) {
    $b[$key] = $value['ns'];
}
natsort($b);
foreach ($b as $key => $value) {
    $c[] = $surut[$key];
}
foreach ($c as $row) {
    $nilais[] = $row['ns'];
    $dessa[] = $row['alternatif'];
}
for ($i = 0; $i < $jumdesa; $i++) {
    $ranks[$i] = array(
        'nilais' => $nilais[$i],
        'dessa' => $dessa[$i]
    );
}

//mengurutkan Nilai R
$surutr = array();
for ($i = 0; $i < $jumdesa; $i++) {
    $surutr[$i] = array(
        'alternatif' => $desa[$i],
        'nr' => $solusiterendah[$i]
    );
}
$br = array();
$cr = array();
foreach ($surutr as $key => $value) {
    $br[$key] = $value['nr'];
}

```



```

natsort($br);
foreach ($br as $key => $value) {
    $scr[] = $urutr[$key];
}
foreach ($scr as $row) {
    $nilair[] = $row['nr'];
    $desar[] = $row['alternatif'];
}
for ($i = 0; $i < $jumdesa; $i++) {
    $rankr [$i] = array(
        'nilair' => $nilair[$i],
        'desar' => $desar[$i]
    );
}

//mengurutkan Nilai Q
$urutq = array();
for ($i = 0; $i < $jumdesa; $i++) {
    $urutq [$i] = array(
        'alternatif' => $desa[$i],
        'nq' => $rasio[$i]
    );
}
$bq = array();
$cq = array();
$rankq = array();
foreach ($urutq as $key => $value) {
    $bq[$key] = $value['nq'];
}
natsort($bq);
foreach ($bq as $key => $value) {
    $cq[] = $urutq[$key];
}
foreach ($cq as $row) {
    $nilaiq[] = $row['nq'];
    $desaq[] = $row['alternatif'];
}
for ($i = 0; $i < $jumdesa; $i++) {
    $rankq [$i] = array(
        'nilaiq' => $nilaiq[$i],
        'desaq' => $desaq[$i]
    );
}

//5.CEK KONDISI 1 DAN 2
//- Hitung DQ
$dq = 1 / ($jumdesa - 1);
//KONDISI 1
$hasil = array();
if ($nilaiq[1] - $nilaiq[0] > $dq || $nilaiq[1] - $nilaiq[0] == $dq) {
    //KONDISI 2
    if ($desaq[0] == $desar[0] && $desaq[0] == $desas[0]) {
        //nilainya yg ranking 1 tetep
        $d = 0;
        $bantu = 0;
        for ($i = 0; $i < $jumdesa; $i++) {
            $d++;
            if ($i == 0) {
                $hasil [$i] = array(
                    'nilaiq' => $nilaiq[$i],
                    'desaq' => $desaq[$i],
                    'rankingq' => $d
                );
            }
        }
    }
}

```

```

    );
  }
  if ($i > 0) {
    if ($nilaiq[$i] == $nilaiq[$i - 1]) {
      $d--;
      $bantu++;
      $hasil [$i] = array(
        'nilaiq' => $nilaiq[$i],
        'desaq' => $desaq[$i],
        'rankingq' => $d
      );
    } else {
      $d = $d + $bantu;
      $bantu = 0;
      $hasil [$i] = array(
        'nilaiq' => $nilaiq[$i],
        'desaq' => $desaq[$i],
        'rankingq' => $d
      );
    }
  }
}
} else {
  //nilai ke 1 dan 2 jadi ranking 1
  $m = 0;
  $bantu1 = 0;
  for ($i = 0; $i < $jumdesa; $i++) {
    if ($i == 0 || $i == 1) {
      $hasil [$i] = array(
        'nilaiq' => $nilaiq[$i],
        'desaq' => $desaq[$i],
        'rankingq' => 1
      );
    }
    $m++;
    if ($i > 1) {
      if ($nilaiq[$i] == $nilaiq[$i - 1]) {
        $m--;
        $bantu1++;
        $hasil [$i] = array(
          'nilaiq' => $nilaiq[$i],
          'desaq' => $desaq[$i],
          'rankingq' => $m
        );
      } else {
        $m = $m + $bantu1;
        $bantu1 = 0;
        $hasil [$i] = array(
          'nilaiq' => $nilaiq[$i],
          'desaq' => $desaq[$i],
          'rankingq' => $m
        );
      }
    }
  }
}
} else {
  $o = 0;
  $g = 1;
  $ban = 0;
  for ($i = 0; $i < $jumdesa; $i++) {
    if ($nilaiq[$i] - $nilaiq[0] < $dq || $nilaiq[$i] - $nilaiq[0] == $dq) {

```

```

$hasil [$i] = array(
    'nilaiq' => $nilaiq[$i],
    'desaq' => $desaq[$i],
    'rankingq' => $g
);
$ban++;
} else {
    $o++;
    if ($i > 0) {
        if ($nilaiq[$i] == $nilaiq[$i - 1]) {
            $o--;
            $ban++;
            $hasil [$i] = array(
                'nilaiq' => $nilaiq[$i],
                'desaq' => $desaq[$i],
                'rankingq' => $o
            );
        } else {
            $o = $o + $ban;
            $ban = 0;
            $hasil [$i] = array(
                'nilaiq' => $nilaiq[$i],
                'desaq' => $desaq[$i],
                'rankingq' => $o
            );
        }
    }
}
}
}

$K = 0;
foreach ($hasil as $row) {
    $rankingfix[$K] = $row['rankingq'];
    $desafix[$K] = $row['desaq'];
    $lurah = $this->m_dss->cek_kelurahan($desafix[$K]);
    if ($lurah) {
        $id[$K] = $lurah->id_kelurahan;
    }
    $K++;
}
for ($i = 0; $i < $jumdesa; $i++) {
    $idkel = $id[$i];
    $ranking = $rankingfix[$i];
    $query = $this->m_dss->simpan_vikor($id_transaksi, $id_user, $idkel, $ranking);
}
$query1 = $this->m_dss->ubah_status_dm($id_transaksi, $id_user);
if ($query && $query1) {
    $data1['normalisasi'] = $normalisasi;
    $data1['idealpositif'] = $idealpositif;
    $data1['idealnegasif'] = $idealnegasif;
    $data1['idealterbot'] = $idealterbot;
    $data1['solusiteringgi'] = $solusiteringgi;
    $data1['solusiterendah'] = $solusiterendah;
    $data1['rasio'] = $rasio;
    $data1['rankq'] = $rankq;
    $data1['ranks'] = $ranks;
    $data1['rankr'] = $rankr;
    $data1['hasil'] = $hasil;
    $data1['kriteria'] = $this->m_dss->ambil_kriteria($id_transaksi, $id_user);
    $data1['metrik'] = $this->m_dss->metrik($id_transaksi, $id_user);
}

```

```

    $data1['kelurahan'] = $this->m_dss->ambil_kelurahan($id_transaksi, $id_user);
    $data1['detail'] = $this->m_dss->ambil_detail_transaksi($id_transaksi, $id_user);
    $data = $this->session();
    $data['content'] = $this->load->view('dm/metode_vikor', $data1, true);
    $this->load->view('layout/template', $data);
} else {
    redirect("c_dss/metode_entropy/$id_transaksi/$id_user");
}
}

public function metode_borda($id_transaksi) {
    $metrik = $this->m_dss->ambil_vikor($id_transaksi);
    $kelurahan = $this->m_dss->ambil_vikor_kelurahan($id_transaksi);
    $dm = $this->m_dss->ambil_vikor_user($id_transaksi);
    $jumuser = count($dm);
    $jumdesa = count($kelurahan);
    $awal = array();
    $nilai = array();
    //matrik ranking sebelum borda
    $kel = "";
    $des = -1;
    $user = 0;
    foreach ($metrik as $row) {
        if ($kel != $row['nama_kelurahan']) {
            $user = 0;
            $des++;
            $kel = $row['nama_kelurahan'];
            $nilai[$des][$user] = $row['ranking'];
            $user++;
        } else {
            $nilai[$des][$user] = $row['ranking'];
            $user++;
        }
    }
    //1.METRIK RANKING 2 D
    for ($j = 0; $j < $jumuser; $j++) {
        for ($i = 0; $i < $jumdesa; $i++) {
            $awal[$i][$j] = $nilai[$i][$j];
        }
    }
    //2. MASUKKAN POINT KE RANKING
    $x = 1;
    $y = $jumdesa;
    $z = 0;
    $hasilpoint = array();
    for ($j = 0; $j < $jumuser; $j++) {
        for ($i = 0; $i < $jumdesa; $i++) {
            if ($awal[$i][$j] == $y) {
                $hasilpoint[$i][$j] = $x;
            } else {
                for ($z = 1; $z < $jumdesa; $z++) {
                    if ($awal[$i][$j] == ($y - $z)) {
                        $hasilpoint[$i][$j] = $x + $z;
                    }
                }
            }
        }
    }
    //3. JUMLAHKAN POINT PER KELURAHAN
    //jumlah len probabilitas per kriteria
    $jumpoint = array();
    for ($i = 0; $i < $jumdesa; $i++) {

```

```

$simpan = 0;
for ($j = 0; $j < $jumuser; $j++) {
    $simpan = $simpan + $hasilpoint[$i][$j];
}
$jumpoint[$i] = $simpan;
}
//4. URUTKAN DARI BESAR KE KECIL
$a = 0;
foreach ($kelurahan as $row) {
    $desa[$a] = $row['nama_kelurahan'];
    $a++;
}
$nilaiborda = array();
$desaborda = array();
$urut = array();
for ($i = 0; $i < $jumdesa; $i++) {
    $urut [$i] = array(
        'alternatif' => $desa[$i],
        'point' => $jumpoint[$i]
    );
}
$b = array();
$c = array();
foreach ($urut as $key => $value) {
    $b[$key] = $value['point'];
}
arsort($b);
foreach ($b as $key => $value) {
    $c[] = $urut[$key];
}
foreach ($c as $row) {
    $nilaiborda[] = $row['point'];
    $desaborda[] = $row['alternatif'];
}
//BERIKAN RANKING, YANG SAMA DAPAT RANKING SAMA
$d = 0;
$bantu = 0;
for ($i = 0; $i < $jumdesa; $i++) {
    $d++;
    if ($i == 0) {
        $hasilborda [$i] = array(
            'ranking' => $nilaiborda[$i],
            'desa' => $desaborda[$i],
            'borda' => $d
        );
    }
    if ($i > 0) {
        if ($nilaiborda[$i] == $nilaiborda[$i - 1]) {
            $d--;
            $bantu++;
            $hasilborda [$i] = array(
                'ranking' => $nilaiborda[$i],
                'desa' => $desaborda[$i],
                'borda' => $d
            );
        }
        } else {
            $d = $d + $bantu;
            $bantu = 0;
            $hasilborda [$i] = array(
                'ranking' => $nilaiborda[$i],
                'desa' => $desaborda[$i],
                'borda' => $d
            );
        }
    }
}

```

```

    );
  }
}
}
//CEK UDAH ADA NILAI BORDA DISITU BELUM? KALO BELUM DITAMBAH, KALO SUDAH
DIUBAH??
$cekborda = $this->m_dss->cek_borda($id_transaksi);
$k = 0;
foreach ($hasilborda as $row) {
    $rankingborda[$k] = $row['borda'];
    $desaborda[$k] = $row['desa'];
    $nborda[$k] = $row['ranking'];
    $lurah = $this->m_dss->cek_kelurahan($desaborda[$k]);
    if ($lurah) {
        $id[$k] = $lurah->id_kelurahan;
    }
    $k++;
}
if ($cekborda == null) {
    for ($i = 0; $i < $jumdesa; $i++) {
        $idkel = $id[$i];
        $rank = $rankingborda[$i];
        $nilai_borda = $nborda[$i];
        $query = $this->m_dss->simpan_borda($id_transaksi, $idkel, $rank, $nilai_borda);
    }
} else {
    for ($i = 0; $i < $jumdesa; $i++) {
        $idkel = $id[$i];
        $rank = $rankingborda[$i];
        $nilai_borda = $nborda[$i];
        $query = $this->m_dss->ubah_borda($id_transaksi, $idkel, $rank, $nilai_borda);
    }
}
if ($query) {
    $data1['borda'] = $hasilborda;
    $data1['point'] = $hasilpoint;
    $data1['jumpoint'] = $jumpoint;
    $data1['user'] = $this->m_dss->ambil_vikor_user($id_transaksi);
    $data1['metrik'] = $this->m_dss->ambil_vikor($id_transaksi);
    $data1['kelurahan'] = $this->m_dss->ambil_vikor_kelurahan($id_transaksi);
    $data = $this->session();
    $data['content'] = $this->load->view('dm/metode_borda', $data1, true);
    $this->load->view('layout/template', $data);
}
}

//=====AKHIR
METODE=====

////=====ADMIN=====
=====

//ADMIN
public function lihat_analisa_dss() {
    $data1['dss'] = $this->m_dss->tabel_analisa_dss();
    $data = $this->session();
    $data['content'] = $this->load->view('admin/lihat_analisa_dss', $data1, true);
    $this->load->view('layout/template', $data);
}

public function tambah_analisa_dss() {
    $data1['kecamatan'] = $this->m_lokasi->tabel_kecamatan();
    $data1['user'] = $this->m_dss->ambil_user();
}

```

```

$data = $this->session();
$data['content'] = $this->load->view('admin/tambah_analisa_dss', $data1, true);
$this->load->view('layout/template', $data);
}

public function proses_tambah_analisa_dss() {
    $id_user = $this->input->post('id_user', true);
    $nama_transaksi = $this->input->post('nama_transaksi', true);
    $tgl_mulai = $this->input->post('tgl_mulai', true);
    $tgl_akhir = $this->input->post('tgl_akhir', true);
    $id_kecamatan = $this->input->post('id_kecamatan', true);
    $query = $this->m_dss->tambah_analisa_dss($nama_transaksi, $tgl_mulai, $tgl_akhir, $id_kecamatan);
    if ($query) {
        $id_transaksi = $this->m_dss->ambil_idakhir_detail();
        $id_akhir = $id_transaksi->id_transaksi;
        foreach ($id_user as $row) {
            $query2 = $this->m_dss->tambah_detail_analisa($id_akhir, $row);
        }
        if ($query2) {
            $this->session->set_userdata('operation', "sukses");
            $this->session->set_userdata('message', "Analisa DSS Sukses di Simpan");
            redirect("c_dss/lihat_analisa_dss");
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Analisa DSS Gagal di Simpan");
            redirect("c_dss/lihat_analisa_dss");
        }
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Analisa DSS Gagal di Simpan");
        redirect("c_dss/lihat_analisa_dss");
    }
}

public function ubah_status_analisa($id_transaksi, $status) {
    $query = $this->m_dss->ubah_status_analisa($id_transaksi, $status);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        if ($status == '1') {
            $this->session->set_userdata('message', "Analisa Diaktifkan");
        } else {
            $this->session->set_userdata('message', "Analisa Dinonaktifkan");
        }
        redirect("c_dss/lihat_analisa_dss");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Status Gagal Diubah");
        redirect("c_dss/lihat_analisa_dss");
    }
}

public function ubah_analisa_dss($id_transaksi) {
    $data1['kecamatan'] = $this->m_lokasi->tabel_kecamatan();
    $data1['user'] = $this->m_dss->ambil_user();
    $data1['transaksi'] = $this->m_dss->ambil_analisa_dss($id_transaksi);
    $data1['detail'] = $this->m_dss->ambil_detail_analisa($id_transaksi);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/ubah_analisa_dss', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_ubah_analisa_dss($id_transaksi) {

```

```

$Id_user = $this->input->post('id_user', true);
$nama_transaksi = $this->input->post('nama_transaksi', true);
$ttl_mulai = $this->input->post('ttl_mulai', true);
$ttl_akhir = $this->input->post('ttl_akhir', true);
$Id_kecamatan = $this->input->post('id_kecamatan', true);
$query = $this->m_dss->ubah_analisa_dss($nama_transaksi, $ttl_mulai, $ttl_akhir, $Id_kecamatan,
$Id_transaksi);
if ($query) {
    $query3 = $this->m_dss->hapus_detail_analisa($Id_transaksi);
    if ($query3) {
        foreach ($Id_user as $row) {
            $query2 = $this->m_dss->tambah_detail_analisa($Id_transaksi, $row);
        }
    }
    if ($query2) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Analisa DSS Sukses di Ubah");
        redirect("c_dss/lihat_analisa_dss");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Analisa DSS Gagal di Ubah");
        redirect("c_dss/lihat_analisa_dss");
    }
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Analisa DSS Gagal di Simpan");
    redirect("c_dss/lihat_analisa_dss");
}
}

public function hapus_analisa_dss($Id_transaksi) {
    $query = $this->m_dss->hapus_detail_analisa($Id_transaksi);
    $query2 = $this->m_dss->hapus_analisa_dss($Id_transaksi);
    if ($query && $query2) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Analisa DSS Sukses di Hapus");
        redirect("c_dss/lihat_analisa_dss");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Analisa DSS gagal di Hapus");
        redirect("c_dss/lihat_analisa_dss");
    }
}

public function detail_analisa_dss($Id_transaksi) {
    $data1['dss'] = $this->m_dss->ambil_analisa_dss($Id_transaksi);
    $data1['dm'] = $this->m_dss->ambil_dm_analisa($Id_transaksi);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/detail_analisa_dss', $data1, true);
    $this->load->view('layout/template', $data);
}

function session() {
    $data['nama'] = $this->session->userdata('nama');
    $data['jabatan'] = $this->session->userdata('jabatan');
    $data['pn'] = $this->session->userdata('pn');
    $data['id_user'] = $this->session->userdata('id_user');
    $data['id_hak_akses'] = $this->session->userdata('id_hak_akses');
    $data['jumlah_user'] = $this->m_user->ambil_jumlah_user();
    $data['jumlah_bri'] = $this->m_kriteria->ambil_jumlah_atmbri();
    $data['jumlah_aktif_analisa'] = $this->m_dss->ambil_analisa_aktif();
    $Id = $this->session->userdata('id_user');
}

```



```

        $data['account'] = $this->m_user->ambil_akun($id);
        $data['jumlah_analisa'] = $this->m_dss->ambil_jumlah_analisa($id);
        return $data;
    }
}

```

2. *Class c_user*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_user extends CI_Controller {

    function __construct() {
        parent::__construct();
    }

    public function lihat_user() {
        $data1['tabel_user'] = $this->m_user->tabel_user();
        $data = $this->session();
        $data['content'] = $this->load->view('admin/lihat_user', $data1, true);
        $this->load->view('layout/template', $data);
    }

    public function detail_user($id_user) {
        $data1['data_ambil'] = $this->m_user->ambil_detail_user($id_user);
        // $data1['data_user'] = $this->m_user->ambil_user($id_user);
        $data = $this->session();
        $data['content'] = $this->load->view('admin/detail_user', $data1, true);
        $this->load->view('layout/template', $data);
    }

    public function tambah_user() {
        $data1['option'] = $this->m_user->combo_hak_akses();
        $data = $this->session();
        $data['content'] = $this->load->view('admin/tambah_user', $data1, true);
        $this->load->view('layout/template', $data);
    }

    public function proses_tambah_user() {
        $config['upload_path'] = './temp_upload/';
        $config['allowed_types'] = 'gif|jpg|png';
        $this->load->library('upload', $config);
        if (!$this->upload->do_upload()) {
            $this->session->set_userdata('operation', "validasi");
            $this->session->set_userdata('message', $this->upload->display_errors());
            redirect('c_user/tambah_user');
        } else {
            $upload_data = $this->upload->data();
            $nama = $this->input->post('nama', true);
            $pn = $this->input->post('pn', true);
            $jenis_kelamin = $this->input->post('jenis_kelamin', true);
            $tgl_lahir = $this->input->post('tgl_lahir', true);
            $alamat = $this->input->post('alamat', true);
            $jabatan = $this->input->post('jabatan', true);
            $no_hp = $this->input->post('no_hp', true);
            $sid_hak_akses = $this->input->post('hak_akses', true);
        }
    }
}

```

```

$username = $this->input->post('username', true);
$password = md5($this->input->post('password', true));
$foto = $upload_data['file_name'];
$format = date('Y-m-d', strtotime($tgl_lahir));
$cekpn = $this->m_user->cek_pn();
foreach ($cekpn as $row) {
    $pnadb = $row['pn'];
}
if ($pn==$pnadb) {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Personal Number Duplicate Entry");
    redirect("c_user/tambah_user");
} else {
    $query = $this->m_user->tambah_user($pn, $nama, $jenis_kelamin, $format, $alamat, $jabatan,
    $no_hp, $id_hak_akses, $username, $foto, $password);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data User Sukses di Simpan");
        redirect("c_user/lihat_user");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data User gagal di Simpan");
        redirect("c_user/lihat_user");
    }
}
}
}

public function ubah_user($id_user) {
    $data1['data_ambill'] = $this->m_user->ambil_detail_user($id_user);
    $data1['option'] = $this->m_user->combo_hak_akses();
    $data = $this->session();
    $data['content'] = $this->load->view('admin/ubah_user', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_ubah_user($id_user) {
    $config['upload_path'] = './temp_upload/';
    $config['allowed_types'] = 'gif|jpg|png';
    $this->load->library('upload', $config);
    $upload_data = $this->upload->data();
    $nama = $this->input->post('nama', true);
    $pn = $this->input->post('pn', true);
    $jenis_kelamin = $this->input->post('jenis_kelamin', true);
    $tgl_lahir = $this->input->post('tgl_lahir', true);
    $alamat = $this->input->post('alamat', true);
    $jabatan = $this->input->post('jabatan', true);
    $no_hp = $this->input->post('no_hp', true);
    $id_hak_akses = $this->input->post('hak_akses', true);
    $foto = $upload_data['file_name'];
    $format = date('Y-m-d', strtotime($tgl_lahir));
    $foto_1 = $this->input->post('foto', true);
    $type = $this->session->userdata('id_hak_akses');
    //jika masukan foto tidak kosong
    if ($foto_1 != "") {
        $cekpn = $this->m_user->cek_ubah_pn($id_user);
        foreach ($cekpn as $row) {
            $pnadb = $row['pn'];
        }
        if ($pnadb != $pn) {
            if ($type == '1') {

```



```

        if ($pndb != $pn) {
            if ($type == '1') {
                $query = $this->m_user->ubah_user_foto($id_user, $pn, $nama, $jenis_kelamin, $tgl_lahir,
                $alamat, $jabatan, $no_hp, $id_hak_akses, $foto);
            } else {
                $query = $this->m_user->ubah_user_foto_dm($id_user, $pn, $nama, $jenis_kelamin, $tgl_lahir,
                $alamat, $jabatan, $no_hp, $foto);
            }
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Personal Number Duplicate Entry");
            if ($type == '1') {
                redirect("c_user/ubah_user/$id_user");
            } else {
                redirect("c_user/biodata/$id_user");
            }
        }
    }
}

if ($query) {
    $this->session->set_userdata('operation', "sukses");
    $this->session->set_userdata('message', "Data User Sukses di Ubah");
    if ($type == '1') {
        redirect("c_user/lihat_user");
    } else {
        redirect("c_user/biodata/$id_user");
    }
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Data User gagal di Ubah");
    if ($type == '1') {
        redirect("c_user/lihat_user");
    } else {
        redirect("c_user/biodata/$id_user");
    }
}
}

public function hapus_foto($id_user, $foto) {
    $name_file = "temp_upload/" . $foto;
    unlink($name_file);
    $this->m_user->hapus_foto($id_user);
    $type = $this->session->userdata('id_hak_akses');
    if ($type == '1') {
        redirect('c_user/ubah_user/' . $id_user);
    } else {
        redirect("c_user/biodata/$id_user");
    }
}

public function hapus_user($id_user) {
    $query = $this->m_user->hapus_user($id_user);
    $name_file = "temp_upload/" . $this->m_user->check_foto($id_user);
    unlink($name_file);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Sukses di Hapus");
        redirect("c_user/lihat_user");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data gagal di Hapus");
        redirect("c_user/lihat_user");
    }
}

```

```
}
}

public function biodata($id_user) {
    $data1['data_user'] = $this->m_user->ambil_detail_user($id_user);
    //$data1['data_user'] = $this->m_user->ambil_user($id_user);
    $data1['option'] = $this->m_user->combo_hak_akses();
    $data = $this->session();
    $data['content'] = $this->load->view('dm/lihat_biodata', $data1, true);
    $this->load->view('layout/template', $data);
}

public function ubah_account($id_user, $id_hak_akses) {
    $username = $this->input->post('username', true);
    $password = md5($this->input->post('password', true));
    $query = $this->m_user->ubah_account($id_user, $username, $password);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Account Sukses di Ubah");
        redirect("c_home/home/$id_hak_akses");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Account Gagal di Ubah");
        redirect("c_home/home/$id_hak_akses");
    }
}

function session() {
    $data['nama'] = $this->session->userdata('nama');
    $data['pn'] = $this->session->userdata('pn');
    $data['id_user'] = $this->session->userdata('id_user');
    $data['jabatan'] = $this->session->userdata('jabatan');
    $data['id_hak_akses'] = $this->session->userdata('id_hak_akses');
    $data['jumlah_user'] = $this->m_user->ambil_jumlah_user();
    $data['jumlah_bri'] = $this->m_kriteria->ambil_jumlah_atmbri();
    $data['jumlah_aktif_analisa'] = $this->m_dss->ambil_analisa_aktif();
    $id = $this->session->userdata('id_user');
    $data['account'] = $this->m_user->ambil_akun($id);
    $data['jumlah_analisa'] = $this->m_dss->ambil_jumlah_analisa($id);
    return $data;
}
}
```

3. Class *c_home*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_home extends CI_Controller {

    function __construct() {
        parent::__construct();
    }

    public function home($type) {
        $this->load->library('googlemaps');
        $config['center'] = '-8.239383846027533, 113.6823472357595';
        $config['zoom'] = 10;
        $this->googlemaps->initialize($config);
        $kelurahan = $this->m_lokasi->data_kelurahan();
        $kat = $this->m_kriteria->data_kategori();
        foreach ($kelurahan as $value) {
            $marker2 = array();
            $polygon = str_replace("MULTIPOLYGON", "", $value->lat);
            $polygon = str_replace("(", "", $polygon);
            $polygon = str_replace(")", "", $polygon);
            $polygon = str_replace(" ", "", $polygon);
            $polygon = str_replace(";", "", $polygon);
            $data = explode(";", $polygon);
            $label = $value->label;
            $nama = $value->nama_kelurahan;
            $marker2['fillColor'] = '#87CEEB';
            for ($i = 0; $i < count($data); $i++) {
                $koordinat = explode(" ", $data[$i]);
                $marker2['points'][] = $koordinat[1] . " " . $koordinat[0];
                $marker2['onclick'] = "
                    var infowindow = new google.maps.InfoWindow(
                        {
                            size: new google.maps.Size(150,50)
                        });
                    infowindow.setContent('$label');
                    if (event) {
                        point = event.latLng;
                    }
                    infowindow.setPosition(point);
                    infowindow.open(map);";
            }
            $this->googlemaps->add_polygon($marker2);
        }
        foreach ($kat as $value) {
            $markerSkIh = array();
            $icon = $value->icon;
            $markerSkIh['icon'] = "http://localhost/dss_atm/temp_upload/$icon";
            $markerSkIh['position'] = "$value->lat, $value->long";
            $markerSkIh['infowindow_content'] = "<p>$value->nama</p><p>$value->alamat</p><p>$value-
            >info</p>";
            $this->googlemaps->add_marker($markerSkIh);
        }
        $data1['kecamatan'] = null;
        $data1['kategori'] = null;
        $data1['map'] = $this->googlemaps->create_map();
        $data1['option1'] = $this->m_lokasi->combo_kecamatan();
    }
}

```

```

$data1['kat'] = $this->m_kriteria->jumlah_kategori();
$data1['option2'] = $this->m_kriteria->combo_kategori();
$data = $this->session();
if ($type == '1') {
    $data['content'] = $this->load->view('admin/home_admin', $data1, true);
    $this->load->view('layout/template', $data);
} else {
    $data['content'] = $this->load->view('dm/home_dm', $data1, true);
    $this->load->view('layout/template', $data);
}
}

public function pilih_kategori($kecamatan, $kategori, $type) {
    $this->load->library('googlemaps');
    $config['center'] = '-8.239383846027533, 113.6823472357595';
    $config['zoom'] = 10;
    $this->googlemaps->initialize($config);
    if ($kecamatan == '0' && $kategori == '0') {
        redirect("c_home/home/$type");
    } else if ($kecamatan == '0' && $kategori != '0') {
        $kelurahan = $this->m_lokasi->data_kelurahan();
        $kat = $this->m_kriteria->pilih_kategori($kategori);
    } else if ($kecamatan != '0' && $kategori == '0') {
        $kelurahan = $this->m_lokasi->pilih_kelurahan($kecamatan);
        $kat = $this->m_kriteria->detail_kategori($kecamatan);
    } else {
        $kelurahan = $this->m_lokasi->pilih_kelurahan($kecamatan);
        $kat = $this->m_kriteria->pilih_kategori_detail($kategori, $kecamatan);
    }
    foreach ($kelurahan as $value) {
        $marker2 = array();
        $polygon = str_replace("MULTIPOLYGON", "", $value->lat);
        $polygon = str_replace("(", "", $polygon);
        $polygon = str_replace(")", "", $polygon);
        $polygon = str_replace(", ", ";", $polygon);
        $polygon = str_replace(" ", "", $polygon);
        $data = explode(";", $polygon);
        $label = $value->label;
        $nama = $value->nama_kelurahan;
        $marker2['fillColor'] = '#87CEEB';
        for ($i = 0; $i < count($data); $i++) {
            $koordinat = explode(" ", $data[$i]);
            $marker2['points'][] = $koordinat[1] . " " . $koordinat[0];
            $marker2['onclick'] = "
                var infowindow = new google.maps.InfoWindow(
                    {
                        size: new google.maps.Size(150,50)
                    });
                infowindow.setContent('<p>$nama</p>');
                if (event) {
                    point = event.latLng;
                }
                infowindow.setPosition(point);
                infowindow.open(map);";
        }
        $this->googlemaps->add_polygon($marker2);
    }
    //atm
    foreach ($kat as $value) {
        $markerSkIh = array();
        $icon = $value->icon;
    }
}

```

```

$markerSkIh['icon'] = "http://localhost/dss_atm/temp_upload/$icon";
$markerSkIh['position'] = "$value->lat, $value->long";
$markerSkIh['infowindow_content'] = "<p>$value->nama</p><p>$value->alamat</p><p>$value-
>info</p>";
$this->googlemaps->add_marker($markerSkIh);
}
$data1['kecamatan'] = $kecamatan;
$data1['kategori'] = $kategori;
$data1['map'] = $this->googlemaps->create_map();
$data1['option1'] = $this->m_lokasi->combo_kecamatan();
$data1['kat'] = $this->m_kriteria->jumlah_kategori();
$data1['option2'] = $this->m_kriteria->combo_kategori();
$data = $this->session();
if ($type == '1') {
    $data['content'] = $this->load->view('admin/home_admin', $data1, true);
    $this->load->view('layout/template', $data);
} else {
    $data['content'] = $this->load->view('dm/home_dm', $data1, true);
    $this->load->view('layout/template', $data);
}
}

public function home_user() {
    $this->load->library('googlemaps');
    $config['center'] = '-8.239383846027533, 113.6823472357595';
    $config['zoom'] = 10;
    $this->googlemaps->initialize($config);
    $kelurahan = $this->m_lokasi->data_kelurahan();
    $kat = $this->m_kriteria->data_kategori();
    foreach ($kelurahan as $value) {
        $marker2 = array();
        $polygon = str_replace("MULTIPOLYGON", "", $value->lat);
        $polygon = str_replace("(", "", $polygon);
        $polygon = str_replace(")", "", $polygon);
        $polygon = str_replace(",", ";", $polygon);
        $polygon = str_replace(" ", "", $polygon);
        $data = explode(";", $polygon);
        $label = $value->label;
        $nama = $value->nama_kelurahan;
        $marker2['fillColor'] = '#87CEEB';
        for ($i = 0; $i < count($data); $i++) {
            $koordinat = explode(" ", $data[$i]);
            $marker2['points'][] = $koordinat[1] . " " . $koordinat[0];
            $marker2['onclick'] = "
                var infowindow = new google.maps.InfoWindow(
                    {
                        size: new google.maps.Size(150,50)
                    });
                infowindow.setContent('$label');
                if (event) {
                    point = event.latLng;
                }
                infowindow.setPosition(point);
                infowindow.open(map);";
        }
        $this->googlemaps->add_polygon($marker2);
    }
    foreach ($kat as $value) {
        $markerSkIh = array();
        $icon = $value->icon;
        $markerSkIh['icon'] = "http://localhost/dss_atm/temp_upload/$icon";
    }
}

```



```

    $markerSkh['position'] = "$value->lat, $value->long";
    $markerSkh['infowindow_content'] = "<p>$value->nama</p><p>$value->alamat</p><p>$value->info</p>";
    $this->googlemaps->add_marker($markerSkh);
  }
  $data1['kecamatan'] = null;
  $data1['kategori'] = null;
  $data1['map'] = $this->googlemaps->create_map();
  $data1['option1'] = $this->m_lokasi->combo_kecamatan();
  $data1['kat'] = $this->m_kriteria->jumlah_kategori();
  $data1['option2'] = $this->m_kriteria->combo_kategori();
  $data['content'] = $this->load->view('user/home_user', $data1, true);
  $this->load->view('layout/template_user', $data);
}

public function pilih_kategori_user($kecamatan, $kategori) {
  $this->load->library('googlemaps');
  $config['center'] = '-8.239383846027533, 113.6823472357595';
  $config['zoom'] = 10;
  $this->googlemaps->initialize($config);
  if ($kecamatan == '0' && $kategori == '0') {
    redirect('c_home/home_admin');
  } else if ($kecamatan == '0' && $kategori != '0') {
    $kelurahan = $this->m_lokasi->data_kelurahan();
    $kat = $this->m_kriteria->pilih_kategori($kategori);
  } else if ($kecamatan != '0' && $kategori == '0') {
    $kelurahan = $this->m_lokasi->pilih_kelurahan($kecamatan);
    $kat = $this->m_kriteria->detail_kategori($kecamatan);
  } else {
    $kelurahan = $this->m_lokasi->pilih_kelurahan($kecamatan);
    $kat = $this->m_kriteria->pilih_kategori_detail($kategori, $kecamatan);
  }
  foreach ($kelurahan as $value) {
    $marker2 = array();
    $polygon = str_replace("MULTIPOLYGON", "", $value->lat);
    $polygon = str_replace("(", "", $polygon);
    $polygon = str_replace(")", "", $polygon);
    $polygon = str_replace(";", "", $polygon);
    $polygon = str_replace(" ", "", $polygon);
    $polygon = str_replace(" ", "", $polygon);
    $data = explode(";", $polygon);
    $label = $value->label;
    $nama = $value->nama_kelurahan;
    $marker2['fillColor'] = '#87CEEB';
    for ($i = 0; $i < count($data); $i++) {
      $koordinat = explode(";", $data[$i]);
      $marker2['points'][] = $koordinat[1] . ", " . $koordinat[0];
      $marker2['onclick'] = "
        var infowindow = new google.maps.InfoWindow(
          {
            size: new google.maps.Size(150,50)
          });
      infowindow.setContent('<p>$nama</p>');
      if (event) {
        point = event.latLng;
      }
      infowindow.setPosition(point);
      infowindow.open(map);";
    }
    $this->googlemaps->add_polygon($marker2);
  }
}
//atm

```

```

foreach ($kat as $value) {
    $markerSkIh = array();
    $icon = $value->icon;
    $markerSkIh['icon'] = "http://localhost/dss_atm/temp_upload/$icon";
    $markerSkIh['position'] = "$value->lat, $value->long";
    $markerSkIh['infowindow_content'] = "<p>$value->nama</p><p>$value->alamat</p><p>$value-
>info</p>";
    $this->googlemaps->add_marker($markerSkIh);
}
$data1['kecamatan'] = $kecamatan;
$data1['kategori'] = $kategori;
$data1['map'] = $this->googlemaps->create_map();
$data1['option1'] = $this->m_lokasi->combo_kecamatan();
$data1['kat'] = $this->m_kriteria->jumlah_kategori();
$data1['option2'] = $this->m_kriteria->combo_kategori();
$data['content'] = $this->load->view('user/home_user', $data1, true);
$this->load->view('layout/template_user', $data);
}

function session() {
    $data['nama'] = $this->session->userdata('nama');
    $data['id_user'] = $this->session->userdata('id_user');
    $data['pn'] = $this->session->userdata('pn');
    $data['jabatan'] = $this->session->userdata('jabatan');
    $data['id_hak_akses'] = $this->session->userdata('id_hak_akses');
    $data['jumlah_user'] = $this->m_user->ambil_jumlah_user();
    $data['jumlah_bri'] = $this->m_kriteria->ambil_jumlah_atmbri();
    $data['jumlah_aktif_analisa'] = $this->m_dss->ambil_analisa_aktif();
    $id = $this->session->userdata('id_user');
    $data['account'] = $this->m_user->ambil_akun($id);
    $data['jumlah_analisa'] = $this->m_dss->ambil_jumlah_analisa($id);
    return $data;
}
}
}

```

4. *Class c_kriteria*

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_kriteria extends CI_Controller {

    function __construct() {
        parent::__construct();
    }

    //ADMIN
    public function lihat_kriteria() {
        $data1['tabel_kriteria'] = $this->m_kriteria->tabel_kriteria();
        $data = $this->session();
        $data['content'] = $this->load->view('admin/lihat_kriteria', $data1, true);
        $this->load->view('layout/template', $data);
    }

    public function tambah_kriteria() {
        $data1['option'] = $this->m_kriteria->combo_sifat_kriteria();
    }
}

```

```

$data1['jenis'] = $this->m_kriteria->combo_jenis_kriteria();
$data1['bentuk'] = $this->m_kriteria->combo_bentuk_kriteria();
$data = $this->session();
$data['content'] = $this->load->view('admin/tambah_kriteria', $data1, true);
$this->load->view('layout/template', $data);
}

public function proses_tambah_kriteria() {
    $nama_kriteria = $this->input->post('nama', true);
    $sid_sifat_kriteria = $this->input->post('sifat_kriteria', true);
    $sid_bentuk_kriteria = $this->input->post('bentuk_kriteria', true);
    if($sid_bentuk_kriteria=='1'){
        $sid_jenis_kriteria = $this->input->post('jenis_kriteria', true);
    }
    else{
        $sid_jenis_kriteria = $this->input->post('jenis_kriteria2', true);
    }
    $ket = $this->input->post('ket', true);
    $query = $this->m_kriteria->tambah_kriteria($nama_kriteria, $sid_sifat_kriteria, $ket, $sid_jenis_kriteria,
    $sid_bentuk_kriteria);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Kriteria Sukses di Simpan, Isi Range Kriteria");
        redirect("c_kriteria/lihat_kriteria");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Kriteria Gagal di Simpan");
        redirect("c_kriteria/lihat_kriteria");
    }
}

public function ubah_kriteria($sid_kriteria) {
    $data1['option'] = $this->m_kriteria->combo_sifat_kriteria();
    $data1['jenis'] = $this->m_kriteria->combo_jenis_kriteria();
    $data1['bentuk'] = $this->m_kriteria->combo_bentuk_kriteria();
    $data1['data_kriteria'] = $this->m_kriteria->ambil_kriteria($sid_kriteria);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/ubah_kriteria', $data1, true);
    $this->load->view('layout/template', $data);
}

public function hapus_kriteria($sid_kriteria) {
    $query = $this->m_kriteria->hapus_kriteria($sid_kriteria);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Kriteria Sukses di Hapus");
        redirect("c_kriteria/lihat_kriteria");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Kriteria gagal di Hapus");
        redirect("c_kriteria/lihat_kriteria");
    }
}

public function proses_ubah_kriteria($sid_kriteria) {
    $nama_kriteria = $this->input->post('nama_kriteria', true);
    $sid_sifat_kriteria = $this->input->post('sifat_kriteria', true);
    $sid_bentuk_kriteria = $this->input->post('bentuk_kriteria', true);
    if($sid_bentuk_kriteria=='1'){
        $sid_jenis_kriteria = $this->input->post('jenis_kriteria1', true);
    }
    else{

```

```

    $id_jenis_kriteria = $this->input->post('jenis_kriteria', true);
    }
    $keterangan = $this->input->post('ket', true);
    $query = $this->m_kriteria->ubah_kriteria($nama_kriteria, $id_sifat_kriteria, $keterangan, $id_kriteria,
    $id_jenis_kriteria, $id_bentuk_kriteria);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Kriteria Sukses di Ubah");
        redirect("c_kriteria/lihat_kriteria");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Kriteria gagal di Ubah");
        redirect("c_kriteria/lihat_kriteria");
    }
}

public function lihat_range_kriteria($id_kriteria) {
    $data1['range_kriteria'] = $this->m_kriteria->range_kriteria($id_kriteria);
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/lihat_range_kriteria', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_tambah_range($id_kriteria) {
    $no = $this->input->post('no', true);
    for ($a = 1; $a <= $no; $a++) {
        $batas_atas = $this->input->post('batas_atas' . $a, true);
        $batas_bawah = $this->input->post('batas_bawah' . $a, true);
        $range = $this->input->post('range' . $a, true);
        $nk = $this->input->post('nk' . $a, true);
        $query = $this->m_kriteria->tambah_range_kriteria($batas_atas, $batas_bawah, $nk, $id_kriteria,
    $range);
    }

    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Range Kriteria Sukses di Simpan");
        redirect("c_kriteria/lihat_range_kriteria/$id_kriteria");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Range Kriteria Gagal di Simpan");
        redirect("c_kriteria/lihat_range_kriteria/$id_kriteria");
    }
}

public function ubah_range_kriteria($id_range, $id_kriteria) {
    $data1['data_ambil'] = $this->m_kriteria->ambil_range($id_kriteria,$id_range);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/ubah_range', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_ubah_range($id_range, $id_kriteria) {
    $batas_atas = $this->input->post('batas_atas', true);
    $batas_bawah = $this->input->post('batas_bawah', true);
    $range = $this->input->post('range', true);
    $nk = $this->input->post('nk', true);
    $query = $this->m_kriteria->ubah_range_kriteria($batas_atas, $batas_bawah, $nk, $id_range, $id_kriteria,
    $range);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
    }
}

```

```

    $this->session->set_userdata('message', "Data Range Kriteria Sukses di Ubah");
    redirect("c_kriteria/lihat_range_kriteria/$id_kriteria");
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Data Range Kriteria gagal di Ubah");
    redirect("c_kriteria/lihat_range_kriteria/$id_kriteria");
}
}

public function lihat_nilai_kriteria($id_kriteria, $id_bentuk_kriteria) {
    $data1['kelurahan'] = $this->m_lokasi->tabel_kelurahan();
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
    //buat lihat kategori
    $data1['tabel_kategori'] = $this->m_kriteria->tabel_kategori($id_kriteria);
    $data1['nilai'] = $this->m_kriteria->tabel_nilai_kriteria($id_kriteria);
    $data = $this->session();
    if ($id_bentuk_kriteria == '1') {
        $data['content'] = $this->load->view('admin/lihat_kategori', $data1, true);
    } else {
        $data['content'] = $this->load->view('admin/lihat_nilai_kriteria', $data1, true);
    }
    $this->load->view('layout/template', $data);
}

public function proses_tambah_nilai_semua($id_kriteria, $id_bentuk_kriteria, $id_jenis_kriteria) {
    $no = $this->input->post('no', true);
    if ($id_jenis_kriteria == '2') {
        for ($a = 1; $a < $no; $a++) {
            $id_kelurahan = $this->input->post('id_kelurahan' . $a, true);
            $nilai = $this->input->post('nilai' . $a, true);
            $query = $this->m_kriteria->tambah_nilai_kriteria($id_kelurahan, $id_kriteria, $nilai);
        }
        if ($query) {
            $this->session->set_userdata('operation', "sukses");
            $this->session->set_userdata('message', "Nilai Kriteria Sukses di Simpan");
            redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Nilai Kriteria Gagal di Simpan");
            redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
        }
    } else {
        for ($a = 1; $a < $no; $a++) {
            $id_kelurahan = $this->input->post('id_kelurahan' . $a, true);
            $nilaiasli = $this->input->post('nilaiasli' . $a, true);
            $query = $this->m_kriteria->tambah_nilai_kriteriaasli($id_kelurahan, $id_kriteria, $nilaiasli);
        }
        if ($query) {
            $nilai = $this->m_kriteria->ambil_nilai_kepentingansemua($id_kriteria);
            foreach ($nilai as $row) {
                $simpan = $row->nilai_kepentingan;
                $id_kelurahan = $row->id_kelurahan;
                $id_kriteria = $row->id_kriteria;
                $query2 = $this->m_kriteria->ubah_nilai_asli($id_kelurahan, $id_kriteria, $simpan);
            }
            if ($query2) {
                $this->session->set_userdata('operation', "sukses");
                $this->session->set_userdata('message', "Nilai Kriteria Sukses di Simpan");
                redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
            } else {
                $this->session->set_userdata('operation', "gagal");
                $this->session->set_userdata('message', "Nilai Kriteria Gagal di Simpan");
            }
        }
    }
}

```

```

        redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
    }
}
}

public function tambah_nilai_kriteria($id_kriteria) {
    $data1['kelurahan'] = $this->m_lokasi->tabel_kelurahan();
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
    $data1['deskriptif'] = $this->m_kriteria->tabel_nilai_kriteria($id_kriteria);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/tambah_nilai_kriteria', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_tambah_nilai($id_kriteria, $id_bentuk_kriteria, $id_jenis_kriteria) {
    if ($id_jenis_kriteria == '2') {
        $id_kelurahan = $this->input->post('id_kelurahan', true);
        $nilai = $this->input->post('nilai', true);
        $query = $this->m_kriteria->tambah_nilai_kriteria($id_kelurahan, $id_kriteria, $nilai);
        if ($query) {
            $this->session->set_userdata('operation', "sukses");
            $this->session->set_userdata('message', "Nilai Kriteria Sukses di Simpan");
            redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Nilai Kriteria Gagal di Simpan");
            redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
        }
    } else {
        $id_kelurahan = $this->input->post('id_kelurahan', true);
        $nilaiasli = $this->input->post('nilaiasli', true);
        $query = $this->m_kriteria->tambah_nilai_kriteriaasli($id_kelurahan, $id_kriteria, $nilaiasli);
        if ($query) {
            $nilai = $this->m_kriteria->ambil_nilai_kepentingansemua($id_kriteria);
            foreach ($nilai as $row) {
                $simpan = $row->nilai_kepentingan;
                $id_kelurahan = $row->id_kelurahan;
                $id_kriteria = $row->id_kriteria;
                $query2 = $this->m_kriteria->ubah_nilai_asli($id_kelurahan, $id_kriteria, $simpan);
            }
            if ($query2) {
                $this->session->set_userdata('operation', "sukses");
                $this->session->set_userdata('message', "Nilai Kriteria Sukses di Simpan");
                redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
            } else {
                $this->session->set_userdata('operation', "gagal");
                $this->session->set_userdata('message', "Nilai Kriteria Gagal di Simpan");
                redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
            }
        }
    }
}

public function ubah_nilai_kriteria($id_nilai, $id_kriteria) {
    $data1['kelurahan'] = $this->m_lokasi->tabel_kelurahan();
    $data1['nilai'] = $this->m_kriteria->ambil_range_deskriptif($id_nilai, $id_kriteria);
    $data1['combo'] = $this->m_kriteria->ambil_deskriptif($id_kriteria);
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
    $data1['nilainum'] = $this->m_kriteria->ambil_nilai_kriteria($id_nilai);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/ubah_nilai_kriteria', $data1, true);
}

```

```

$this->load->view('layout/template', $data);
}

public function proses_ubah_nilai($id_nilai, $id_kriteria, $id_bentuk_kriteria, $id_jenis_kriteria) {
    if ($id_jenis_kriteria == '2') {
        $id_kelurahan = $this->input->post('id_kelurahan', true);
        $nilai = $this->input->post('nilai', true);
        $query = $this->m_kriteria->ubah_nilai_kriteria($id_kelurahan, $nilai, $id_nilai);
        if ($query) {
            $this->session->set_userdata('operation', "sukses");
            $this->session->set_userdata('message', "Nilai Kriteria Sukses di Ubah");
            redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Nilai Kriteria gagal di Ubah");
            redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
        }
    } else {
        $id_kelurahannum = $this->input->post('id_kelurahannum', true);
        $nilainum = $this->input->post('nilainum', true);
        $query = $this->m_kriteria->ubah_nilaiasli_kriteria($id_kelurahannum, $nilainum, $id_nilai);
        if ($query) {
            $nilai = $this->m_kriteria->ambil_nilai_kepentingan($id_kriteria, $id_nilai, $id_kelurahannum);
            foreach ($nilai as $row) {
                $nilaikep = $row->nilai_kepentingan;
                $id_kel = $row->id_kelurahan;
                $id_krit = $row->id_kriteria;
                $id_nil = $row->id_nilai;
                $query2 = $this->m_kriteria->ubah_nilai_aslinum($id_kel, $id_krit, $nilaikep, $id_nil);
            }
            if ($query2) {
                $this->session->set_userdata('operation', "sukses");
                $this->session->set_userdata('message', "Nilai Kriteria Sukses di Simpan");
                redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
            } else {
                $this->session->set_userdata('operation', "gagal");
                $this->session->set_userdata('message', "Nilai Kriteria Gagal di Simpan");
                redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
            }
        }
    }
}

public function hapus_nilai_kriteria($id_nilai, $id_kriteria, $id_bentuk_kriteria) {
    $query = $this->m_kriteria->hapus_nilai_kriteria($id_nilai);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Nilai Kriteria Sukses di Hapus");
        redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Nilai Kriteria gagal di Hapus");
        redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
    }
}

public function tambah_kategori($id_kriteria) {
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/tambah_kategori', $data1, true);
    $this->load->view('layout/template', $data);
}

```

```

public function proses_tambah_kategori() {
    $config['upload_path'] = './temp_upload/';
    $config['allowed_types'] = 'gif|jpg|png';
    $this->load->library('upload', $config);
    if (!$this->upload->do_upload()) {
        $this->session->set_userdata('operation', "validasi");
        $this->session->set_userdata('message', $this->upload->display_errors());
        redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
    } else {
        $upload_data = $this->upload->data();
        $nama = $this->input->post('nama', true);
        $id_kriteria = $this->input->post('id_kriteria', true);
        $id_bentuk_kriteria = $this->input->post('id_bentuk_kriteria', true);
        $icon = $upload_data['file_name'];
        $keterangan = $this->input->post('keterangan', true);
        $query = $this->m_kriteria->tambah_kategori($nama, $id_kriteria, $keterangan, $icon);
        if ($query) {
            $this->session->set_userdata('operation', "sukses");
            $this->session->set_userdata('message', "Data Kategori Sukses di Simpan");
            redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Data Kategori Gagal di Simpan");
            redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
        }
    }
}

public function ubah_kategori($id_kategori, $id_kriteria) {
    $data1['data_kategori'] = $this->m_kriteria->ambil_kategori($id_kategori);
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/ubah_kategori', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_ubah_kategori($id) {
    $config['upload_path'] = './temp_upload/';
    $config['allowed_types'] = 'gif|jpg|png';
    $this->load->library('upload', $config);
    $upload_data = $this->upload->data();
    $nama = $this->input->post('nama', true);
    $foto = $upload_data['file_name'];
    $foto_1 = $this->input->post('foto', true);
    $id_kriteria = $this->input->post('id_kriteria', true);
    $id_bentuk_kriteria = $this->input->post('id_bentuk_kriteria', true);
    $keterangan = $this->input->post('keterangan', true);
    if ($foto_1 != "") {
        $query = $this->m_kriteria->ubah_kategori_icon($nama, $keterangan, $id, $foto_1);
    } else if ($this->m_kriteria->check_icon($id) == "") {
        $query = $this->m_kriteria->ubah_kategori_tanpa_icon($nama, $keterangan, $id);
    } else {
        $config['upload_path'] = './temp_upload/';
        $config['allowed_types'] = 'gif|jpg|png';
        $this->load->library('upload', $config);
        if (!$this->upload->do_upload()) {
            $this->session->set_userdata('operation', "validasi");
            $this->session->set_userdata('message', $this->upload->display_errors());
            redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
        } else {
            $upload_data = $this->upload->data();

```



```

        $foto = $upload_data['file_name'];
        $query = $this->m_kriteria->ubah_kategori_icon($nama, $keterangan, $id, $foto);
    }
}
if ($query) {
    $this->session->set_userdata('operation', "sukses");
    $this->session->set_userdata('message', "Data Kategori Sukses di Ubah");
    redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Data Kategori gagal di Ubah");
    redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
}
}

public function hapus_icon($id_kategori, $foto, $id_kriteria) {
    $name_file = "temp_upload/" . $foto;
    unlink($name_file);
    $this->m_kriteria->hapus_icon($id_kategori);
    redirect("c_kriteria/ubah_kategori/$id_kategori/$id_kriteria");
}

public function hapus_kategori($id_kategori, $id_kriteria, $id_bentuk_kriteria) {
    $query = $this->m_kriteria->hapus_kategori($id_kategori);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Kategori Sukses di Hapus");
        redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Kategori gagal di Hapus");
        redirect("c_kriteria/lihat_nilai_kriteria/$id_kriteria/$id_bentuk_kriteria");
    }
}

public function lihat_nilai_kriteria_peta($id_kategori, $id_kriteria) {
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
    $data1['kategori'] = $this->m_kriteria->ambil_kategori($id_kategori);
    $data1['nilai'] = $this->m_kriteria->tabel_nilai_kriteria_peta($id_kategori);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/lihat_nilai_kriteria_peta', $data1, true);
    $this->load->view('layout/template', $data);
}

public function tambah_nilai_kriteria_peta($id_kriteria, $id_kategori) {
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
    $data1['kategori'] = $this->m_kriteria->ambil_kategori($id_kategori);
    $data1['option'] = $this->m_lokasi->combo_kelurahan();
    $data = $this->session();
    $data['content'] = $this->load->view('admin/tambah_nilai_kriteria_peta', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_tambah_nilai_kriteria_peta($id_kategori, $id_kriteria) {
    $nama = $this->input->post('nama', true);
    $alamat = $this->input->post('alamat', true);
    $id_kelurahan = $this->input->post('id_kelurahan', true);
    $info = $this->input->post('info', true);
    $geom = $this->input->post('geom', true);
    $query = $this->m_kriteria->tambah_nilai_kriteria_peta($id_kriteria, $nama, $alamat, $id_kelurahan,
    $info, $id_kategori, $geom);
    if ($query) {

```

```

$ceknilai = $this->m_kriteria->cek_nilai($id_kriteria, $id_kelurahan);
foreach ($ceknilai as $row) {
    $a = $row['nilai_asli'];
    $b = $row['id_nilai'];
}
if ($a == 0 && $b == 0) {
    $niliasli = 1;
    $query2 = $this->m_kriteria->tambah_nilai_kriteriaasli($id_kelurahan, $id_kriteria, $niliasli);
    if ($query2) {
        $nilai = $this->m_kriteria->ambil_nilai_kepentingan_kategori($id_kriteria, $id_kelurahan);
        foreach ($nilai as $row) {
            $simpan = $row['nilai_kepentingan'];
            $id_nilai1 = $row['id_nilai'];
            $query3 = $this->m_kriteria->ubah_nilai_asli_peta($id_nilai1, $simpan);
        }
    }
} if ($a != 0 && $b != 0) {
    $niliasli = $a + 1;
    $query2 = $this->m_kriteria->ubah_nilai_kategori($b, $niliasli);
    if ($query2) {
        $nilai1 = $this->m_kriteria->ambil_nilai_kepentingan_kategori($id_kriteria, $id_kelurahan);
        foreach ($nilai1 as $row) {
            $simpan2 = $row['nilai_kepentingan'];
            $id_nilai2 = $row['id_nilai'];
            $query3 = $this->m_kriteria->ubah_nilai_asli_peta($id_nilai2, $simpan2);
        }
    }
}
}
}
if ($query3) {
    $this->session->set_userdata('operation', "sukses");
    $this->session->set_userdata('message', "Nilai Kriteria Sukses di Simpan");
    redirect("c_kriteria/lihat_nilai_kriteria_peta/$id_kategori/$id_kriteria");
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Nilai Kriteria Gagal di Simpan");
    redirect("c_kriteria/lihat_nilai_kriteria_peta/$id_kategori/$id_kriteria");
}
}

public function hapus_nilai_kriteria_peta($id_nilai, $id_kriteria, $id_kategori, $id_kelurahan) {
    $ceknilai = $this->m_kriteria->cek_nilai($id_kriteria, $id_kelurahan);
    foreach ($ceknilai as $row) {
        $a = $row['nilai_asli'];
        $b = $row['id_nilai'];
    }
    if($a==1){
        $query = $this->m_kriteria->hapus_nilai_kriteria($b);
    }
    if($a>1){
        $niliasli = $a - 1;
        $query2 = $this->m_kriteria->ubah_nilai_kategori($b, $niliasli);
        if ($query2) {
            $nilai1 = $this->m_kriteria->ambil_nilai_kepentingan_kategori($id_kriteria, $id_kelurahan);
            foreach ($nilai1 as $row) {
                $simpan2 = $row['nilai_kepentingan'];
                $id_nilai2 = $row['id_nilai'];
                $query = $this->m_kriteria->ubah_nilai_asli_peta($id_nilai2, $simpan2);
            }
        }
    }
}
}
if ($query) {

```

```

$query3 = $this->m_kriteria->hapus_nilai_kriteria($id_nilai);
}
if ($query3) {
    $this->session->set_userdata('operation', "sukses");
    $this->session->set_userdata('message', "Nilai Kriteria Sukses di Hapus");
    redirect("c_kriteria/lihat_nilai_kriteria_peta/$id_kategori/$id_kriteria");
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Nilai Kriteria gagal di Hapus");
    redirect("c_kriteria/lihat_nilai_kriteria_peta/$id_kategori/$id_kriteria");
}
}

public function ubah_nilai_kriteria_peta($id_nilai, $id_kriteria, $id_kategori) {
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
    $data1['kategori'] = $this->m_kriteria->ambil_kategori($id_kategori);
    $data1['nilai'] = $this->m_kriteria->ambil_nilai_kriteria($id_nilai);
    $data1['option'] = $this->m_lokasi->combo_kelurahan();
    $data = $this->session();
    $data['content'] = $this->load->view('admin/ubah_nilai_kriteria_peta', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_ubah_nilai_kriteria_peta($id_nilai, $id_kategori, $id_kriteria) {
    $nama = $this->input->post('nama', true);
    $alamat = $this->input->post('alamat', true);
    $id_kelurahan = $this->input->post('id_kelurahan', true);
    $info = $this->input->post('info', true);
    $geom = $this->input->post('geom', true);
    if ($geom == "") {
        $query = $this->m_kriteria->ubah_nilai_kriteria_peta($nama, $alamat, $id_kelurahan, $info, $id_nilai);
    } else {
        $query = $this->m_kriteria->ubah_nilai_kriteria_petageom($nama, $alamat, $id_kelurahan, $info,
$id_nilai, $geom);
    }
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Nilai Kriteria Sukses di Ubah");
        redirect("c_kriteria/lihat_nilai_kriteria_peta/$id_kategori/$id_kriteria");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Nilai Kriteria gagal di Ubah");
        redirect("c_kriteria/lihat_nilai_kriteria_peta/$id_kategori/$id_kriteria");
    }
}

public function detail_nilai_kriteria_peta($id_nilai, $id_kriteria, $id_kategori) {
    $nilai = $this->m_kriteria->tabel_nilai_kategori($id_nilai);
    $this->load->library('googlemaps');
    $config['center'] = '-8.239383846027533, 113.6823472357595';
    $config['zoom'] = 10;
    $this->googlemaps->initialize($config);
    foreach ($nilai as $value) {
        $markerSkjh = array();
        $icon = $value->icon;
        $markerSkjh['icon'] = "http://localhost/dss_atm/temp_upload/$icon";
        $markerSkjh['position'] = "$value->lat, $value->long";
        $markerSkjh['infowindow_content'] = "<p>$value->nama</p><p>$value->alamat</p><p>$value-
>info</p>";
        $this->googlemaps->add_marker($markerSkjh);
    }
    $data = $this->session();
}

```

```

$data1['data_ambil'] = $this->m_kriteria->ambil_kriteria($id_kriteria);
$data1['kategori'] = $this->m_kriteria->ambil_kategori($id_kategori);
$data1['nilai'] = $this->m_kriteria->ambil_nilai_kriteria($id_nilai);
$data1['map'] = $this->googlemaps->create_map();
$data['content'] = $this->load->view('admin/detail_nilai_kriteria_peta', $data1, true);
$this->load->view('layout/template', $data);
}

public function lihat_atm() {
    $data1['data_ambil'] = $this->m_kriteria->ambil_kriteria(10);
    $data1['nilai'] = $this->m_kriteria->tabel_nilai_kriteria_peta(1);
    $data['content'] = $this->load->view('user/lihat_atm', $data1, true);
    $this->load->view('layout/template_user', $data);
}

public function detail_atm($id_nilai) {
    $nilai = $this->m_kriteria->tabel_nilai_kategori($id_nilai);
    $this->load->library('googlemaps');
    $config['center'] = '-8.239383846027533, 113.6823472357595';
    $config['zoom'] = 10;
    $this->googlemaps->initialize($config);
    foreach ($nilai as $value) {
        $markerSkh = array();
        $icon = $value->icon;
        $markerSkh['icon'] = "http://localhost/dss_atm/temp_upload/$icon";
        $markerSkh['position'] = "$value->lat, $value->long";
        $markerSkh['infowindow_content'] = "<p>$value->nama</p><p>$value->alamat</p><p>$value->info</p>";
        $this->googlemaps->add_marker($markerSkh);
    }
    $data1['nilai'] = $this->m_kriteria->ambil_nilai_kriteria($id_nilai);
    $data1['map'] = $this->googlemaps->create_map();
    $data['content'] = $this->load->view('user/detail_atm', $data1, true);
    $this->load->view('layout/template_user', $data);
}

function session() {
    $data['nama'] = $this->session->userdata('nama');
    $data['jabatan'] = $this->session->userdata('jabatan');
    $data['pn'] = $this->session->userdata('pn');
    $data['id_user'] = $this->session->userdata('id_user');
    $data['id_hak_akses'] = $this->session->userdata('id_hak_akses');
    $data['jumlah_user'] = $this->m_user->ambil_jumlah_user();
    $data['jumlah_bri'] = $this->m_kriteria->ambil_jumlah_atmbri();
    $data['jumlah_aktif_analisa'] = $this->m_dss->ambil_analisa_aktif();
    $id = $this->session->userdata('id_user');
    $data['account'] = $this->m_user->ambil_akun($id);
    $data['jumlah_analisa'] = $this->m_dss->ambil_jumlah_analisa($id);
    return $data;
}
}

```

5. *Class c_login*

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_login extends CI_Controller {

    function __construct() {
        parent::__construct();
    }

    public function index() {
        $type = $this->session->userdata('id_hak_akses');
        $login = $this->session->userdata('login');
        if ($login != "") {
            if ($type == '1' || $type == '2') {
                redirect("c_home/home/$type");
            } else {
                redirect('c_login/logout');
            }
        }
        $this->load->view('login');
    }

    public function proses() {
        $this->form_validation->set_rules('username', 'username', 'required');
        $this->form_validation->set_rules('password', 'password', 'required');
        if ($this->form_validation->run()) {
            $username = $this->input->post('username');
            $password = md5($this->input->post('password'));
            $query = $this->m_login->login($username, $password);
            if ($query == true) {
                redirect(base_url());
            } else {
                $this->session->set_userdata('operation', "gagal");
                $this->session->set_userdata('message', "Username dan Password tidak ditemukan");
                redirect(base_url());
            }
        } else {
            $this->session->set_userdata('operation', "validasi");
            $this->session->set_userdata('message', validation_errors());
            redirect(base_url());
        }
    }

    public function logout() {
        $this->session->sess_destroy();
        $this->session->unset_userdata('username');
        $this->session->unset_userdata('login');
        $this->session->unset_userdata('type');
        $this->session->unset_userdata('status');
        redirect(base_url());
    }
}
```

6. *Class c_lokasi*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class C_lokasi extends CI_Controller {

    function __construct() {
        parent::__construct();
    }

    public function lihat_kecamatan() {
        $data1['tabel_kecamatan'] = $this->m_lokasi->tabel_kecamatan();
        $data = $this->session();
        $data['content'] = $this->load->view('admin/lihat_kecamatan', $data1, true);
        $this->load->view('layout/template', $data);
    }

    public function detail_kecamatan($id) {
        $kecamatan = $this->m_lokasi->data_kecamatan();
        $kecamatan_detail = $this->m_lokasi->ambil_kecamatan($id);
        $this->load->library('googlemaps');
        $config['center'] = '-8.239383846027533, 113.6823472357595';
        $config['zoom'] = 10;
        $this->googlemaps->initialize($config);
        foreach ($kecamatan as $value) {
            $marker = array();
            $polygon1 = str_replace("MULTIPOLYGON", "", $value->lat);
            $polygon2 = str_replace("(", "", $polygon1);
            $polygon3 = str_replace(")", "", $polygon2);
            $polygon4 = str_replace(",", ";", $polygon3);
            $polygon5 = str_replace(" ", "", $polygon4);
            $data = explode(";", $polygon5);
            $label = $value->label;
            $nama = $value->nama_kecamatan;
            $label2 = $kecamatan_detail->label;
            if ($label == $label2) {
                $marker['fillColor'] = '#0F0';
            } else {
                $marker['fillColor'] = '#87CEEB';
            }
            for ($i = 0; $i < count($data); $i++) {
                $koordinat = explode(" ", $data[$i]);
                $marker['points'][] = $koordinat[1] . ", " . $koordinat[0];
                $marker['onclick'] = "
                    var infowindow = new google.maps.InfoWindow(

                        {
                            size: new google.maps.Size(150,50)
                        });
                    infowindow.setContent('$label');
                    if (event) {
                        point = event.latLng;
                    }
                    infowindow.setPosition(point);
                    infowindow.open(map);";
            }
            $this->googlemaps->add_polygon($marker);
        }
        $data = $this->session();
    }
}

```

```

$data1['data_kecamatan'] = $this->m_lokasi->ambil_kecamatan($id);
$data1['map'] = $this->googlemaps->create_map();
$data['content'] = $this->load->view('admin/detail_kecamatan', $data1, true);
$this->load->view('layout/template', $data);
}

public function tambah_kecamatan() {
    $data = $this->session();
    $data['content'] = $this->load->view('admin/tambah_kecamatan', "", true);
    $this->load->view('layout/template', $data);
}

public function proses_tambah_kecamatan() {
    $nama_kecamatan = $this->input->post('nama_kecamatan', true);
    $kode_pos = $this->input->post('kode_pos', true);
    $label = $this->input->post('label', true);
    $geom = $this->input->post('geom', true);
    $query = $this->m_lokasi->tambah_kecamatan($nama_kecamatan, $kode_pos,$label,$geom);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Kecamatan Sukses di Simpan");
        redirect("c_lokasi/lihat_kecamatan");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Kecamatan Gagal di Simpan");
        redirect("c_lokasi/lihat_kecamatan");
    }
}

public function ubah_kecamatan($id_kecamatan) {
    $data1['data_kecamatan'] = $this->m_lokasi->ambil_kecamatan($id_kecamatan);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/ubah_kecamatan', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_ubah_kecamatan($id_kecamatan) {
    $nama_kecamatan = $this->input->post('nama_kecamatan', true);
    $kode_pos = $this->input->post('kode_pos', true);
    $label = $this->input->post('label', true);
    $geom = $this->input->post('geom', true);
    if($geom==""){
        $query = $this->m_lokasi->ubah_kecamatan($nama_kecamatan, $kode_pos, $id_kecamatan, $label);
    }
    else{
        $query = $this->m_lokasi->ubah_kecamatangeom($nama_kecamatan, $kode_pos, $id_kecamatan, $label,
$geom);
    }
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Kecamatan Sukses di Ubah");
        redirect("c_lokasi/lihat_kecamatan");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Kecamatan gagal di Ubah");
        redirect("c_lokasi/lihat_kecamatan");
    }
}

public function hapus_kecamatan($id_kecamatan) {
    $query = $this->m_lokasi->hapus_kecamatan($id_kecamatan);
    if ($query) {

```

```

$this->session->set_userdata('operation', "sukses");
$this->session->set_userdata('message', "Data Kecamatan Sukses di Hapus");
redirect("c_lokasi/lihat_kecamatan");
} else {
$this->session->set_userdata('operation', "gagal");
$this->session->set_userdata('message', "Data Kecamatan gagal di Hapus");
redirect("c_lokasi/lihat_kecamatan");
}
}

public function lihat_kelurahan() {
    $data1['tabel_kelurahan'] = $this->m_lokasi->tabel_kelurahan();
    $data = $this->session();
    $data['content'] = $this->load->view('admin/lihat_kelurahan', $data1, true);
    $this->load->view('layout/template', $data);
}

public function detail_kelurahan($id) {
    $kelurahan = $this->m_lokasi->data_kelurahan();
    $kelurahandetail = $this->m_lokasi->ambil_kelurahan($id);
    $this->load->library('googlemaps');
    $config['center'] = '-8.239383846027533, 113.6823472357595';
    $config['zoom'] = 10;
    $this->googlemaps->initialize($config);
    foreach ($kelurahan as $value) {
        $marker = array();
        $polygon1 = str_replace("MULTIPOLYGON", "", $value->lat);
        $polygon2 = str_replace(",", "", $polygon1);
        $polygon3 = str_replace(")", "", $polygon2);
        $polygon4 = str_replace(";", "", $polygon3);
        $polygon5 = str_replace(" ", "", $polygon4);
        $data = explode(";", $polygon5);
        $label = $value->label;
        $nama = $value->nama_kelurahan;
        $label2 = $kelurahandetail->label;
        if ($label == $label2) {
            $marker['fillColor'] = '#0F0';
        } else {
            $marker['fillColor'] = '#87CEEB';
        }
        for ($i = 0; $i < count($data); $i++) {
            $koordinat = explode(",", $data[$i]);
            $marker['points'][] = $koordinat[1] . ", " . $koordinat[0];
            $marker['onclick'] = "
                var infowindow = new google.maps.InfoWindow(
                    {
                        size: new google.maps.Size(150,50)
                    });
                infowindow.setContent('$label');
                if (event) {
                    point = event.latLng;
                }
                infowindow.setPosition(point);
                infowindow.open(map);";
        }
        $this->googlemaps->add_polygon($marker);
    }
    $data = $this->session();
    $data1['data_kelurahan'] = $this->m_lokasi->ambil_kelurahan($id);
    $data1['map'] = $this->googlemaps->create_map();
    $data['content'] = $this->load->view('admin/detail_kelurahan', $data1, true);
}

```



```

$this->load->view('layout/template', $data);
}

public function tambah_kelurahan() {
    $data1['option'] = $this->m_lokasi->tabel_kecamatan();
    $data = $this->session();
    $data['content'] = $this->load->view('admin/tambah_kelurahan', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_tambah_kelurahan() {
    $nama_kelurahan = $this->input->post('nama_kelurahan', true);
    $id_kecamatan = $this->input->post('nama_kecamatan', true);
    $label = $this->input->post('label', true);
    $geom = $this->input->post('geom', true);
    $query = $this->m_lokasi->tambah_kelurahan($nama_kelurahan, $id_kecamatan, $label, $geom);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Kelurahan Sukses di Simpan");
        redirect("c_lokasi/lihat_kelurahan");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Kelurahan Gagal di Simpan");
        redirect("c_lokasi/lihat_kelurahan");
    }
}

public function ubah_kelurahan($id_kelurahan) {
    $data1['option'] = $this->m_lokasi->tabel_kecamatan();
    $data1['data_kelurahan'] = $this->m_lokasi->ambil_kelurahan($id_kelurahan);
    $data = $this->session();
    $data['content'] = $this->load->view('admin/ubah_kelurahan', $data1, true);
    $this->load->view('layout/template', $data);
}

public function proses_ubah_kelurahan($id_kelurahan) {
    $nama_kelurahan = $this->input->post('nama_kelurahan', true);
    $id_kecamatan = $this->input->post('nama_kecamatan', true);
    $label = $this->input->post('label', true);
    $geom = $this->input->post('geom', true);
    if($geom==""){
        $query = $this->m_lokasi->ubah_kelurahan($nama_kelurahan, $id_kecamatan,$label, $id_kelurahan);
    }
    else{
        $query = $this->m_lokasi->ubah_kelurhangeom($nama_kelurahan, $id_kecamatan, $label,
        $id_kelurahan, $geom);
    }
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data Kelurahan Sukses di Ubah");
        redirect("c_lokasi/lihat_kelurahan");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Kelurahan gagal di Ubah");
        redirect("c_lokasi/lihat_kelurahan");
    }
}

public function hapus_kelurahan($id_kelurahan) {
    $query = $this->m_lokasi->hapus_kelurahan($id_kelurahan);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
    }
}

```

```

        $this->session->set_userdata('message', "Data Kelurahan Sukses di Hapus");
        redirect("c_lokasi/lihat_kelurahan");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data Kelurahan gagal di Hapus");
        redirect("c_lokasi/lihat_kelurahan");
    }
}

function session() {
    $data['nama'] = $this->session->userdata('nama');
    $data['pn'] = $this->session->userdata('pn');
    $data['id_user'] = $this->session->userdata('id_user');
    $data['jabatan'] = $this->session->userdata('jabatan');
    $data['id_hak_akses'] = $this->session->userdata('id_hak_akses');
    $data['jumlah_user'] = $this->m_user->ambil_jumlah_user();
    $data['jumlah_bri'] = $this->m_kriteria->ambil_jumlah_atmbri();
    $data['jumlah_aktif_analisa'] = $this->m_dss->ambil_analisa_aktif();
    $id = $this->session->userdata('id_user');
    $data['account'] = $this->m_user->ambil_akun($id);
    $data['jumlah_analisa'] = $this->m_dss->ambil_jumlah_analisa($id);
    return $data;
}
}

```

D.2 Kode Program Pada *Package Models*

1. *Class m_login*

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_login extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function login($username, $password) {
        $this->db->where('username', $username);
        $this->db->where('password', $password);
        $query = $this->db->get('tb_user');
        if ($query->num_rows > 0) {
            foreach ($query->result() as $row) {
                $data_ambil['login'] = 'ada';
                $data_ambil['nama'] = $row->nama;
                $data_ambil['jabatan'] = $row->jabatan;
                $data_ambil['pn'] = $row->pn;
                $data_ambil['id_user'] = $row->id_user;
                $data_ambil['id_hak_akses'] = $row->id_hak_akses;
                $data_ambil['foto'] = $row->foto;
                $this->session->set_userdata($data_ambil);
            }
            return true;
        } else {
            return false;
        }
    }
}

```

```

}
}

```

2. Class *m_dss*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_dss extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    //ADMIN
    function tabel_analisa_dss() {
        $query = $this->db->query("select distinct nama_transaksi, a.id_transaksi,nama_kecamatan, tgl_mulai,
        tgl_akhir, status from tb_transaksi a join tb_status b on a.id_status=b.id_status join tb_detail_transaksi c on
        a.id_transaksi=c.id_transaksi join tb_kecamatan d on a.id_kecamatan=d.id_kecamatan order by
        a.id_transaksi");
        return $query->result();
    }

    function ambil_user() {
        $query = $this->db->query("SELECT * from tb_user where id_hak_akses='2'");
        return $query->result();
    }

    function ambil_analisa_dss($id_transaksi) {
        $query = $this->db->query("SELECT distinct id_transaksi, nama_transaksi, tgl_mulai, tgl_akhir,
        a.id_kecamatan, nama_kecamatan from tb_transaksi a join tb_kecamatan b on a.id_kecamatan=b.id_kecamatan
        where id_transaksi='$id_transaksi'");
        return $query->row();
    }

    function ambil_dm_analisa($id_transaksi) {
        $query = $this->db->query("SELECT nama, a.id_status from tb_detail_transaksi a join tb_user b on
        a.id_user=b.id_user where id_transaksi='$id_transaksi'");
        return $query->result_array();
    }

    function ambil_detail_analisa($id_transaksi) {
        $query = $this->db->query("SELECT * from tb_detail_transaksi where id_transaksi='$id_transaksi'");
        return $query->result_array();
    }

    function tambah_analisa_dss($nama_transaksi, $tgl_mulai, $tgl_akhir, $id_kecamatan) {
        $this->db->set('nama_transaksi', $nama_transaksi);
        $this->db->set('tgl_mulai', $tgl_mulai);
        $this->db->set('tgl_akhir', $tgl_akhir);
        $this->db->set('id_kecamatan', $id_kecamatan);
        $this->db->set('id_status', '1');
        $query = $this->db->insert('tb_transaksi');
        if ($query) {
            return true;
        } else {
            return false;
        }
    }
}

```

```
}
}

function ambil_idakhir_detail() {
    $query = $this->db->query("SELECT id_transaksi from tb_transaksi order by id_transaksi desc limit 1");
    return $query->row();
}

function ambil_idakhir_kriteria() {
    $query = $this->db->query("SELECT id_kriteria from tb_kriteria order by id_kriteria desc limit 1");
    return $query->row();
}

function tambah_detail_analisa($id_transaksi, $id_user) {
    $this->db->set('id_transaksi', $id_transaksi);
    $this->db->set('id_user', $id_user);
    $this->db->set('id_status', '2');
    $query = $this->db->insert('tb_detail_transaksi');
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_status_analisa($id_transaksi, $status) {
    $data = array(
        'id_status' => $status
    );
    $this->db->where('id_transaksi', $id_transaksi);
    $query = $this->db->update('tb_transaksi', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_status_dm($id_transaksi, $id_user) {
    $data = array(
        'id_status' => 1
    );
    $this->db->where('id_transaksi', $id_transaksi);
    $this->db->where('id_user', $id_user);
    $query = $this->db->update('tb_detail_transaksi', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function hapus_detail_analisa($id_transaksi) {
    $query = $this->db->delete('tb_detail_transaksi', array('id_transaksi' => $id_transaksi));
    $query1 = $this->db->delete('tb_entropy', array('id_transaksi' => $id_transaksi));
    $query2 = $this->db->delete('tb_vikor', array('id_transaksi' => $id_transaksi));
    $query3 = $this->db->delete('tb_borda', array('id_transaksi' => $id_transaksi));
    if ($query && $query1 && $query2 && $query3) {
        return true;
    } else {
        return false;
    }
}
```

```

}

function hapus_analisa_dss($id_transaksi) {
    $query = $this->db->delete('tb_transaksi', array('id_transaksi' => $id_transaksi));
    if ($query) {
        return true;
    } else {
        return false;
    }
}

//DM
function tabel_transaksi($id) {
    $query = $this->db->query("select a.id_transaksi, tgl_mulai, tgl_akhir, nama_kecamatan, a.id_status as
statustransaksi, c.id_status as statususer, nama_transaksi from tb_transaksi a join tb_status b on
a.id_status=b.id_status join tb_detail_transaksi c on a.id_transaksi=c.id_transaksi join tb_kecamatan d on
a.id_kecamatan=d.id_kecamatan where c.id_user='$id'");
    return $query->result();
}

function tabel_lihat_dss($id_transaksi, $id_user) {
    $query = $this->db->query("select a.id_kelurahan,
nama_kelurahan,ranking,b.id_kelurahan,label,st_astext(geom) as lat from tb_vikor a join tb_kelurahan b on
a.id_kelurahan=b.id_kelurahan where id_transaksi='$id_transaksi' and id_user='$id_user' order by ranking");
    return $query->result();
}

function ambil_ranking_terbaik($id_transaksi, $id_user) {
    $query = $this->db->query("select a.id_kelurahan, nama_kelurahan from tb_vikor a join tb_kelurahan b
on a.id_kelurahan=b.id_kelurahan where id_transaksi='$id_transaksi' and id_user='$id_user' and ranking=1 ");
    return $query->result();
}

function tabel_lihat_gdss($id_transaksi) {
    $query = $this->db->query("select a.id_kelurahan,
nama_kecamatan,nama_kelurahan,nilai_borda,ranking,b.id_kelurahan,b.label,st_astext(b.geom) as lat from
tb_borda a join tb_kelurahan b on a.id_kelurahan=b.id_kelurahan join tb_kecamatan c on
b.id_kecamatan=c.id_kecamatan where id_transaksi='$id_transaksi' order by ranking");
    return $query->result_array();
}

function peta_gdss($id_transaksi) {
    $query = $this->db->query("select a.id_kelurahan,
nama_kecamatan,nama_kelurahan,nilai_borda,ranking,b.id_kelurahan,b.label,st_astext(b.geom) as lat from
tb_borda a join tb_kelurahan b on a.id_kelurahan=b.id_kelurahan join tb_kecamatan c on
b.id_kecamatan=c.id_kecamatan where id_transaksi='$id_transaksi' order by ranking");
    return $query->result();
}

function data_atm($id_transaksi) {
    $query = $this->db->query("select icon,nama,alamat,info,st_y(st_astext(a.geom)) as lat,
st_x(st_astext(a.geom)) as long from tb_nilai_kriteria a join tb_kategori_library b on
a.id_kategori_library=b.id_kategori_library
join tb_kelurahan c on a.id_kelurahan=c.id_kelurahan join tb_borda d on c.id_kelurahan=d.id_kelurahan
where d.id_transaksi='$id_transaksi' and a.id_kategori_library=1");
    return $query->result();
}

function ambil_ranking_terbaik_gdss($id_transaksi) {
    $query = $this->db->query("select a.id_kelurahan, nama_kelurahan from tb_borda a join tb_kelurahan b
on a.id_kelurahan=b.id_kelurahan where id_transaksi='$id_transaksi' and ranking=1 ");
    return $query->result();
}

```

```

}

function ambil_detail_transaksi($id_transaksi, $id_user) {
    $query = $this->db->query("select * from tb_detail_transaksi where id_transaksi='$id_transaksi' and
id_user='$id_user'");
    return $query->row();
}

function hapus_analisa_dm($id_transaksi, $id_user) {
    $query = $this->db->delete('tb_vikor', array
('id_transaksi' => $id_transaksi,
'id_user' => $id_user
));
    $query2 = $this->db->delete('tb_entropy', array
('id_transaksi' => $id_transaksi,
'id_user' => $id_user
));
    if ($query && $query2) {
        return true;
    } else {
        return false;
    }
}

function tambah_bobot_awal($id_transaksi, $id_user, $id_kriteria, $bobot_awal) {
    $this->db->set('id_transaksi', $id_transaksi);
    $this->db->set('id_user', $id_user);
    $this->db->set('id_kriteria', $id_kriteria);
    $this->db->set('bobot_awal', $bobot_awal);
    $this->db->set('bobot_entropy', '0');
    $query = $this->db->insert('tb_entropy');
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function metrik($id_transaksi, $id_user) {
    $data = $this->db->query("select a.nama_kelurahan, e.nama_kriteria, f.nilai as nilai_kepentingan
from tb_kelurahan a
join tb_kecamatan b on a.id_kecamatan=b.id_kecamatan
join tb_transaksi c on b.id_kecamatan=c.id_kecamatan
join tb_entropy d on c.id_transaksi=d.id_transaksi
join tb_kriteria e on d.id_kriteria=e.id_kriteria
join tb_nilai_kriteria f on e.id_kriteria=f.id_kriteria
where c.id_transaksi='$id_transaksi' and d.id_user='$id_user' and nilai!=0 and
a.id_kelurahan=f.id_kelurahan
group by a.id_kelurahan,e.nama_kriteria,nilai_kepentingan
order by a.nama_kelurahan, e.nama_kriteria");
    return $data->result_array();
}

function ambil_kelurahan($id_transaksi, $id_user) {
    $data = $this->db->query("SELECT distinct nama_kelurahan from tb_entropy a join tb_transaksi b on
a.id_transaksi=b.id_transaksi join tb_kecamatan c on b.id_kecamatan=c.id_kecamatan join tb_kelurahan d on
c.id_kecamatan=d.id_kecamatan where a.id_transaksi='$id_transaksi' and a.id_user='$id_user' order by
nama_kelurahan");
    return $data->result_array();
}

function ambil_kriteria($id_transaksi, $id_user) {

```

```

    $data = $this->db->query("SELECT distinct nama_kriteria,bobot_awal,bobot_entropy,a.id_kriteria as
id_kriteria,id_sifat_kriteria from tb_entropy a join tb_kriteria b on a.id_kriteria=b.id_kriteria where
a.id_transaksi='$id_transaksi' and a.id_user='$id_user' order by nama_kriteria");
    return $data->result_array();
}

function simpan_bobot_entropy($id_transaksi, $id_user, $id_kriteria, $bobot_entropy) {
    $data = array(
        'bobot_entropy' => $bobot_entropy
    );
    $this->db->where('id_transaksi', $id_transaksi);
    $this->db->where('id_user', $id_user);
    $this->db->where('id_kriteria', $id_kriteria);
    $query = $this->db->update('tb_entropy', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function simpan_vikor($id_transaksi, $id_user, $id, $ranking) {
    $this->db->set('id_transaksi', $id_transaksi);
    $this->db->set('id_user', $id_user);
    $this->db->set('id_kelurahan', $id);
    $this->db->set('ranking', $ranking);
    $query = $this->db->insert('tb_vikor');
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function cek_kelurahan($nama) {
    $query = $this->db->query("select id_kelurahan from tb_kelurahan where nama_kelurahan='$nama'");
    return $query->row();
}

function ambil_vikor($id_transaksi) {
    $query = $this->db->query("select b.nama_kelurahan,c.nama,a.ranking
from tb_vikor a
join tb_kelurahan b on a.id_kelurahan=b.id_kelurahan
join tb_user c
on a.id_user=c.id_user
where a.id_transaksi='$id_transaksi'
group by a.id_kelurahan,b.nama_kelurahan,c.nama,a.ranking
order by b.nama_kelurahan,c.nama
");
    return $query->result_array();
}

function ambil_vikor_kelurahan($id_transaksi) {
    $query = $this->db->query("select distinct nama_kelurahan from tb_vikor a join tb_kelurahan b on
a.id_kelurahan=b.id_kelurahan where id_transaksi='$id_transaksi' order by nama_kelurahan");
    return $query->result_array();
}

function ambil_vikor_user($id_transaksi) {
    $query = $this->db->query("select distinct nama from tb_vikor a join tb_user b on a.id_user=b.id_user
where id_transaksi='$id_transaksi' order by nama");
    return $query->result_array();
}

```

```
}  
  
function cek_borda($id_transaksi) {  
    $query = $this->db->query("select * from tb_borda where id_transaksi='$id_transaksi'");  
    return $query->row();  
}  
  
function simpan_borda($id_transaksi, $id, $ranking, $nilai_borda) {  
    $this->db->set('id_transaksi', $id_transaksi);  
    $this->db->set('id_kelurahan', $id);  
    $this->db->set('ranking', $ranking);  
    $this->db->set('nilai_borda', $nilai_borda);  
    $query = $this->db->insert('tb_borda');  
    if ($query) {  
        return true;  
    } else {  
        return false;  
    }  
}  
  
function ubah_borda($id_transaksi, $id, $ranking, $nilai_borda) {  
    $data = array(  
        'ranking' => $ranking,  
        'nilai_borda' => $nilai_borda  
    );  
    $this->db->where('id_transaksi', $id_transaksi);  
    $this->db->where('id_kelurahan', $id);  
    $query = $this->db->update('tb_borda', $data);  
    if ($query) {  
        return true;  
    } else {  
        return false;  
    }  
}  
  
function ubah_analisa_dss($nama_transaksi, $tgl_mulai, $tgl_akhir, $id_kecamatan, $id_transaksi) {  
    $data = array(  
        'nama_transaksi' => $nama_transaksi,  
        'tgl_mulai' => $tgl_mulai,  
        'tgl_akhir' => $tgl_akhir,  
        'id_kecamatan' => $id_kecamatan  
    );  
    $this->db->where('id_transaksi', $id_transaksi);  
    $query = $this->db->update('tb_transaksi', $data);  
    if ($query) {  
        return true;  
    } else {  
        return false;  
    }  
}  
  
function ambil_jumlah_analisa($id_user){  
    $query = $this->db->query("select count(id_transaksi) as jumlah from tb_detail_transaksi where  
id_status='2' and id_user='$id_user'");  
    return $query->row();  
}  
  
function ambil_analisa_aktif(){  
    $query = $this->db->query("select count(id_transaksi) as jumlah from tb_transaksi where id_status='1'");  
    return $query->row();  
}
```



```
}

```

3. Class *m_kriteria*

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_kriteria extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function tabel_kriteria() {
        $query = $this->db->query('select * from tb_kriteria a join tb_sifat_kriteria b on
a.id_sifat_kriteria=b.id_sifat_kriteria join tb_jenis_kriteria c on a.id_jenis_kriteria=c.id_jenis_kriteria join
tb_bentuk_kriteria d on a.id_bentuk_kriteria=d.id_bentuk_kriteria order by a.id_kriteria');
        return $query->result();
    }

    function ambil_jumlah_kriteria() {
        $query = $this->db->query('select count(id_kriteria) as jumlah from tb_kriteria');
        return $query->row();
    }

    function tambah_kriteria($nama_kriteria, $sid_sifat_kriteria, $ket, $sid_jenis_kriteria, $sid_bentuk_kriteria) {
        $this->db->set('nama_kriteria', $nama_kriteria);
        $this->db->set('id_sifat_kriteria', $sid_sifat_kriteria);
        $this->db->set('id_jenis_kriteria', $sid_jenis_kriteria);
        $this->db->set('id_bentuk_kriteria', $sid_bentuk_kriteria);
        $this->db->set('keterangan', $ket);
        $query = $this->db->insert('tb_kriteria');
        if ($query) {
            return true;
        } else {
            return false;
        }
    }

    function ambil_kriteria($sid_kriteria) {
        $query = $this->db->query("select a.id_jenis_kriteria, a.id_kriteria, nama_kriteria, a.id_bentuk_kriteria
from tb_kriteria a join tb_jenis_kriteria b on a.id_jenis_kriteria=b.id_jenis_kriteria where
id_kriteria=$sid_kriteria");
        return $query->row();
    }

    function ubah_kriteria($nama_kriteria, $sid_sifat_kriteria, $keterangan, $sid_kriteria, $sid_jenis_kriteria,
$sid_bentuk_kriteria) {
        $data = array(
            'nama_kriteria' => $nama_kriteria,
            'id_sifat_kriteria' => $sid_sifat_kriteria,
            'id_jenis_kriteria' => $sid_jenis_kriteria,
            'id_bentuk_kriteria' => $sid_bentuk_kriteria,
            'keterangan' => $keterangan
        );
        $this->db->where('id_kriteria', $sid_kriteria);
        $query = $this->db->update('tb_kriteria', $data);
    }
}

```

```
if ($query) {
    return true;
} else {
    return false;
}
}

function hapus_kriteria($id_kriteria) {
    $query = $this->db->delete('tb_kriteria', array('id_kriteria' => $id_kriteria));
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function range_kriteria($id_kriteria) {
    $query = $this->db->query("select * from tb_range a join tb_kriteria b on a.id_kriteria=b.id_kriteria where
b.id_kriteria='$id_kriteria'");
    return $query->result();
}

function tambah_range_kriteria($batas_atas, $batas_bawah, $nk, $id_kriteria, $range) {
    $this->db->set('batas_atas', $batas_atas);
    $this->db->set('batas_bawah', $batas_bawah);
    $this->db->set('range', $range);
    $this->db->set('nilai_kepentingan', $nk);
    $this->db->set('id_kriteria', $id_kriteria);
    $query = $this->db->insert('tb_range');
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_range_kriteria($batas_atas, $batas_bawah, $nk, $id_range, $id_kriteria, $range) {
    $data = array(
        'batas_atas' => $batas_atas,
        'batas_bawah' => $batas_bawah,
        'range' => $range,
        'nilai_kepentingan' => $nk
    );
    $this->db->where('id_kriteria', $id_kriteria);
    $this->db->where('id_range', $id_range);
    $query = $this->db->update('tb_range', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function tabel_kategori($id_kriteria) {
    $query = $this->db->query("select id_kategori_library, kategori_library,nama_kriteria,icon, a.keterangan
as ket from tb_kategori_library a join tb_kriteria b on a.id_kriteria=b.id_kriteria where
a.id_kriteria='$id_kriteria'");
    return $query->result();
}

function tambah_kategori($nama, $id_kriteria, $keterangan, $icon) {
```

```

    $query = $this->db->query("INSERT INTO
tb_kategori_library(kategori_library,id_kriteria,keterangan,icon)
VALUES ('$nama','$id_kriteria','$keterangan','$icon');");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ambil_kategori($id) {
    $this->db->where('id_kategori_library', $id);
    $query = $this->db->get('tb_kategori_library');
    return $query->row();
}

function ubah_kategori($nama, $keterangan, $id) {
    $query = $this->db->query("UPDATE tb_kategori_library SET kategori_library='$nama',
keterangan='$keterangan' WHERE id_kategori_library='$id'");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_kategori_icon($nama, $keterangan, $id, $foto_1) {
    $data = array(
        'kategori_library' => $nama,
        'keterangan' => $keterangan,
        'icon' => $foto_1
    );
    $this->db->where('id_kategori_library', $id);
    $query = $this->db->update('tb_kategori_library', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_kategori_tanpa_icon($nama, $keterangan, $id) {
    $data = array(
        'kategori_library' => $nama,
        'keterangan' => $keterangan
    );
    $this->db->where('id_kategori_library', $id);
    $query = $this->db->update('tb_kategori_library', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function check_icon($id) {
    $query = $this->db->query("select icon from tb_kategori_library where id_kategori_library='" . $id . "'");
    return $query->row();
}

function hapus_icon($id) {

```

```

    $query = $this->db->query("UPDATE tb_kategori_library SET icon = " WHERE id_kategori_library = "
    . $id . """);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function hapus_kategori($id_kategori_library) {
    $query = $this->db->delete('tb_kategori_library', array('id_kategori_library' => $id_kategori_library));
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function tabel_nilai_kriteria($id_kriteria) {
    $query = $this->db->query("select * from tb_nilai_kriteria a join tb_kelurahan b on
a.id_kelurahan=b.id_kelurahan join tb_kriteria d on a.id_kriteria=d.id_kriteria where
a.id_kriteria='$id_kriteria'");
    return $query->result();
}

function ambil_deskriptif($id_kriteria) {
    $query = $this->db->query("select range,nilai_kepentingan from tb_range a join tb_kriteria b on
a.id_kriteria=b.id_kriteria where a.id_kriteria='$id_kriteria' order by id_range");
    return $query->result();
}

function tabel_nilai_kriteria_peta($id_kategori) {
    $query = $this->db->query("select * from tb_nilai_kriteria a join tb_kelurahan b on
a.id_kelurahan=b.id_kelurahan join tb_kategori_library c on a.id_kategori_library=c.id_kategori_library where
a.id_kategori_library='$id_kategori'");
    return $query->result();
}

function tambah_nilai_kriteria($id_kelurahan, $id_kriteria, $nilai) {
    $this->db->set('id_kelurahan', $id_kelurahan);
    $this->db->set('id_kriteria', $id_kriteria);
    $this->db->set('nilai', $nilai);
    $query = $this->db->insert('tb_nilai_kriteria');
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function tambah_nilai_kriteriaasli($id_kelurahan, $id_kriteria, $nilaiasli) {
    $this->db->set('id_kelurahan', $id_kelurahan);
    $this->db->set('id_kriteria', $id_kriteria);
    $this->db->set('nilai_asli', $nilaiasli);
    $query = $this->db->insert('tb_nilai_kriteria');
    if ($query) {
        return true;
    } else {
        return false;
    }
}

```

```

function ambil_nilai_kriteria($id_nilai) {
    $this->db->where('id_nilai', $id_nilai);
    $query = $this->db->get('tb_nilai_kriteria');
    return $query->row();
}

function ubah_nilai_kriteria($id_kelurahan, $nilai, $id_nilai) {
    $data = array(
        'id_kelurahan' => $id_kelurahan,
        'nilai' => $nilai
    );
    $this->db->where('id_nilai', $id_nilai);
    $query = $this->db->update('tb_nilai_kriteria', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_nilaiasli_kriteria($id_kelurahan, $nilai, $id_nilai) {
    $data = array(
        'id_kelurahan' => $id_kelurahan,
        'nilai_asli' => $nilai
    );
    $this->db->where('id_nilai', $id_nilai);
    $query = $this->db->update('tb_nilai_kriteria', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_nilai_kategori($id_nilai, $nilai) {
    $query = $this->db->query("UPDATE public.tb_nilai_kriteria
SET nilai_asli='$nilai'
WHERE id_nilai='$id_nilai'");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function hapus_nilai_kriteria($id_nilai) {
    $query = $this->db->delete('tb_nilai_kriteria', array('id_nilai' => $id_nilai));
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function tambah_nilai_kriteria_peta($id_kriteria, $nama, $alamat, $id_kelurahan, $info, $id_kategori,
$geom) {
    $query = $this->db->query("INSERT INTO
tb_nilai_kriteria(id_kriteria,nama,alamat,id_kelurahan,info,id_kategori_library,geom)
VALUES
('$id_kriteria','$nama','$alamat','$id_kelurahan','$info','$id_kategori',ST_GeomFromGeoJSON('$geom'))");
    if ($query) {
        return true;
    }
}

```

```

    } else {
        return false;
    }
}

function ubah_nilai_kriteria_peta($nama, $alamat, $id_kelurahan, $info, $id_nilai) {
    $query = $this->db->query("UPDATE tb_nilai_kriteria SET nama='$nama', alamat='$alamat',
    id_kelurahan='$id_kelurahan',info='$info' WHERE id_nilai='$id_nilai'");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_nilai_kriteria_petageom($nama, $alamat, $id_kelurahan, $info, $id_nilai, $geom) {
    $query = $this->db->query("UPDATE tb_nilai_kriteria SET nama='$nama',alamat='$alamat',
    id_kelurahan='$id_kelurahan',info='$info', geom=ST_GeomFromGeoJSON('$geom') WHERE
    id_nilai='$id_nilai'");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ambil_nilai_kepentingansemua($id_kriteria) {
    $query = $this->db->query("select a.id_kelurahan, a.id_kriteria,nilai_kepentingan,nilai_asli
    from tb_nilai_kriteria a join tb_kriteria b
    on a.id_kriteria=b.id_kriteria
    join tb_range c on b.id_kriteria=c.id_kriteria
    where a.id_kriteria='$id_kriteria' and (nilai_asli>batas_bawah or nilai_asli=batas_bawah) and
    (nilai_asli<batas_atas or nilai_asli=batas_atas)
    and a.id_kriteria=c.id_kriteria
    order by a.id_kelurahan");
    return $query->result();
}

function ambil_nilai_kepentingan_kategori($id_kriteria, $id_kelurahan) {
    $query = $this->db->query("select a.id_kelurahan, a.id_kriteria,nilai_kepentingan,nilai_asli,id_nilai
    from tb_nilai_kriteria a join tb_kriteria b
    on a.id_kriteria=b.id_kriteria
    join tb_range c on b.id_kriteria=c.id_kriteria
    where nilai_asli!=0 and a.id_kriteria='$id_kriteria' and a.id_kelurahan='$id_kelurahan' and
    (nilai_asli>batas_bawah or nilai_asli=batas_bawah) and (nilai_asli<batas_atas or nilai_asli=batas_atas)
    and a.id_kriteria=c.id_kriteria
    order by a.id_kelurahan");
    return $query->result_array();
}

function ambil_nilai_kepentingan($id_kriteria, $id_nilai, $id_kelurahan) {
    $query = $this->db->query("select a.id_kelurahan, a.id_kriteria,nilai_kepentingan,nilai_asli,id_nilai
    from tb_nilai_kriteria a join tb_kriteria b
    on a.id_kriteria=b.id_kriteria
    join tb_range c on b.id_kriteria=c.id_kriteria
    where a.id_kriteria='$id_kriteria' and id_nilai='$id_nilai' and a.id_kelurahan='$id_kelurahan' and
    (nilai_asli>batas_bawah or nilai_asli=batas_bawah) and (nilai_asli<batas_atas or nilai_asli=batas_atas)
    and a.id_kriteria=c.id_kriteria
    order by a.id_kelurahan");
    return $query->result();
}

```

```

function ambil_range_deskriptif($id_nilai, $id_kriteria) {
    $query = $this->db->query("select a.id_nilai, a.id_kelurahan, a.id_kriteria, nilai, range
        from tb_nilai_kriteria a join tb_kriteria b
        on a.id_kriteria=b.id_kriteria
        join tb_range c on b.id_kriteria=c.id_kriteria
        where a.id_kriteria='$id_kriteria' and a.id_nilai='$id_nilai' and a.nilai=c.nilai_kepentingan
        and a.id_kriteria=c.id_kriteria
        group by a.id_nilai, a.id_kriteria, nilai, range, a.id_kelurahan
        order by a.id_nilai");
    return $query->row();
}

function ubah_nilai_asli($id_kelurahan, $id_kriteria, $nilai) {
    $data = array(
        'nilai' => $nilai
    );
    $this->db->where('id_kelurahan', $id_kelurahan);
    $this->db->where('id_kriteria', $id_kriteria);
    $query = $this->db->update('tb_nilai_kriteria', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_nilai_aslinum($id_kelurahan, $id_kriteria, $nilai, $id_nilai) {
    $data = array(
        'nilai' => $nilai
    );
    $this->db->where('id_kelurahan', $id_kelurahan);
    $this->db->where('id_kriteria', $id_kriteria);
    $this->db->where('id_nilai', $id_nilai);
    $query = $this->db->update('tb_nilai_kriteria', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_nilai_asli_peta($id_nilai, $nilai) {
    $query = $this->db->query("UPDATE tb_nilai_kriteria SET nilai='$nilai' WHERE id_nilai='$id_nilai'");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function tabel_nilai_kategori($id_nilai) {
    $query = $this->db->query("select icon, nama, alamat, info, st_y(st_astext(a.geom)) as lat,
        st_x(st_astext(a.geom)) as long from tb_nilai_kriteria a join tb_kategori_library b on
        a.id_kategori_library=b.id_kategori_library where id_nilai='$id_nilai'");
    return $query->result();
}

function pilih_kategori($kategori) {
    $query = $this->db->query("select icon, nama, alamat, info, st_y(st_astext(a.geom)) as lat,
        st_x(st_astext(a.geom)) as long from tb_nilai_kriteria a join tb_kategori_library b on
        a.id_kategori_library=b.id_kategori_library where b.id_kategori_library='$kategori'");
    return $query->result();
}

```

```
}

function pilih_kategori_detail($kategori, $kecamatan) {
    $query = $this->db->query("select icon,nama,alamat,info,st_y(st_astext(a.geom)) as lat,
st_x(st_astext(a.geom)) as long from tb_nilai_kriteria a join tb_kategori_library b on
a.id_kategori_library=b.id_kategori_library join tb_kelurahan c on a.id_kelurahan=c.id_kelurahan join
tb_kecamatan d on c.id_kecamatan=d.id_kecamatan where b.id_kategori_library='$kategori' and
d.id_kecamatan='$kecamatan'");
    return $query->result();
}

function data_kategori() {
    $query = $this->db->query("select icon,nama,alamat,info,st_y(st_astext(a.geom)) as lat,
st_x(st_astext(a.geom)) as long from tb_nilai_kriteria a join tb_kategori_library b on
a.id_kategori_library=b.id_kategori_library");
    return $query->result();
}

function detail_kategori($kecamatan) {
    $query = $this->db->query("select icon,nama,alamat,info,st_y(st_astext(a.geom)) as lat,
st_x(st_astext(a.geom)) as long from tb_nilai_kriteria a join tb_kategori_library b on
a.id_kategori_library=b.id_kategori_library join tb_kelurahan c on a.id_kelurahan=c.id_kelurahan join
tb_kecamatan d on c.id_kecamatan=d.id_kecamatan where d.id_kecamatan='$kecamatan'");
    return $query->result();
}

function jumlah_kategori() {
    $query = $this->db->query("select a.id_kategori_library,icon,kategori_library, count(id_nilai) as jumlah
from tb_kategori_library a join tb_nilai_kriteria b on a.id_kategori_library=b.id_kategori_library
group by a.id_kategori_library order by a.id_kategori_library");
    return $query->result();
}

function cek_nilai($kriteria, $kelurahan) {
    $query = $this->db->query("select nilai_asli,id_nilai from tb_nilai_kriteria where nilai_asli!=0 and
id_kriteria='$kriteria' and id_kelurahan='$kelurahan'");
    return $query->result_array();
}

function combo_kategori() {
    $query = $this->db->query("select * from tb_kategori_library");
    return $query->result();
}

function combo_sifat_kriteria() {
    $query = $this->db->query("SELECT * from tb_sifat_kriteria");
    return $query->result();
}

function combo_jenis_kriteria() {
    $query = $this->db->query("SELECT * from tb_jenis_kriteria");
    return $query->result();
}

function combo_bentuk_kriteria() {
    $query = $this->db->query("SELECT * from tb_bentuk_kriteria");
    return $query->result();
}

function ambil_range($id_kriteria, $id_range) {
    $query = $this->db->query("select * from tb_range a join tb_kriteria b on a.id_kriteria=b.id_kriteria where
a.id_kriteria='$id_kriteria' and id_range='$id_range'");
}
```



```

        return $query->row();
    }

    function ambil_jumlah_atmbri() {
        $query = $this->db->query("select count(id_nilai) as jumlah from tb_kategori_library a join
        tb_nilai_kriteria b on a.id_kategori_library=b.id_kategori_library
        where kategori_library like '%BRI%'");
        return $query->row();
    }
}

```

4. *Class m_lokasi*

```

<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_lokasi extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function tabel_kecamatan() {
        $query = $this->db->query('select * from tb_kecamatan where nama_kecamatan is not null');
        return $query->result();
    }

    function data_kecamatan() {
        $query = $this->db->query('select id_kecamatan,label,nama_kecamatan,st_astext(geom) as lat from
        tb_kecamatan');
        return $query->result();
    }

    function data_kategori() {
        $query = $this->db->query('SELECT id_kategori_library, id_kriteria, kategori_library,
        keterangan FROM public.tb_kategori_library;');
        return $query->result();
    }

    function tambah_kecamatan($nama_kecamatan, $kode_pos, $label, $geom) {
        $query = $this->db->query("INSERT INTO tb_kecamatan(nama_kecamatan,kode_pos,label,geom)
        VALUES ('$nama_kecamatan','$kode_pos','$label',
        ST_AsText(ST_Multi(ST_GeomFromGeoJSON('$geom'))));");
        if ($query) {
            return true;
        } else {
            return false;
        }
    }

    function ambil_kecamatan($id_kecamatan) {
        $this->db->where('id_kecamatan', $id_kecamatan);
        $query = $this->db->get('tb_kecamatan');
        return $query->row();
    }

    function ubah_kecamatan($nama_kecamatan, $kode_pos, $id_kecamatan, $label) {

```

```

    $query = $this->db->query("UPDATE tb_kecamatan SET nama_kecamatan='$nama_kecamatan',
label='$label',
    kode_pos='$kode_pos', id_status='2' WHERE id_kecamatan='$id_kecamatan'");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_kecamatangeom($nama_kecamatan, $kode_pos, $id_kecamatan, $label, $geom) {
    $query = $this->db->query("UPDATE tb_kecamatan SET nama_kecamatan='$nama_kecamatan',
label='$label',
    geom=ST_GeomFromGeoJSON('$geom'),      kode_pos='$kode_pos',      id_status='2'      WHERE
id_kecamatan='$id_kecamatan'");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function hapus_kecamatan($id_kecamatan) {
    $query = $this->db->delete('tb_kecamatan', array('id_kecamatan' => $id_kecamatan));
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function tabel_kelurahan() {
    $query = $this->db->query('select a.id_kelurahan, a.nama_kelurahan, b.id_kecamatan,
b.nama_kecamatan, a.shape_area, a.shape_leng from tb_kelurahan a join tb_kecamatan b on
a.id_kecamatan=b.id_kecamatan');
    return $query->result();
}

function data_kelurahan() {
    $query = $this->db->query('select id_kelurahan,label,nama_kelurahan,st_astext(geom) as lat from
tb_kelurahan');
    return $query->result();
}

function tambah_kelurahan($nama_kelurahan, $id_kecamatan, $label, $geom) {
    $query = $this->db->query("INSERT INTO tb_kelurahan(nama_kelurahan,label,id_kecamatan,geom)
VALUES
('$nama_kelurahan','$label','$id_kecamatan',
ST_AsText(ST_Multi(ST_GeomFromGeoJSON('$geom'))));");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ambil_kelurahan($id_kelurahan) {
    $query = $this->db->query("select
id_kelurahan,a.label,nama_kelurahan,nama_kecamatan,a.shape_leng,a.shape_area from tb_kelurahan a join
tb_kecamatan b on a.id_kecamatan=b.id_kecamatan where id_kelurahan=$id_kelurahan");
    return $query->row();
}

```

```

function ubah_kelurahan($nama_kelurahan, $id_kecamatan, $label, $id_kelurahan) {
    $query = $this->db->query("UPDATE public.tb_kelurahan SET nama_kelurahan='$nama_kelurahan',
label='$label',
id_kecamatan='$id_kecamatan'
WHERE id_kelurahan='$id_kelurahan'");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_kelurahangeom($nama_kelurahan, $id_kecamatan, $label, $id_kelurahan, $geom) {
    $query = $this->db->query("UPDATE public.tb_kelurahan SET nama_kelurahan='$nama_kelurahan',
label='$label',
geom=ST_GeomFromGeoJSON('$geom'), id_kecamatan='$id_kecamatan'
WHERE id_kelurahan='$id_kelurahan'");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function hapus_kelurahan($id_kelurahan) {
    $query = $this->db->delete('tb_kelurahan', array('id_kelurahan' => $id_kelurahan));
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function combo_kecamatan() {
    $query = $this->db->query('select * from tb_kecamatan where nama_kecamatan is not null');
    return $query->result();
}

function combo_kelurahan() {
    $query = $this->db->query('select * from tb_kelurahan');
    return $query->result();
}

function pilih_kelurahan($id_kecamatan) {
    $query = $this->db->query("select id_kelurahan,a.label,nama_kelurahan,st_astext(a.geom) as lat from
tb_kelurahan a join tb_kecamatan b on a.id_kecamatan=b.id_kecamatan where
b.id_kecamatan='$id_kecamatan'");
    return $query->result();
}
}

```

5. *Class m_user*

```

<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class M_user extends CI_Model {

```

```

function __construct() {
    parent::__construct();
}

function tabel_user() {
    $query = $this->db->query('select * from tb_user a join tb_hak_akses b on a.id_hak_akses=b.id_hak_akses
join tb_status c on a.id_status=c.id_status order by a.id_user ');
    return $query->result();
}

function cek_pn() {
    $query = $this->db->query("select pn from tb_user");
    return $query->result_array();
}

function cek_ubah_pn($id_user) {
    $query = $this->db->query("select pn from tb_user where id_user!='$id_user'");
    return $query->result_array();
}

function ambil_user($id_user) {
    $this->db->where('id_user', $id_user);
    $query = $this->db->get('tb_user');
    return $query->row();
}

function ambil_detail_user($id_user){
    $query = $this->db->query("select * from tb_user a join tb_hak_akses b on
a.id_hak_akses=b.id_hak_akses join tb_jenis_kelamin c on a.id_jenis_kelamin=c.id_jenis_kelamin where
id_user =' . $id_user . '");
    return $query->row();
}

function tambah_user($pn, $nama, $jenis_kelamin, $format, $alamat, $jabatan, $no_hp, $id_hak_akses,
$username, $foto, $password) {
    $this->db->set('pn', $pn);
    $this->db->set('nama', $nama);
    $this->db->set('id_jenis_kelamin', $jenis_kelamin);
    $this->db->set('tgl_lahir', $format);
    $this->db->set('alamat', $alamat);
    $this->db->set('jabatan', $jabatan);
    $this->db->set('no_hp', $no_hp);
    $this->db->set('id_hak_akses', $id_hak_akses);
    $this->db->set('username', $username);
    $this->db->set('password', $password);
    $this->db->set('foto', $foto);
    $this->db->set('id_status', '1');
    $query = $this->db->insert('tb_user');
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_user_foto($id_user,$pn, $nama, $jenis_kelamin, $format, $alamat, $jabatan, $no_hp,
$id_hak_akses,$foto_1) {
    $data = array(
        'nama' => $nama,
        'pn' => $pn,

```

```
'id_jenis_kelamin' => $jenis_kelamin,
'alamat' => $alamat,
'tgl_lahir' => $format,
'jabatan' => $jabatan,
'foto' => $foto_1,
'no_hp' => $no_hp,
'id_hak_akses' => $id_hak_akses
);
$this->db->where('id_user', $id_user);
$query = $this->db->update('tb_user', $data);
if ($query) {
    return true;
} else {
    return false;
}
}

function ubah_user_foto_dm($id_user,$pn, $nama, $jenis_kelamin, $format, $alamat, $jabatan, $no_hp,
$foto_1) {
    $data = array(
        'nama' => $nama,
        'pn' => $pn,
        'id_jenis_kelamin' => $jenis_kelamin,
        'alamat' => $alamat,
        'tgl_lahir' => $format,
        'jabatan' => $jabatan,
        'foto' => $foto_1,
        'no_hp' => $no_hp
    );
    $this->db->where('id_user', $id_user);
    $query = $this->db->update('tb_user', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function check_foto($id_user) {
    $query = $this->db->query("select foto from tb_user where pn='\" . $id_user . \"'");
    return $query->row();
}

function ubah_user_tanpa_foto($id_user,$pn, $nama, $jenis_kelamin, $format, $alamat, $jabatan, $no_hp,
$id_hak_akses) {
    $data = array(
        'nama' => $nama,
        'pn' => $pn,
        'id_jenis_kelamin' => $jenis_kelamin,
        'alamat' => $alamat,
        'tgl_lahir' => $format,
        'jabatan' => $jabatan,
        'no_hp' => $no_hp,
        'id_hak_akses' => $id_hak_akses
    );
    $this->db->where('id_user', $id_user);
    $query = $this->db->update('tb_user', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}
```

```
}

function ubah_user_tanpa_foto_dm($id_user,$pn, $nama, $jenis_kelamin, $format, $alamat, $jabatan,
$no_hp) {
    $data = array(
        'nama' => $nama,
        'pn' => $pn,
        'id_jenis_kelamin' => $jenis_kelamin,
        'alamat' => $alamat,
        'tgl_lahir' => $format,
        'jabatan' => $jabatan,
        'no_hp' => $no_hp
    );
    $this->db->where('id_user', $id_user);
    $query = $this->db->update('tb_user', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function hapus_foto($id_user) {
    $query = $this->db->query("UPDATE tb_user SET foto = '' WHERE id_user = " . $id_user . "");
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function hapus_user($id_user) {
    $query = $this->db->delete('tb_user', array('id_user' => $id_user));
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ubah_account($id_user,$username,$password) {
    $data = array(
        'username' => $username,
        'password' => $password
    );
    $this->db->where('id_user', $id_user);
    $query = $this->db->update('tb_user', $data);
    if ($query) {
        return true;
    } else {
        return false;
    }
}

function ambil_jumlah_user(){
    $query = $this->db->query("SELECT count(id_user) as jumlah from tb_user");
    return $query->row();
}

function ambil_akun($id){
    $query = $this->db->query("SELECT * from tb_user where id_user='$id'");
    return $query->row();
}
```

```
}  
  
function combo_hak_akses(){  
    $query = $this->db->query("SELECT * from tb_hak_akses");  
    return $query->result();  
}  
  
}
```



E. Pengujian *Black Box*

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	<i>Login</i>	Menjalankan sistem	Menampilkan <i>Form Login</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi <i>username</i> dan <i>password</i> benar lalu menekan tombol “Login”	Menampilkan halaman dashboard sesuai level aktor masing-masing	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Tidak mengisi <i>username</i> atau <i>password</i> atau keduanya lalu menekan tombol “Login”	Menampilkan <i>message</i> “ <i>username</i> atau <i>password</i> harus diisi” di bawah masukan yang kosong	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi <i>username</i> dan <i>password</i> salah lalu menekan tombol “Login”	Menampilkan <i>alert</i> “Maaf! <i>username</i> atau <i>password</i> tidak ditemukan”	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2.	<i>Logout</i>	Menekan nama di bagian pojok kanan atas halaman lalu menekan tombol “Logout”	Menampilkan <i>form login</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
3.	Melihat Informasi Peta	Memilih menu <i>dashboard</i>	Menampilkan halaman <i>dashboard</i> yang berisi peta kecamatan di Kabupaten Jember dan semua kategori informasi peta	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		Memilih kecamatan dan kategori yang akan ditampilkan ke dalam peta	Menampilkan peta beserta informasinya sesuai dengan kecamatan dan kategori yang dipilih	[] Berhasil [] Gagal
	Melihat Detail Informasi Peta	Menekan salah satu <i>marker icon</i> , area atau garis pada peta	Menampilkan info <i>window</i> pada peta sesuai detail yang ingin dilihat	[] Berhasil [] Gagal
		Menekan tombol " <i>close</i> " pada info <i>window</i>	Menampilkan peta beserta informasinya	[] Berhasil [] Gagal
4.	Melihat Data <i>User</i>	Memilih menu " <i>User</i> "	Menampilkan data <i>user</i> pada tabel di halaman data <i>user</i>	[] Berhasil [] Gagal
	Melihat Detail <i>User</i>	Menekan tombol " <i>Detail User</i> " pada baris data yang ingin dilihat detailnya	Menampilkan detail <i>user</i> pada halaman detail <i>user</i>	[] Berhasil [] Gagal
		Menekan tombol " <i>Back</i> "	Menampilkan data <i>user</i> pada tabel di halaman data <i>user</i>	[] Berhasil [] Gagal
	Menambah Data <i>User</i>	Menekan tombol " <i>Tambah User</i> "	Menampilkan <i>form</i> isian data <i>user</i>	[] Berhasil [] Gagal
		Mengisi seluruh isian <i>form</i> data <i>user</i> dan lengkap dan benar lalu menekan tombol " <i>Save</i> "	Menyimpan data ke <i>database</i> dan Menampilkan <i>alert</i> " <i>Selamat! Data user sukses disimpan</i> " pada halaman data <i>user</i>	[] Berhasil [] Gagal
		Mengisi isian <i>form</i> data <i>user</i> tidak lengkap lalu menekan tombol " <i>Save</i> "	Menampilkan <i>message</i> " <i>masukan harus diisi</i> " di bawah <i>field</i> masukan yang kosong	[] Berhasil [] Gagal
		Mengisi masukan angka pada <i>form</i> data <i>user</i>	Menampilkan <i>message</i> " <i>input harus berupa angka</i> " di bawah <i>field</i> masukan yang salah	[] Berhasil [] Gagal

	dengan huruf lalu menekan tombol "Save"		
	Mengisi masukan <i>personal number</i> yang sama dengan data yang ada pada <i>database</i> lalu menekan tombol "Save"	Menampilkan <i>alert "Personal number duplicate entry"</i> pada <i>form</i> isian data <i>user</i>	[] Berhasil [] Gagal
	Menekan tombol "Clear" pada <i>form</i> isian data <i>user</i>	Menampilkan <i>form</i> isian data <i>user</i> yang sudah dikosongkan	[] Berhasil [] Gagal
	Menekan tombol "Back" pada <i>form</i> isian data <i>user</i>	Menampilkan data <i>user</i> pada tabel di halaman data <i>user</i>	[] Berhasil [] Gagal
Mengubah Data User	Menekan tombol "Ubah User" pada baris data yang akan diubah	Menampilkan data <i>user</i> ke dalam <i>form</i> ubah data <i>user</i>	[] Berhasil [] Gagal
	Mengubah isian <i>form</i> ubah data <i>user</i> dengan lengkap dan benar lalu menekan tombol "Save Edit"	Mengubah data di <i>database</i> lalu menampilkan <i>alert "Selamat! Data user sukses diubah"</i> pada halaman data <i>user</i>	[] Berhasil [] Gagal
	Mengisi isian <i>form</i> data <i>user</i> tidak lengkap lalu menekan tombol "Save Edit"	Menampilkan <i>message "masukan harus diisi"</i> di bawah <i>field</i> masukan yang kosong	[] Berhasil [] Gagal
	Mengisi masukan angka pada <i>form</i> data <i>user</i> dengan huruf lalu menekan tombol "Save Edit"	Menampilkan <i>message "input harus berupa angka"</i> di bawah <i>field</i> masukan yang salah	[] Berhasil [] Gagal

	Mengisi masukan <i>personal number</i> yang sama dengan data yang ada pada <i>database</i> lalu menekan tombol “ <i>Save Edit</i> ”	Menampilkan <i>alert</i> “ <i>Personal number duplicate entry</i> ” pada <i>form</i> isian data <i>user</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “ <i>Clear</i> ” pada <i>form</i> ubahan data <i>user</i>	Menampilkan <i>form</i> isian data <i>user</i> yang belum diubah	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “ <i>Back</i> ” pada <i>form</i> ubahan data <i>user</i>	Menampilkan data <i>user</i> pada tabel di halaman data <i>user</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “Hapus Foto” pada <i>form</i> ubahan data <i>user</i>	Menampilkan <i>alert</i> “Apakah Anda yakin akan menghapus foto ini?” disertai dengan tombol OK dan <i>Cancel</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “OK” pada <i>alert</i>	Menghapus foto dari data <i>user</i> di <i>database</i> lalu menampilkan <i>form</i> isian data <i>user</i> yang belum diubah	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “ <i>Cancel</i> ” pada <i>alert</i>	Menampilkan <i>form</i> isian data <i>user</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Menghapus Data User	Menekan tombol “Hapus User” pada baris data yang akan dihapus	Menampilkan <i>alert</i> “Apakah Anda yakin akan menghapus data ini?” disertai dengan tombol OK dan <i>Cancel</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “OK” pada <i>alert</i>	Menghapus data <i>user</i> yang dipilih dari <i>database</i> lalu menampilkan data <i>user</i> pada tabel di halaman data <i>user</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “ <i>Cancel</i> ” pada <i>alert</i>	Menampilkan data <i>user</i> pada halaman data <i>user</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

5.	Melihat Data Kecamatan	Memilih menu Lokasi lalu memilih submenu Data Kecamatan	Menampilkan data kecamatan pada tabel di halaman data kecamatan	[] Berhasil [] Gagal
	Melihat Detail Kecamatan	Menekan tombol "Detail Kecamatan" pada salah satu baris data	Menampilkan detail dan peta data kecamatan pada halaman detail data kecamatan	[] Berhasil [] Gagal
		Menekan tombol "Back"	Menampilkan data kecamatan pada tabel di halaman data kecamatan	[] Berhasil [] Gagal
Menambah Data Kecamatan	Menekan tombol "Tambah Kecamatan"		Menampilkan <i>form</i> isian tambah data kecamatan dan peta <i>digital</i>	[] Berhasil [] Gagal
	Mengisi isian form tambah data kecamatan, menekan <i>icon drawing polygon</i> pada peta, menggambar <i>polygon</i> baru lalu menekan 2 kali pada peta		Menampilkan hasil gambar pada <i>field</i> masukan "Hasil <i>Drawing</i> "	[] Berhasil [] Gagal
	Menekan tombol "Parse JSON"		Menampilkan koordinat dan tipe <i>polygon</i> yang telah dibuat pada <i>field</i> masukan "Geom"	[] Berhasil [] Gagal
	Menekan tombol "Clear Map"		Menghapus hasil <i>drawing polygon</i> pada peta dan menghapus hasil gambar pada <i>field</i> masukan "Hasil <i>Drawing</i> "	[] Berhasil [] Gagal
	Menekan tombol "Save"		Menyimpan data ke <i>database</i> lalu menampilkan alert "Selamat! Data kecamatan sukses disimpan" pada halaman data kecamatan	[] Berhasil [] Gagal

	Mengisi isian <i>form</i> tambah data kecamatan secara tidak lengkap lalu menekan tombol “ <i>Save</i> ”	Menampilkan <i>message</i> “masukan harus diisi” di bawah <i>field</i> masukan yang kosong	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Mengisi isian berupa masukan angka pada <i>form</i> tambah data kecamatan dengan huruf lalu menekan tombol “ <i>Save</i> ”	Menampilkan <i>message</i> “input harus berupa angka” di bawah <i>field</i> masukan yang salah	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “ <i>Clear</i> ” pada <i>form</i> ubah data kecamatan	Menampilkan <i>form</i> isian data kecamatan yang sudah dikosongkan	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “ <i>Back</i> ” pada <i>form</i> ubah data kecamatan	Menampilkan data kecamatan pada tabel di halaman data kecamatan	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Mengubah Data Kecamatan	Menekan tombol “Ubah Kecamatan” pada baris data yang akan diubah	Menampilkan <i>form</i> isian ubah data kecamatan dan peta	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Mengisi isian <i>form</i> ubah data kecamatan, menekan <i>icon drawing polygon</i> pada peta, menggambar <i>polygon</i> sesuai dengan perubahan area kecamatan, lalu klik 2 kali pada peta	Menampilkan hasil gambar pada <i>field</i> masukan “ <i>Hasil Drawing</i> ”	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “ <i>Parse JSON</i> ”	Menampilkan koordinat dan tipe <i>polygon</i> yang telah dibuat pada <i>field</i> masukan “ <i>Geom</i> ”	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Menekan tombol “ <i>Clear Map</i> ”	Menghapus hasil <i>drawing polygon</i> pada peta dan menghapus hasil gambar pada <i>field</i> masukan “ <i>Hasil Drawing</i> ”	[] Berhasil [] Gagal
Menekan tombol “ <i>Save Edit</i> ”	Mengubah data dari <i>database</i> lalu menampilkan alert “Selamat! Data kecamatan sukses diubah” Mengambil data kecamatan dari <i>database</i> lalu menampilkan data kecamatan pada tabel di halaman data kecamatan	[] Berhasil [] Gagal
Klik tombol “ <i>Back</i> ”	Mengambil data kecamatan dari <i>database</i> lalu menampilkan data kecamatan pada tabel di halaman data kecamatan	[] Berhasil [] Gagal
Mengisi seluruh isian <i>form</i> ubah data kecamatan tanpa mengubah koordinat area kecamatan lalu menekan tombol “ <i>Save Edit</i> ”	Mengubah data dari <i>database</i> , menampilkan alert “Selamat! Data kecamatan sukses diubah”, mengambil data kecamatan dari <i>database</i> lalu menampilkan data kecamatan pada tabel di halaman data kecamatan	[] Berhasil [] Gagal
Mengubah isian <i>form</i> ubah data kecamatan secara tidak lengkap lalu menekan tombol “ <i>Save Edit</i> ”	Menampilkan <i>message</i> “masukan harus diisi” di bawah <i>field</i> masukan yang kosong	[] Berhasil [] Gagal
Mengubah isian berupa ubahan angka pada <i>form</i> ubah data kecamatan dengan huruf lalu menekan tombol “ <i>Save</i> ”	Menampilkan <i>message</i> “input harus berupa angka” di bawah <i>field</i> masukan yang salah	[] Berhasil [] Gagal

<i>Edit</i>			
		Menekan tombol " <i>Clear</i> "	Menampilkan <i>form</i> isian data kecamatan yang sudah dikosongkan [] Berhasil [] Gagal
Menghapus Data Kecamatan		Menekan tombol "Hapus Kecamatan" pada baris data yang akan dihapus	Menampilkan <i>alert</i> "Apakah Anda yakin akan menghapus data ini?" disertai dengan tombol OK dan <i>Cancel</i> [] Berhasil [] Gagal
		Menekan tombol "OK" pada <i>alert</i>	Menghapus data kecamatan yang dipilih dari <i>database</i> lalu Menampilkan <i>alert</i> "Selamat! Data kecamatan sukses dihapus" lalu Mengambil data kecamatan dari <i>database</i> lalu Menampilkan data kecamatan pada tabel di halaman data kecamatan [] Berhasil [] Gagal
		Menekan tombol " <i>Cancel</i> " pada <i>alert</i>	Menampilkan data kecamatan pada tabel di halaman data kecamatan [] Berhasil [] Gagal
6.	Melihat Data Kelurahan	Memilih menu Lokasi lalu Memilih submenu Data Kelurahan	Mengambil data kelurahan dari <i>database</i> lalu menampilkan data kelurahan pada tabel di halaman data kelurahan [] Berhasil [] Gagal
	Melihat Detail Kelurahan	Menekan tombol "Detail Kelurahan" pada baris data yang akan dilihat detailnya	Mengambil data kelurahan yang dipilih dari <i>database</i> lalu menampilkan detail dan peta data kelurahan pada halaman detail data kelurahan [] Berhasil [] Gagal

	Klik tombol <i>back</i>	Mengambil data kelurahan dari <i>database</i> lalu menampilkan data kelurahan pada tabel di halaman data kelurahan	[] Berhasil [] Gagal
Menambah Data Kelurahan	Menekan tombol “Tambah Kelurahan”	Menampilkan <i>form</i> isian tambah data kelurahan dan peta	[] Berhasil [] Gagal
	Mengisi seluruh isian <i>form</i> tambah data kelurahan, Klik <i>icon drawing polygon</i> pada peta, menggambar <i>polygon</i> baru lalu klik 2 kali pada peta saat sudah selesai menggambar	Menampilkan hasil gambar pada <i>field</i> masukan “Hasil <i>Drawing</i> ”	[] Berhasil [] Gagal
	Menekan tombol “ <i>Parse JSON</i> ”	Mengambil koordinat dan tipe <i>polygon</i> lalu menampilkan koordinat dan tipe <i>polygon</i> yang telah dibuat pada <i>field</i> masukan “ <i>Geom</i> ”	[] Berhasil [] Gagal
	Menekan tombol “ <i>Save</i> ”	Menyimpan data ke <i>database</i> lalu menampilkan alert “Selamat! Data kelurahan sukses disimpan”, lalu Mengambil data kelurahan dari <i>database</i> kemudian Menampilkan data kelurahan pada tabel di halaman data kelurahan	[] Berhasil [] Gagal
	Mengisi isian <i>form</i> tambah data kelurahan secara tidak lengkap lalu menekan	Menampilkan <i>message</i> “masukan harus diisi” di bawah <i>field</i> masukan yang kosong	[] Berhasil [] Gagal

	tombol “ <i>Save</i> ”		
	Menekan tombol “ <i>Clear Map</i> ”	Menghapus hasil <i>drawing polygon</i> pada peta	[] Berhasil [] Gagal
	Menekan tombol “ <i>Clear</i> ”	Menampilkan <i>form</i> isian data kelurahan yang sudah dikosongkan	[] Berhasil [] Gagal
	Klik tombol “ <i>Back</i> ”	Mengambil data kelurahan dari <i>database</i> lalu menampilkan data kelurahan pada tabel di halaman data kelurahan	[] Berhasil [] Gagal
Mengubah Data Kelurahan	Menekan tombol “Ubah Kelurahan” pada baris data yang akan diubah	Mengambil data kelurahan yang dipilih dari <i>database</i> lalu Menampilkan <i>form</i> isian ubah data kelurahan dan peta	[] Berhasil [] Gagal
	Mengisi seluruh isian <i>form</i> ubah data kelurahan, Klik <i>icon drawing polygon</i> pada peta, Menggambar <i>polygon</i> sesuai dengan perubahan area kelurahan lalu Klik 2 kali pada peta jika sudah selesai menggambar	Menampilkan hasil gambar pada <i>field</i> masukan “ <i>Hasil Drawing</i> ”	[] Berhasil [] Gagal
	Menekan tombol “ <i>Parse JSON</i> ”	Mengambil koordinat dan tipe <i>polygon</i> lalu menampilkan koordinat dan tipe <i>polygon</i> yang telah dibuat pada <i>field</i>	[] Berhasil [] Gagal

masukan “Geom”		
Menekan tombol “Save Edit”	Mengubah data dari <i>database</i> lalu menampilkan alert “Selamat! Data kelurahan sukses diubah” lalu mengambil data kelurahan dari <i>database</i> , lalu menampilkan data kelurahan pada tabel di halaman data kelurahan	[] Berhasil [] Gagal
Mengisi seluruh isian <i>form</i> ubah data kelurahan tanpa mengubah koordinat area kelurahan lalu menekan tombol “Save Edit”	Mengubah data dari <i>database</i> lalu menampilkan alert “Selamat! Data kelurahan sukses diubah” lalu mengambil data kelurahan dari <i>database</i> , lalu menampilkan data kelurahan pada tabel di halaman data kelurahan	[] Berhasil [] Gagal
Mengubah isian <i>form</i> ubah data kelurahan secara tidak lengkap lalu menekan tombol “Save Edit”	Menampilkan <i>message</i> “masukan harus diisi” di bawah <i>field</i> masukan yang kosong	[] Berhasil [] Gagal
Menekan tombol “Clear Map”	Menghapus hasil <i>drawing polygon</i> pada peta	[] Berhasil [] Gagal
Menekan tombol “Clear”	Menampilkan <i>form</i> isian data kelurahan yang sudah dikosongkan	[] Berhasil [] Gagal
Klik tombol “Back”	Mengambil data kelurahan dari <i>database</i> lalu menampilkan data kelurahan pada tabel di halaman data kelurahan	[] Berhasil [] Gagal

	Menghapus Data Kelurahan	Menekan tombol “Hapus Kelurahan” pada baris data yang akan dihapus	Menampilkan <i>alert</i> “Apakah Anda yakin akan menghapus data ini?” disertai dengan tombol OK dan <i>Cancel</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Menekan tombol “OK” pada <i>alert</i>	Menghapus data kelurahan yang dipilih dari <i>database</i> , Menampilkan <i>alert</i> “Selamat! Data kelurahan sukses dihapus”, Mengambil data kelurahan dari <i>database</i> lalu Menampilkan data kelurahan pada tabel di halaman data kelurahan	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Menekan tombol “ <i>Cancel</i> ” pada <i>alert</i>	Menampilkan data kelurahan pada tabel di halaman data kelurahan	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
7.	Melihat Data Kriteria	Memilih menu Data Kriteria	Mengambil data kriteria dari <i>database</i> lalu menampilkan data kriteria pada tabel di halaman data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menambah Data Kriteria	Menekan tombol “Tambah Kriteria”	Menampilkan <i>form</i> isian data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi seluruh isian <i>form</i> data kriteria lalu menekan tombol “ <i>Save</i> ”	Menyimpan masukan ke <i>database</i> lalu menampilkan <i>alert</i> “Selamat! Data kriteria sukses disimpan” lalu Mengambil data kriteria dari <i>database</i> lalu menampilkan data kriteria pada tabel di halaman data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi seluruh isian <i>form</i> data kriteria secara tidak lengkap lalu menekan	Menampilkan <i>message</i> “masukan harus diisi” di bawah <i>field</i> masukan yang kosong	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

tombol “ <i>Save</i> ”			
Mengubah Data Kriteria	Menekan tombol “Ubah Kriteria” pada baris data yang akan diubah	Mengambil data kriteria yang dipilih dari <i>database</i> lalu menampilkan data kriteria ke dalam <i>form</i> ubah data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Mengubah isian <i>form</i> ubah data kriteria lalu menekan tombol “ <i>Save Edit</i> ”	Menyimpan data yang diubah ke <i>database</i> , menampilkan <i>alert</i> “Selamat! Data kriteria sukses diubah”, mengambil data kriteria dari <i>database</i> lalu menampilkan data kriteria pada tabel di halaman data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Mengubah isian <i>form</i> ubah data kriteria secara tidak lengkap lalu menekan tombol “ <i>Save Edit</i> ”	Menampilkan <i>message</i> “masukan harus diisi” di bawah <i>field</i> ubahan yang kosong	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Menekan tombol “ <i>Clear</i> ”	Menampilkan <i>form</i> isian data kriteria yang belum diubah	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Klik tombol “ <i>Back</i> ”	Mengambil data kriteria dari <i>database</i> lalu menampilkan data kriteria pada tabel di halaman data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Menghapus Data Kriteria	Menekan tombol “Hapus Kriteria” pada baris data yang akan dihapus	Menampilkan <i>alert</i> “Apakah Anda yakin akan menghapus data ini?” disertai dengan tombol OK dan <i>Cancel</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

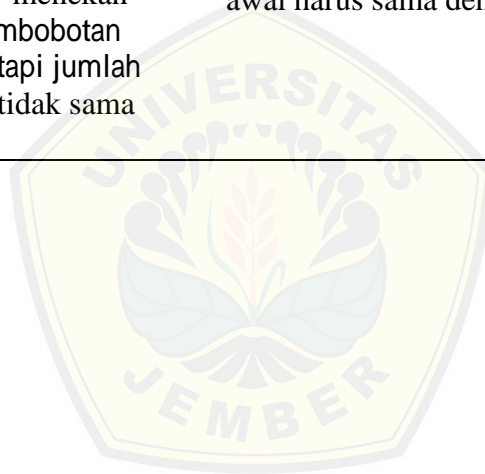
		Menekan tombol “OK” pada <i>alert</i>	Menghapus data kriteria yang dipilih dari <i>database</i> lalu Menampilkan <i>alert</i> “Selamat! Data kriteria sukses dihapus”, Mengambil data kriteria dari <i>database</i> lalu menampilkan data kriteria pada tabel di halaman data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Menekan tombol “Cancel” pada <i>alert</i>	Menampilkan data kriteria pada halaman data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8.	Melihat Batasan Kriteria	Memilih menu Data Kriteria	Mengambil data kriteria dari <i>database</i> lalu menampilkan data kriteria pada tabel di halaman data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Menekan tombol “Batasan Kriteria” pada salah satu baris data kriteria yang dipilih	Memeriksa ada atau tidaknya batasan kriteria dari kriteria yang dipilih pada <i>database</i> , Terdapat batasan kriteria dari kriteria yang dipilih, Mengambil batasan kriteria yang sudah ada di <i>database</i> berdasarkan kriteria yang dipilih, lalu Menampilkan batasan kriteria pada tabel di halaman batasan kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol “Back”	Mengambil data kriteria dari <i>database</i> lalu menampilkan data kriteria pada tabel di halaman data kriteria	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Mengubah Batasan Kriteria	Menekan tombol “Ubah Batasan Kriteria” pada	Mengambil batasan kriteria yang dipilih dari <i>database</i> lalu menampilkan batasan	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

	baris data yang akan diubah	kriteria ke dalam <i>form</i> ubah batasan kriteria	
	Mengubah isian <i>form</i> ubah batasan kriteria lalu menekan tombol “ <i>Save Edit</i> ”	Menyimpan data yang diubah ke <i>database</i> lalu menampilkan <i>alert</i> “Selamat! Batasan kriteria sukses diubah” , mengambil batasan kriteria dari <i>database</i> lalu menampilkan batasan kriteria pada tabel di halaman batasan kriteria	[] Berhasil [] Gagal
	Mengisi isian <i>form</i> ubah batasan kriteria secara tidak lengkap lalu menekan tombol “ <i>Save Edit</i> ”	Menampilkan <i>message</i> “masukan harus diisi” di bawah <i>field</i> masukan yang kosong	[] Berhasil [] Gagal
	Mengisi isian berupa masukan angka pada <i>form</i> ubah batasan kriteria dengan huruf lalu menekan tombol “ <i>Save Edit</i> ”	Menampilkan <i>message</i> “input harus berupa angka” di bawah <i>field</i> masukan yang salah	[] Berhasil [] Gagal
	Menekan tombol “ <i>Clear</i> ”	Menampilkan <i>form</i> isian batasan kriteria yang belum diubah	[] Berhasil [] Gagal
	Klik tombol “ <i>Back</i> ”	Mengambil batasan kriteria dari <i>database</i> lalu menampilkan batasan kriteria pada tabel di halaman batasan kriteria	[] Berhasil [] Gagal
Menambah Batasan Kriteria	Menekan tombol “Batasan Kriteria” pada salah satu	Memeriksa apakah ada data batasan kriteria pada kriteria yang dipilih dari	[] Berhasil [] Gagal

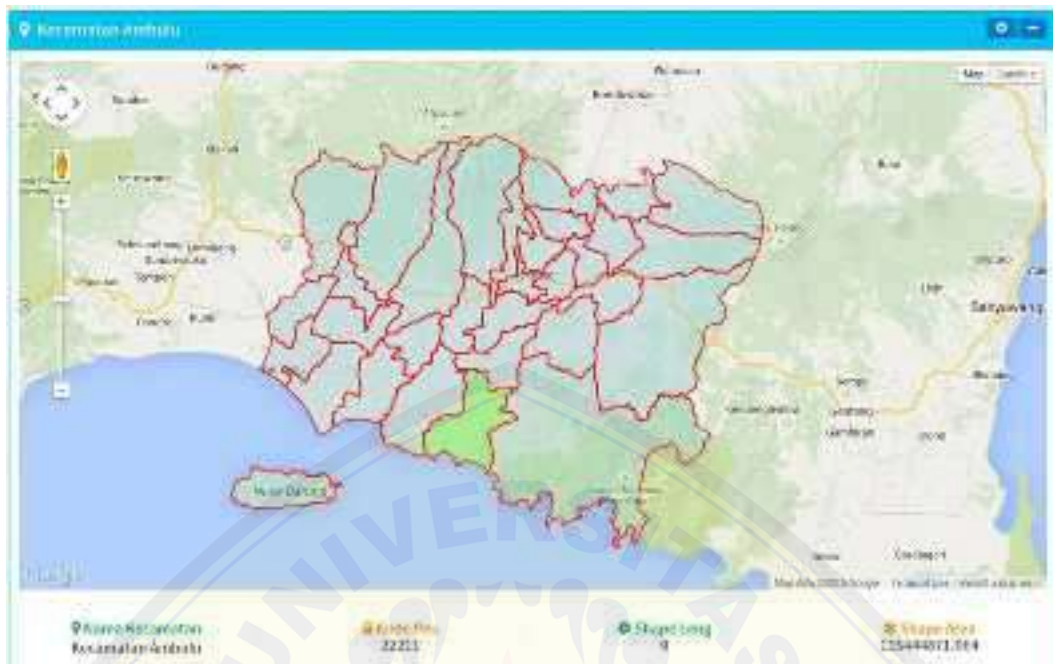
		baris data kriteria yang dipilih	<i>database</i> lalu Tidak terdapat batasan kriteria pada kriteria yang dipilih lalu menampilkan <i>form</i> isian batasan kriteria,	
		Mengisi seluruh isian <i>form</i> batasan kriteria lalu Menekan tombol “Save All”	Menyimpan masukan ke <i>database</i> , menampilkan <i>alert</i> “Selamat! Batasan kriteria sukses disimpan”, Mengambil batasan kriteria dari <i>database</i> , lalu menampilkan batasan kriteria pada tabel di halaman batasan kriteria	[] Berhasil [] Gagal
		Mengisi isian <i>form</i> tambah batasan kriteria secara tidak lengkap lalu menekan tombol “Save”	Menampilkan <i>message</i> “masukan harus diisi” di bawah <i>field</i> masukan yang kosong	[] Berhasil [] Gagal
		Mengisi isian berupa masukan angka pada <i>form</i> tambah batasan kriteria dengan huruf lalu menekan tombol “Save”	Menampilkan <i>message</i> “input harus berupa angka” di bawah <i>field</i> masukan yang salah	[] Berhasil [] Gagal
		Menekan tombol “Clear”	Menampilkan <i>form</i> isian batasan kriteria yang sudah dikosongkan	[] Berhasil [] Gagal
		Klik tombol “Back”	Mengambil data kriteria dari <i>database</i> lalu menampilkan data kriteria pada tabel di halaman data kriteria	[] Berhasil [] Gagal
9.	Memroses Pengambilan Keputusan	Menekan tombol “Proses DSS”	Mengambil data kriteria dari <i>database</i> lalu menampilkan data kriteria, <i>field</i> masukan bobot awal kriteria pada tabel di halaman pilih kriteria dan tombol “Tambah Kriteria”	[] Berhasil [] Gagal

Pilih kriteria yang akan dipakai dalam pengambilan keputusan	Mengaktifkan <i>field</i> masukan bobot awal kriteria yang dipilih agar bisa diisi oleh <i>decision maker</i>	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Mengisi bobot awal kriteria lalu menekan tombol “Pembobotan Entropy”	Menghitung perhitungan metode <i>Entropy</i> , Menyimpan bobot akhir kriteria ke <i>database</i> , lalu menampilkan metrik kriteria, metrik kriteria ternormalisasi, metrik probabilitas kriteria, metrik natural logaritma probabilitas kriteria, hasil pengukuran entropy setiap kriteria, hasil perhitungan bobot entropy setiap kriteria dan hasil pembobotan entropy pada halaman pembobotan entropy	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Menekan tombol “Perankingan VIKOR”	Menghitung perhitungan metode VIKOR, Menyimpan hasil perankingan VIKOR ke <i>database</i> , lalu menampilkan metrik kriteria ternormalisasi, metrik solusi ideal positif negatif setiap kriteria, metrik solusi ideal terbobot ,metrik nilai S,R dan Q, hasil <i>decreasing order</i> dari nilai S,R dan Q, serta hasil perankingan VIKOR pada halaman perankingan VIKOR	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Menekan tombol “Analisa Kelompok”	Menghitung perhitungan metode Borda, menyimpan hasil perhitungan Borda ke <i>database</i> , lalu menampilkan metrik ranking kelurahan dari para pengambil keputusan, metrik <i>point</i> Borda, hasil Borda <i>count</i> setiap kelurahan, dan hasil	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

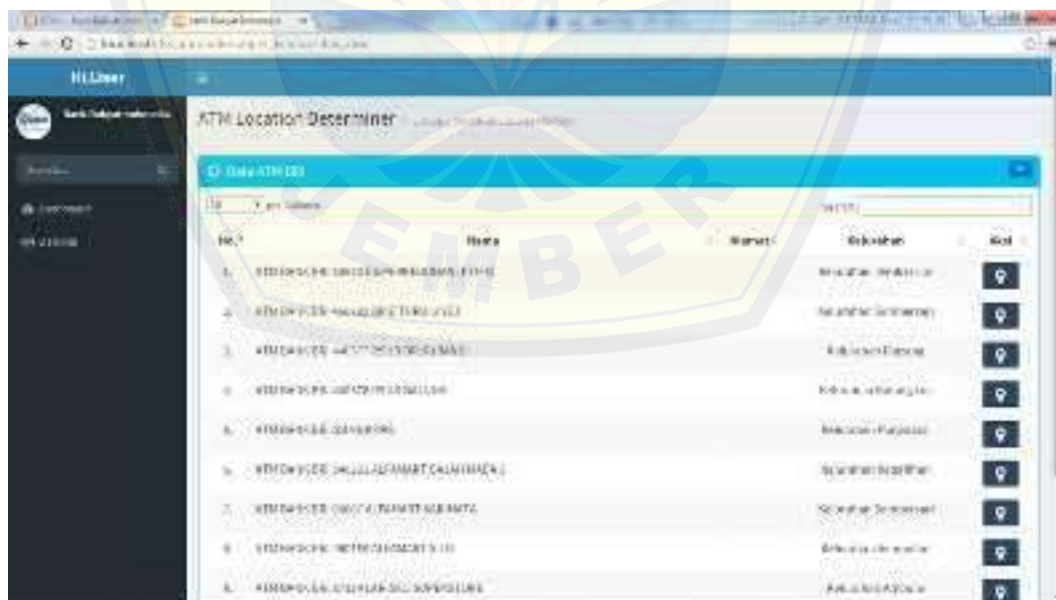
	rekomendasi lokasi penempatan ATM pada halaman analisa kelompok	
Menekan tombol “Lihat Hasil Analisa”	Mengambil data pengambilan keputusan yang hanya dilakukan oleh DM dari <i>database</i> , lalu menampilkan data pengambilan keputusan pada tabel di halaman data pengambilan keputusan	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Mengisi bobot awal kriteria lalu menekan tombol “Pembobotan Entropy” tetapi jumlah bobot awal tidak sama dengan 1	Menampilkan message “Jumlah bobot awal harus sama dengan 1”	<input type="checkbox"/> Berhasil <input type="checkbox"/> Gagal



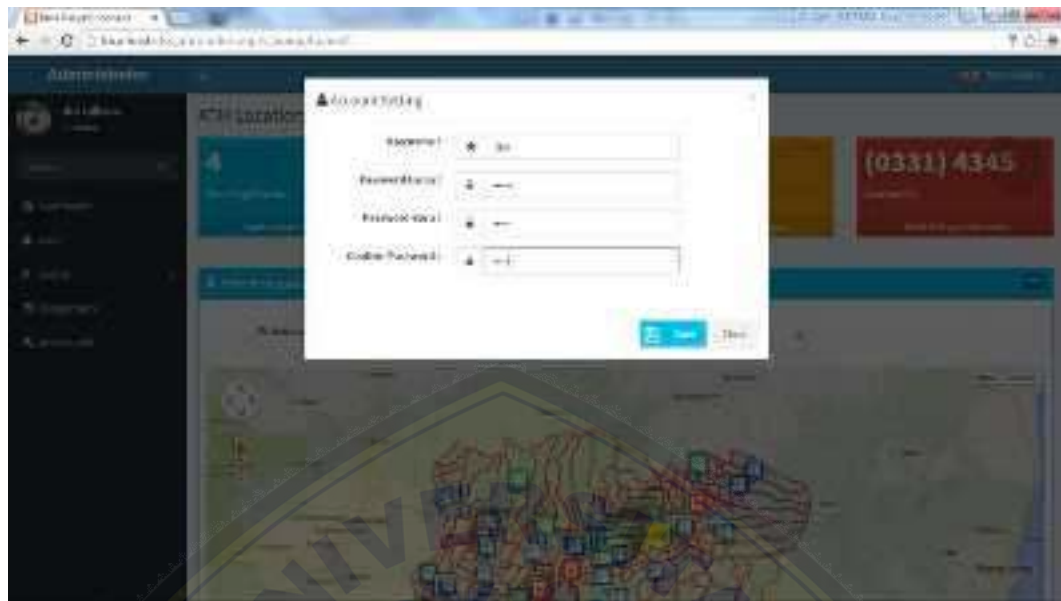
F.3 Tampilan Fitur Melihat Detail Kecamatan



F.4 Tampilan Fitur Melihat Data ATM BRI



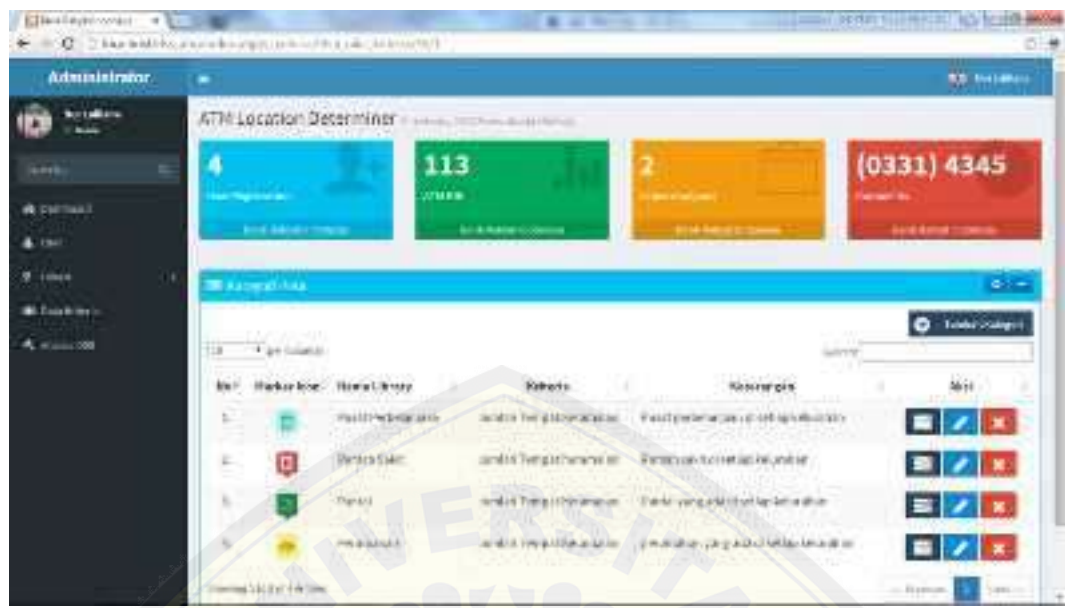
F.5 Tampilan Fitur Mengubah Akun



F.6 Tampilan Fitur Mengelola Nilai Kriteria



F.7 Tampilan Fitur Mengelola Kategori Peta



F.8 Tampilan Fitur Mengelola Nilai Kategori Peta



F.9 Tampilan Fitur Melihat Detail Analisa DSS



F.10 Tampilan Fitur Logout

