



**SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PRIORITAS  
PERBAIKAN JALAN KABUPATEN JEMBER MENGGUNAKAN  
METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)  
BERBASIS WEB**

**SKRIPSI**

oleh :

**Guntur Gibran**

**112410101089**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2015**



**SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PRIORITAS  
PERBAIKAN JALAN KABUPATEN JEMBER MENGGUNAKAN  
METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)  
BERBASIS WEB**

**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 :

**Guntur Gibran**

**112410101089**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2015**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Kedua Orangtua saya, Ayahanda Elis Bagianto dan Ibunda Sri Endah Indrawati;
2. Saudara-saudaraku beserta seluruh keluarga besar;
3. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
4. Seluruh teman-teman yang selalu memberikan bantuan dan dukungan;
5. Almamater Program Studi Sistem Informasi Universitas Jember.

**MOTO**

*“Berdo’a, tawakal, dan berikhtiar Allah bersama kita”*

*“Saya datang, saya bimbingan, saya ujian, saya revisi dan saya menang”*



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Guntur Gibran

NIM : 112410101089

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pendukung Keputusan Penentuan Prioritas Perbaikan Jalan Kabupaten Jember Menggunakan Metode *Simple Additive Weighting* (SAW) Berbasis Web”, 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, Desember 2015

Yang menyatakan,

Guntur Gibran

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Sistem Pendukung Keputusan Penentuan Prioritas Perbaikan Jalan Kabupaten Jember Menggunakan Metode *Simple Additive Weighting* (SAW) Berbasis Web”, telah diuji dan disahkan pada :

hari, tanggal : 28 Desember 2015

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Anang Andrianto ST., MT.  
NIP 196906151997021002

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

**SKRIPSI**

**SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PRIORITAS  
PERBAIKAN JALAN KABUPATEN JEMBER MENGGUNAKAN  
METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)  
BERBASIS WEB**

oleh :

Guntur Gibran

112410101089

Pembimbing :

Dosen Pembimbing Utama : Anang Andrianto ST., MT.  
Dosen Pembimbing Pendamping : Muhamad Arief Hidayat S.Kom., M.Kom.

**PENGESAHAN**

Skripsi berjudul “Sistem Pendukung Keputusan Penentuan Prioritas Perbaikan Jalan Kabupaten Jember Menggunakan Metode *Simple Additive Weighting* (SAW) Berbasis Web”, telah diuji dan disahkan pada :

hari, tanggal : 28 Desember 2015

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji:

Penguji I,

Penguji II,

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

Nelly Oktavia A, S.Si., MT.  
NIP 198410242009122008

Mengesahkan  
Ketua Program Studi,

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



## RINGKASAN

**Sistem Pendukung Keputusan Penentuan Prioritas Perbaikan Jalan Kabupaten Jember Menggunakan Metode *Simple Additive Weighting* (SAW) Berbasis Web;** Guntur Gibran, 112410101089; 2015; 216 halaman; Program Studi Sistem Informasi Universitas Jember.

Dinas Pekerjaan Umum Bina Marga Kabupaten Jember memiliki tanggungjawab dalam melakukan perencanaan dan pemeliharaan infrastruktur di wilayah Kabupaten Jember. Pelaksanaan perencanaan dan pemeliharaan infrastruktur masih menemukan banyak kendala salah satunya adalah keterbatasan dana dari pemerintah pusat karena tidak dapat menangani semua kebutuhan perbaikan jalan. Dalam penentuan prioritas perbaikan jalan, dinas terkait terkadang sering salah menentukan ruas-ruas jalan yang membutuhkan perbaikan sehingga banyak jalan-jalan yang seharusnya menjadi prioritas justru tidak mendapatkan perhatian. Oleh karena itu pada penelitian ini dibuat sistem yang mampu menangani permasalahan tersebut.

Sistem pendukung keputusan penentuan prioritas perbaikan jalan Kabupaten Jember yang dibangun dalam penelitian ini merupakan sistem yang mampu menampilkan urutan jalan yang didahulukan untuk diperbaiki berdasarkan analisis data kriteria lalulintas harian rata-rata, kondisi jalan, laporan masyarakat yang diimplementasikan dengan menggunakan metode SAW. Adapun subkriteria dari kondisi jalan antara lain tipe keretakan, lebar keretakan, luas kerusakan, alur, tambalan, lubang, kekasaran permukaan, amblas. Implementasi metode SAW dalam sistem terdapat pada fitur analisis prioritas, sehingga dinas PU Bina Marga Kabupaten Jember dapat dengan mudah mengetahui ruas jalan mana pada setiap up wilayah masing-masing yang menjadi prioritas untuk diperbaiki.

## PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Sistem Pendukung Keputusan Penentuan Prioritas Perbaikan Jalan Kabupaten Jember Menggunakan Metode *Simple Additive Weighting* (SAW) Berbasis Web”. 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. Ibu saya Sri Endah Indrawati, ibu paling super didunia dan ayah saya Elis Bagianto yang senantiasa memberikan motivasi, memberi arahan bimbingan dan semangat yang tak henti-hentinya kepada saya.
2. Keluarga besar saya yang senantiasa mensupport baik materi dan non materi.
3. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
4. Anang Andrianto, ST., MT selaku Dosen Pembimbing Utama dan Muhamad Arief Hidayat, S.Kom., M.Kom selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
5. Muhamad Arief Hidayat, S.Kom., M.Kom selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
6. Para pegawai kantor DPU Bina Marga Kabupaten Jember (Mas Horis, Pak Deni dan Pak Subandi) yang mengarahkan dan membantu penelitian saya.
7. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
8. Para sahabat-sahabat saya Dulur Jember (Resha, Amris, Dany, Indra, Yogi, Firdaus, Rizka Ndut, Rizqa Ceng, Qilba), Suwono Group (Mas Nopex, Resa,

Yanggi, Rian, Bayu, Dimas, Rama, Mas Dedy, Mas Daus, Mas Andre, Mas Fuad, Mas Lukman).

9. Para NEFOTION mania yang senantiasa menjadi bagian dalam masa-masa perkuliahan di Universitas Jember, senantiasa menjadi bagian dari keluarga selama saya hidup di Jember sebagai perantau.
10. Semua pihak yang tidak dapat disebutkan satu per satu

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, 28 Desember 2015

Penulis

**DAFTAR ISI**

PERSEMBAHAN .....	ii
MOTO .....	iii
PERNYATAAN .....	iv
PRAKATA .....	ix
DAFTAR ISI .....	xi
DAFTAR TABEL .....	xiv
DAFTAR GAMBAR .....	xv
DAFTAR LAMPIRAN .....	xvii
BAB 1. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah .....	2
1.3. Tujuan .....	3
1.4. Batasan Masalah .....	3
1.5. Sistematika Penulisan .....	3
BAB 2. TINJAUAN PUSTAKA .....	5
2.1 Penelitian Terdahulu .....	5
2.2 Sistem Informasi .....	5
2.3 Sistem Pendukung Keputusan .....	6
2.4 Jalan .....	7
2.4.1 Klasifikasi Jalan Menurut Fungsinya .....	7
2.4.2 Klasifikasi Jalan Menurut Statusnya .....	8
2.5 Metode <i>Simple Additive Weighting</i> .....	8
2.5.1 Pengertian Metode <i>Simple Additive Weighting</i> (SAW) .....	9
2.5.2 Langkah-langkah Metode <i>Simple Additive Weighting</i> (SAW) .....	10
2.6 Model <i>Waterfall</i> .....	10
BAB 3. METODOLOGI PENELITIAN .....	15

3.1	Jenis Penelitian .....	15
3.2	Tempat dan Waktu Penelitian .....	15
3.3	Tahapan Penelitian .....	15
3.3.1	Teknik Pengumpulan Data .....	16
3.3.2	Tahap Analisis Data .....	17
3.3.3	Tahap Pengembangan Sistem .....	18
<b>BAB 4. PENGEMBANGAN SISTEM.....</b>		<b>19</b>
4.1	Deskripsi Umum Sistem.....	19
4.1.1	SOP ( <i>statement of purpose</i> ) Sistem .....	19
4.2	Analisis Kebutuhan .....	20
4.3	Desain Sistem .....	21
4.3.1	<i>Bussiness Process</i> .....	21
4.3.2	<i>Usecase Diagram</i> .....	22
4.3.3	<i>Scenario</i> .....	25
4.3.4	<i>Activity diagram</i> .....	29
4.3.5	<i>Sequence diagram</i> .....	33
4.3.6	<i>Class diagram</i> .....	39
4.3.7	<i>Entity Relationship Diagram ( ERD )</i> .....	39
4.4	Penulisan Kode Program .....	40
4.5	Pengujian Sistem .....	41
4.5.1	<i>White Box Testing</i> .....	41
4.5.2	<i>Black Box Testing</i> .....	50
<b>BAB 5. HASIL DAN PEMBAHASAN.....</b>		<b>54</b>
5.1	Hasil Pembuatan Sistem.....	54
5.1.1	Tampilan utama sistem .....	54
5.1.2	Tampilan fitur menambahkan laporan masyarakat .....	55
5.1.3	Tampilan <i>login</i> sistem.....	56
5.1.4	Tampilan fitur mengelola data user.....	56

5.1.5	Tampilan fitur mengelola data jalan .....	58
5.1.6	Tampilan fitur mengelola data survei .....	60
5.1.7	Tampilan fitur mengelola data survei detail.....	64
5.1.8	Tampilan fitur melihat laporan masyarakat .....	65
5.1.9	Tampilan fitur mengubah bobot kriteria .....	66
5.1.10	Tampilan fitur mengubah bobot subkriteria.....	66
5.1.11	Tampilan fitur mengubah bobot nilai subkriteria.....	67
5.1.12	Tampilan fitur analisis prioritas jalan .....	68
5.2	Pembahasan .....	70
5.2.1	Implementasi Metode <i>Simple Additive Weighting</i> (SAW) .....	70
5.2.2	Pembahasan SPK Penentuan Prioritas Perbaikan Jalan .....	94
BAB 6. PENUTUP .....		96
6.1	Kesimpulan.....	96
6.2	Saran .....	97
DAFTAR PUSTAKA .....		98
LAMPIRAN .....		99



**DAFTAR TABEL**

Tabel 2.1 Tabel pengujian <i>black box</i> .....	14
Tabel 4.1 Definisi <i>usecase</i> .....	23
Tabel 4.2 Definisi Aktor .....	25
Tabel 4.3 Skenario tambah data user .....	25
Tabel 4.4 Skenario ubah data user .....	26
Tabel 4.5 Skenario hapus data user .....	27
Tabel 4.6 Skenario analisis prioritas perbaikan jalan.....	28
Tabel 4.7 <i>Test case</i> fitur mengelola data user .....	43
Tabel 4.8 <i>Test case</i> fitur mengelola data jalan .....	45
Tabel 4.9 <i>Test case</i> fitur analisis prioritas jalan.....	48
Tabel 4.10 <i>Test case</i> fitur menambahkan data laporan masyarakat.....	50
Tabel 4.11 Pengujian <i>black box</i> fitur mengelola data user .....	51
Tabel 4.12 Pengujian <i>black box</i> fitur mengelola data jalan .....	52
Tabel 5.1 Data Kriteria.....	70
Tabel 5.2 Data Subkriteria .....	70
Tabel 5.3 Subkriteria 1 LHR.....	71
Tabel 5.4 Subkriteria 2 tipe retak .....	72
Tabel 5.5 Subkriteria 3 lebar retak .....	72
Tabel 5.6 Subkriteria 4 luas kerusakan .....	73
Tabel 5.7 Subkriteria 5 kedalaman alur .....	73
Tabel 5.8 Subkriteria 6 Luas tambalan .....	73
Tabel 5.9 Subkriteria 7 luas lubang .....	74
Tabel 5.10 Subkriteria 8 kekasaran permukaan .....	74
Tabel 5.11 Subkriteria 9 Amblas .....	75
Tabel 5.12 Subkriteria 10 laporan masyarakat.....	75
Tabel 5.13 Tabel Sampel Alternatif .....	76
Tabel 5.14 Penilaian Alternatif .....	76
Tabel 5.15 Matrik Rating Kinerja .....	77
Tabel 5.16 Data bobot W .....	85
Tabel 5.17 Hasil Nilai Preferensi .....	86
Tabel 5.18 Perbandingan hasil nilai preferensi .....	93

## DAFTAR GAMBAR

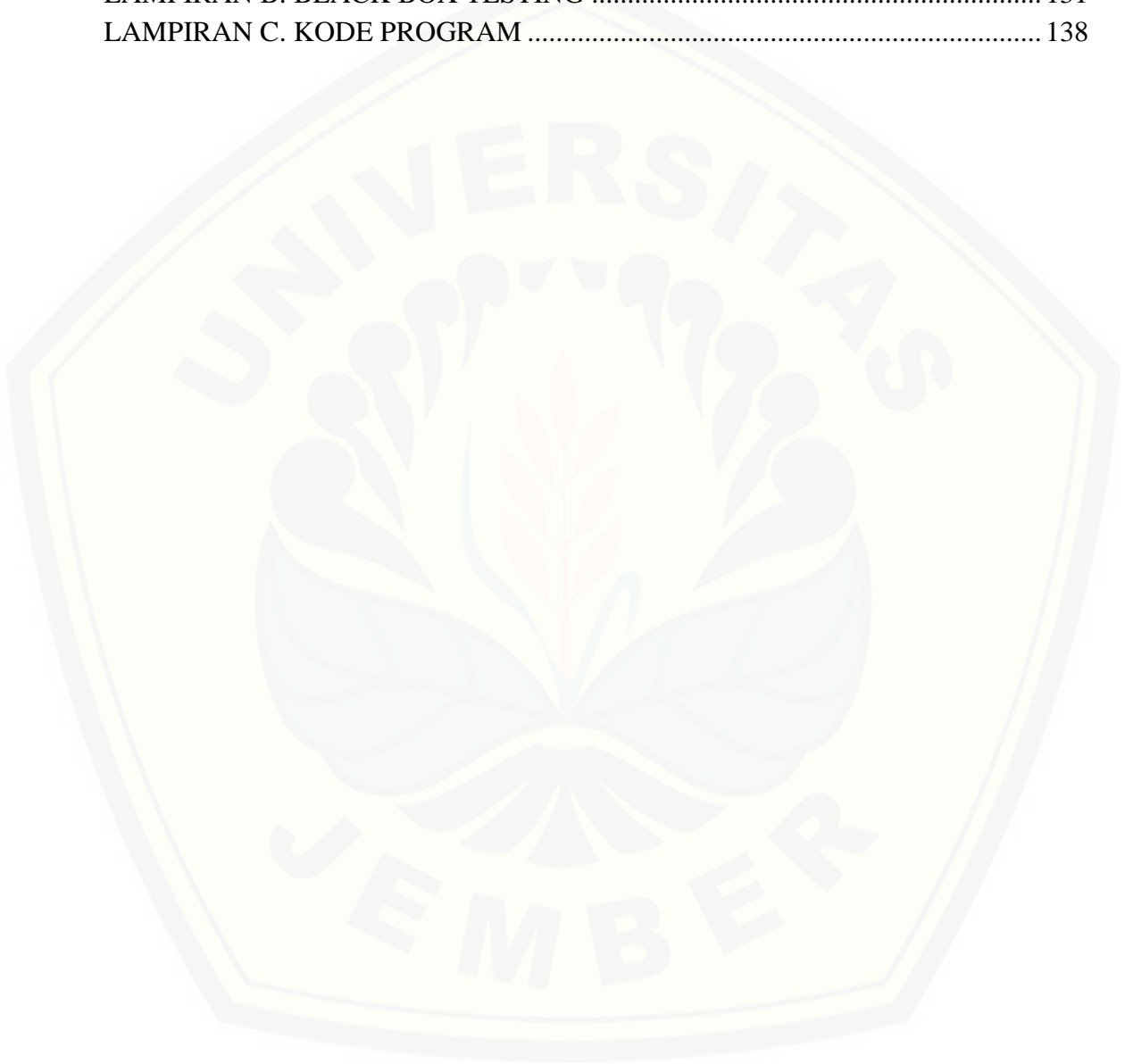
Gambar 2.1 Alur model <i>waterfall</i> .....	11
Gambar 3.1 Diagram Alir Penelitian (Sumber:Hasil Analisis, 2015).....	16
Gambar 3.2 <i>Flowchart</i> metode SAW pada sistem.....	17
Gambar 4.1 <i>Business process</i> sistem .....	21
Gambar 4.2 <i>Usecase diagram</i> sistem.....	22
Gambar 4.3 <i>Activity diagram</i> tambah data user .....	30
Gambar 4.4 <i>Activity diagram</i> ubah data user .....	31
Gambar 4.5 <i>Activity diagram</i> hapus data user .....	32
Gambar 4.6 <i>Activity diagram</i> analisis prioritas perbaikan jalan .....	33
Gambar 4.7 <i>boundary</i> .....	34
Gambar 4.8 <i>Control</i> .....	34
Gambar 4.9 Entity .....	35
Gambar 4.10 <i>Sequence diagram</i> tambah data user .....	35
Gambar 4.11 <i>Sequence diagram</i> ubah data user .....	36
Gambar 4.12 <i>Sequence diagram</i> hapus data user.....	37
Gambar 4.13 <i>Sequence diagram</i> analisis prioritas perbaikan jalan .....	38
Gambar 4.14 <i>class diagram</i> sistem .....	39
Gambar 4.15 ERD Sistem.....	40
Gambar 4.16 <i>Listing</i> program tambah data user .....	41
Gambar 4.17 <i>Listing</i> program ubah data user .....	42
Gambar 4.18 <i>Listing</i> program hapus data user.....	42
Gambar 4.19 Diagram alir fitur mengelola data user.....	42
Gambar 4.20 <i>Listing</i> program tambah data jalan .....	44
Gambar 4.21 <i>Listing</i> program ubah data jalan .....	44
Gambar 4.22 <i>Listing</i> program hapus data jalan .....	44
Gambar 4.23 Diagram alir fitur mengelola data jalan.....	45
Gambar 4.24 <i>Listing</i> program <i>function</i> ambil_data_surveidetail() .....	46
Gambar 4.25 <i>Listing</i> program <i>function</i> HitungMatrikRatingKinerja().....	47
Gambar 4.26 Diagram alir fitur analisis prioritas jalan.....	48
Gambar 4.27 <i>Listing</i> program fitur menambahkan data laporan masyarakat .....	49
Gambar 4.28 Diagram alir fitur menambahkan data laporan masyarakat .....	50
Gambar 5.1 Tampilan utama sistem.....	55
Gambar 5.2 Tampilan menu laporkan.....	55
Gambar 5.3 Tampilan halaman <i>login</i> .....	56



Gambar 5.4 Tampilan fitur tambah data <i>user</i> .....	57
Gambar 5.5 Tampilan fitur ubah data <i>user</i> .....	57
Gambar 5.6 Tampilan fitur hapus data <i>user</i> .....	58
Gambar 5.7 Tampilan fitur tambah data jalan .....	59
Gambar 5.8 Tampilan fitur ubah data jalan .....	59
Gambar 5.9 Tampilan fitur hapus data jalan.....	60
Gambar 5.10 Tampilan fitur tambah data survei .....	61
Gambar 5.11 Tampilan fitur ubah data survei .....	62
Gambar 5.12 Tampilan fitur hapus data survei.....	62
Gambar 5.13 Tampilan tabel kerusakan periodik .....	63
Gambar 5.14 Grafik tingkat kerusakan .....	63
Gambar 5.15 Tampilan fitur tambah data survei detail.....	64
Gambar 5.16 Tampilan fitur hapus data survei detail .....	65
Gambar 5.17 Tampilan fitur melihat laporan masyarakat .....	65
Gambar 5.18 Tampilan fitur mengubah bobot kriteria .....	66
Gambar 5.19 Tampilan fitur mengubah data subkriteria .....	67
Gambar 5.20 Tampilan fitur mengubah data bobot nilai subkriteria .....	68
Gambar 5.21 Tampilan fitur analisis prioritas jalan.....	69
Gambar 5.22 Tampilan hasil analisis .....	69
Gambar 5.23 Hasil nilai preferensi .....	93

**DAFTAR LAMPIRAN**

LAMPIRAN A. PERANCANGAN SISTEM.....	100
LAMPIRAN B. BLACK BOX TESTING .....	131
LAMPIRAN C. KODE PROGRAM .....	138



## **BAB 1. PENDAHULUAN**

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

### **1.1. Latar Belakang**

Semakin tinggi tingkat mobilitas masyarakat berbanding lurus dengan kebutuhan sarana dan prasarana yang mumpuni. Salah satu infrastruktur yang menjadi penunjang mobilitas utama masyarakat adalah prasarana jalan. Namun kenyataannya saat ini banyak kita jumpai kondisi jalan yang membutuhkan perbaikan. Ketika fasilitas tersebut rusak dan terlambat diperbaiki maka akan sangat berpengaruh terhadap kegiatan masyarakat. Sesuai dengan UU No. 38 Tahun 2004 pelaksanaan pembangunan jalan diserahkan kepada daerah dalam hal ini yaitu dinas pekerjaan umum. Dinas Pekerjaan Umum Bina Marga merupakan institusi pemerintah yang bertugas untuk pembinaan dan pelaksanaan tugas operasional kebinamargaan yang meliputi perencanaan, penyelenggaraan, pembinaan, pengendalian, pemeliharaan prasaran-prasarana infrastruktur seperti jalan, jembatan, gorong-gorong, trotoar, dan penerangan jalan yang berhubungan langsung dengan kegiatan masyarakat sehari-hari. Oleh sebab itu, merupakan tugas bagi Dinas Pekerjaan Umum Bina Marga untuk dapat mengelola ruas jalan-jalan yang rusak untuk diperbaiki.

Dinas Pekerjaan Umum Bina Marga Kabupaten Jember memiliki tanggungjawab dalam melakukan perencanaan dan pemeliharaan infrastruktur di wilayah Kabupaten Jember. Pelaksanaan perencanaan dan pemeliharaan infrastruktur masih menemukan banyak kendala salah satunya adalah keterbatasan dana dari pemerintah pusat karena tidak dapat menangani semua kebutuhan perbaikan jalan. Oleh karena itu, Dinas Pekerjaan Umum Bina Marga Kabupaten Jember dalam pelaksanaannya selama ini dinilai kurang obyektif. Perbaikan yang dilaksanakan tidak berdasarkan prioritas

tingkat lalu lintas harian rata-rata, kondisi kerusakan jalan, dan adanya laporan dari masyarakat, namun hanya memenuhi kuota dari dana yang tersedia. Sehingga banyak jalan-jalan yang seharusnya menjadi prioritas justru tidak mendapatkan perhatian.

*Simple Additive Weighting* (SAW) merupakan sebuah metode yang dinilai mampu digunakan pada permasalahan tersebut. Metode ini dipilih karena dapat menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif. Dengan metode perankingan ini, diharapkan penilaian akan lebih tepat karena didasarkan pada nilai kriteria dan bobot yang telah ditentukan sehingga akan didapat hasil yang akurat terhadap ruas jalan mana yang menjadi prioritas untuk diperbaiki.

Sistem Pendukung Keputusan Penentuan Prioritas Perbaikan Jalan Kabupaten Jember Menggunakan Metode *Simple Additive Weighting* (SAW) Berbasis Web diharapkan mampu membantu Dinas Pekerjaan Umum Bina Marga Kabupaten Jember dalam menentukan urutan prioritas perbaikan jalan. Sebab sistem ini nantinya akan mampu mengelola data ruas-ruas jalan yang rusak. Selanjutnya data tersebut ditampilkan dalam bentuk urutan prioritas perbaikan jalan. Sehingga Dinas Pekerjaan Umum Bina Marga Kabupaten Jember dapat dengan mudah mengetahui ruas-ruas jalan mana yang diprioritaskan terlebih dahulu untuk kemudian dilakukan kebijakan dan langkah untuk memperbaiki jalan-jalan yang rusak.

## 1.2. Perumusan Masalah

Berdasarkan latar belakang diatas maka dirumuskan permasalahan sebagai berikut :

1. Bagaimana metode *Simple Additive Weighting* (SAW) menentukan urutan prioritas perbaikan jalan?
2. Bagaimana merancang dan membangun sistem pendukung keputusan penentuan prioritas perbaikan jalan Kabupaten Jember dengan menerapkan metode *Simple Additive Weighting* (SAW)?

### 1.3. Tujuan

Tujuan dalam penelitian ini merupakan adalah :

1. Menerapkan metode *Simple Additive Weighting* (SAW) untuk menentukan urutan prioritas perbaikan jalan.
2. Merancang dan membangun sistem pendukung keputusan penentuan prioritas perbaikan jalan Kabupaten Jember dengan menerapkan metode *Simple Additive Weighting* (SAW).

### 1.4. Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Aplikasi hanya digunakan untuk menentukan urutan prioritas perbaikan jalan.
2. Kriteria yang digunakan dalam sistem ini adalah lalulintas harian rata-rata, kondisi jalan, laporan masyarakat.
3. Aplikasi menggunakan metode *Simple Additive Weighting* (SAW).

### 1.5. Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

#### BAB 1. PENDAHULUAN

Bab ini menjelaskan latar belakang, perumusan masalah, tujuan dan sistematika penulisan.

#### BAB 2. TINJAUAN PUSTAKA

Bab ini berisi tentang kajian materi, penelitian terdahulu dan informasi apa saja yang digunakan dalam penelitian ini. Dimulai dari kajian pustaka mengenai penelitian terdahulu hingga model *waterfall*.

#### BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode penelitian yang akan digunakan. Mulai dari tahap studi literatur, pengumpulan data, analisis, dan perancangan sistem.

## BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan tentang analisis dan perancangan sistem yang dikembangkan.

## BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang telah dilakukan dengan memaparkan hasil penelitian dan hasil percobaan pengimplementasian sistem.

## BAB 6. PENUTUP

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



## BAB 2. TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan teori-teori dan pustaka yang digunakan dalam penelitian. Teori yang dibahas meliputi penelitian terdahulu, sistem, sistem informasi, sistem pendukung keputusan, jalan, metode *Simple Additive Weighting*, model *waterfall*.

### 2.1 Penelitian Terdahulu

Penelitian sebelumnya mengenai penentuan prioritas penanganan jalan di wilayah Balai Pemeliharaan Jalan Mojokerto menggunakan metode *Analytical Hierarchy Process* (Faiz, 2009), menyatakan bahwa metode AHP dapat diterapkan untuk menentukan urutan ruas-ruas jalan yang akan mendapat penanganan. Penelitian ini menghitung setiap alternatif dengan menggunakan metode AHP secara manual, belum dibuat sistem untuk masalah tersebut.

Penelitian lain yang berjudul “Sistem Pendukung Keputusan Berbasis Web Untuk Pemilihan Jodoh Islami Dengan Menggunakan Metode *Simple Additive Weighting* (SAW)” (Purwati, 2013) menyatakan bahwa metode SAW tepat digunakan pada permasalahan tersebut. Metode ini dipilih karena dapat menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif. Dengan metode perankingan tersebut, diharapkan penilaian akan lebih tepat karena didasarkan pada nilai kriteria dan bobot yang telah ditentukan sehingga akan didapat hasil yang akurat terhadap siapa yang akan menjadi calon pasangan terbaik.

### 2.2 Sistem Informasi

Menurut (Muhyuzir, 2001), sistem informasi adalah data yang dikumpulkan, dikelompokkan dan diolah sedemikian rupa sehingga menjadi satu kesatuan informasi yang saling terkait dan saling mendukung sehingga menjadi suatu informasi yang berharga bagi yang menerimanya. Sedangkan pengertian sistem

informasi yang lain yaitu suatu sistem yang menerima masukan data dan instruksi, mengolah data tersebut sesuai dengan instruksi dan mengeluarkan hasilnya. (Davis, 1991).

Beberapa definisi diatas dapat disimpulkan bahwa sistem informasi merupakan sebuah sistem yang terintegrasi secara optimal dan berbasis komputer yang dapat menghimpun dan menyajikan berbagai jenis data yang akurat untuk berbagai macam kebutuhan.

### 2.3 Sistem Pendukung Keputusan

Menurut (Turban, 2001), Sistem Pendukung Keputusan (SPK) atau *Decision Support System* adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan situasi yang tidak terstruktur, dimana tidak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat.

Sedangkan menurut (Morton, 1978), Sistem Pendukung Keputusan merupakan penggabungan beberapa sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah-masalah semi struktur.

Sementara itu tiga tujuan yang harus dicapai oleh sistem pendukung keputusan, yaitu :

1. Sistem harus dapat membantu manajer dalam membuat keputusan guna memecahkan masalah semi terstruktur.
2. Sistem harus dapat mendukung manajer, bukan mencoba menggantikannya.
3. Sistem harus dapat meningkatkan efektifitas pengambilan keputusan manajer.



Dari pengertian diatas dapat dijelaskan bahwa Sistem Pendukung Keputusan merupakan alat yang digunakan untuk membantu pengambilan keputusan, dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat.

## **2.4 Jalan**

Menurut UU No. 38 tahun 2004 tentang jalan, definisi jalan adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap, dan perlengkapannya yang diperuntukkan bagi lalu lintas, yang berada ditatas permukaan tanah, dibawah permukaan tanah dan atau air, serta diatas permukaan air, kecuali jalan kereta api dan jalan kabel. Jalan umum adalah jalan yang diperuntukkan bagi lalu lintas umum, jalan khusus adalah jalan yang dibangun oleh instansi, badan usaha, perseorangan, atau kelompok masyarakat untuk kepentingan sendiri.

### **2.4.1 Klasifikasi Jalan Menurut Fungsinya**

Jalan umum adalah jalan yang diperuntukkan untuk lalu lintas umum, menurut fungsinya dikelompokkan kedalam jalan arteri, jalan kolektor, jalan lokal, dan jalan lingkungan.

1. Jalan arteri adalah jalan umum yang berfungsi melayani angkutan utama dengan ciri perjalanan jarak jauh, kecepatan rata-rata tinggi, dan jumlah jalan masuk dibatasi secara berdaya guna.
2. Jalan kolektor adalah jalan umum yang berfungsi melayani angkutan pengumpul atau pembagi dengan ciri perjalanan jarak sedang kecepatan rata-rata sedang, dan jumlah jalan masuk dibatasi.
3. Jalan lokal adalah jalan umum yang berfungsi melayani angkutan setempat dengan ciri perjalanan jarak dekat, kecepatan rata-rata rendah, dan jumlah jalan masuk tidak dibatasi.

4. Jalan lingkungan adalah jalan umum yang berfungsi melayani angkutan lingkungan dengan ciri perjalanan jarak dekat, dan kecepatan rata-rata rendah.

#### 2.4.2 Klasifikasi Jalan Menurut Statusnya

Jalan umum menurut statusnya dikelompokkan kedalam jalan nasional, jalan provinsi, jalan kabupaten, jalan kota dan jalan desa.

1. Jalan Nasional merupakan jalan arteri dan jalan kolektor dalam sistem jaringan jalan primer yang menghubungkan antar ibukota provinsi, dan jalan strategis nasional, serta jalan tol.
2. Jalan provinsi adalah jalan jalan kolektor dalam sistem jaringan jalan primer yang menghubungkan ibukota provinsi dengan ibukota kabupaten/kota, atau antar ibukota kabupaten/kota, dan jalan strategis provinsi.
3. Jalan kabupaten adalah jalan lokal dalam sistem jaringan jalan primer yang tidak termasuk pada jalan nasional dan provinsi yang menghubungkan ibukota kabupaten dan ibukota kecamatan, antar ibukota kecamatan, dengan pusat kegiatan lokal.
4. Jalan kota adalah jalan umum dalam sistem jaringan jalan sekunder yang menghubungkan antar pusat pelayanan dalam kota, menghubungkan pusat pelayanan dengan persil, menghubungkan antar persil, serta menghubungkan antar pusat pemukiman yang berada dalam kota.
5. Jalan desa merupakan jalan umum yang menghubungkan kawasan dan/atau antar pemukiman dalam desa, serta jalan lingkungan.

### 2.5 Metode *Simple Additive Weighting*

Pada sub bab ini dijelaskan tentang metode *Simple Additive Weighting* (SAW) diantaranya, pengertian metode *Simple Additive Weighting* (SAW), langkah-langkah metode *Simple Additive Weighting* (SAW).

### 2.5.1 Pengertian Metode *Simple Additive Weighting* (SAW)

Menurut (Kusumadewi, 2006), metode SAW sering juga dikenal dengan istilah metode penjumlahan terbobot. Konsep dasar SAW adalah mencari penjumlahan terbobot dari kinerja setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat dibandingkan dengan semua rating alternatif yang ada.

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases} \quad \text{..... persamaan 1}$$

Keterangan :

$r_{ij}$  = nilai rating kinerja normalisasi

$X_{ij}$  = nilai yang dimiliki dari setiap kriteria

$Max_{ij}$  = nilai terbesar dari setiap kriteria

$Min_{ij}$  = nilai terkecil dari setiap kriteria

Atribut keuntungan (*benefit*) = nilai terbesar adalah terbaik

Atribut biaya (*cost*) = nilai terkecil adalah terbaik

Dimana  $r_{ij}$  adalah rating kinerja ternormalisasi dari alternatif  $A_i$  pada atribut  $C_j$  :  $i = 1, 2, \dots, m$  dan  $j = 1, 2, \dots, n$ . Nilai preferensi untuk setiap alternatif ( $V_i$ ) diberikan sebagai berikut:

$$V_i = \sum_{j=1}^n w_j r_{ij} \quad \text{..... persamaan 2}$$

Keterangan :

$V_i$  = nilai preferensi

$w_j$  = bobot ranking

$r_{ij}$  = rating kinerja ternormalisasi

Nilai  $V_i$  lebih besar mengindikasikan bahwa alternatif  $A_i$  lebih terpilih.

### 2.5.2 Langkah-langkah Metode *Simple Additive Weighting* (SAW)

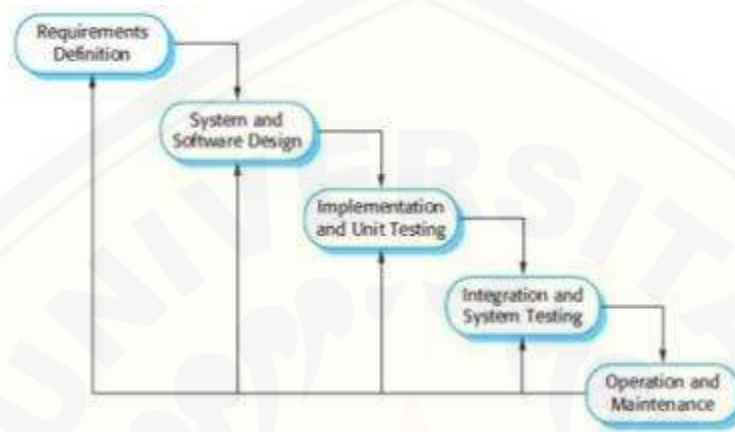
Menurut (Kusumadewi, 2006), dalam jurnal Implementasi Metode *Simple Additive Weighting* Dalam Sistem Informasi Lowongan Kerja Berbasis Web untuk Rekomendasi Pencari Kerja Terbaik (Darmastuti, 2013), langkah-langkah metode SAW :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (C), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R dengan menggunakan rumus persamaan 1.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik dengan menggunakan rumus persamaan 2.

## 2.6 Model *Waterfall*

Pembuatan sistem informasi pada penelitian ini menggunakan model *waterfall*. Metode ini merupakan metode yang tepat untuk membangun sebuah perangkat lunak yang tidak terlalu besar dan sumber daya manusia yang terlibat dalam jumlah yang terbatas. Menurut (Sommerville, 2011), tahapan utama dari *waterfall model* langsung mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan pada *waterfall model*, yaitu *requirement analysis and definition, system and software design, implementation and unit testing, integration and system testing*, dan *operation and*

*maintenance*. Alur *life cycle* pengembangan perangkat lunak pada model waterfall diselesaikan pertahapan dan berurutan. Alur dari model *waterfall* dapat dilihat pada gambar 2.1 berikut :



Gambar 2.1 Alur model *waterfall*

Keterangan dari skema di atas adalah :

1. *Requirements Definition* (Analisis Kebutuhan)

Tahap ini menganalisis kebutuhan yang akan digunakan dalam pembuatan aplikasi. Meliputi pengumpulan data kebutuhan fungsional dan non-fungsional dari aplikasi yang akan kita bangun.

2. *System and Software Design*

Jika proses analisis kebutuhan telah diketahui maka proses selanjutnya adalah pada tahapan desain sistem. Proses pendesainan sistem dari aplikasi yang akan kita bangun yaitu dengan menggunakan *Unified Modeling Language* (UML). Penggunaan UML karena sudah menggunakan konsep *Object Oriented Design* yang tentunya akan sangat memudahkan developer untuk membangun sebuah sistem. UML diagram yang akan dibuat antara lain:

- a. *Business Process*

*Business Proses* digunakan untuk menggambarkan inputan data yang dibutuhkan sistem, output dari sistem serta tujuan dari pembuatan sistem.



b. *Use Case Diagram*

*Use case* merupakan model yang menggambarkan fungsi atau tugas yang dilakukan oleh *user*, baik manusia maupun mesin / komputer. *Use Case* model ini dapat digunakan untuk menggambarkan *job spesification* dan *job description*, serta keterkaitan antar *job*.

c. *Scenario*

*Scenario* diagram digunakan untuk menjelaskan atau menceritakan fitur atau isi yang ada di *use case* diagram. *Scenario* menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.

d. *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. *Activity* diagram mempunyai fungsi yang sama dengan *scenario* namun diimplementasikan dalam diagram alir.

e. *Sequence Diagram*

*Sequence* diagram digunakan untuk menunjukkan rangkaian pesan yang dikirim antar object juga interaksi antar object.

f. *Class Diagram*

*Class* diagram digunakan untuk menggambarkan struktur statis class dalam sistem. *Class Diagram* dibuat untuk memudahkan dalam proses pengkodean. *Entity Relationship Diagram*

g. ERD

Merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

3. *Implementation and Unit Testing*

Tahap ini mengimplementasikan desain yang telah dibuat menjadi sebuah aplikasi berbasis web. Beberapa hal yang dilakukan dalam tahap implementasi adalah menulis kode program (*coding*) menggunakan bahasa pemrograman *HTML* dan *PHP: Hypertext Preprocessor (PHP)*. *HTML* dan *PHP* merupakan bahasa pemrograman utama dalam pembangunan aplikasi berbasis web. Manajemen basis data yang digunakan dalam pembangunan aplikasi adalah *DBMS MySQL*.

#### 4. *Integration and System Testing*

Tahap pengujian yaitu uji coba terhadap sistem yang dibuat. Pengujian wajib dilakukan untuk menguji apakah sistem ini sudah sesuai dengan kebutuhan dari user atau belum. Dan apakah masih ada kesalahan maupun kelemahan terhadap sistem yang kami bangun tersebut. Diharapkan proses pengujian / testing dapat menyempurnakan sistem yang kami buat. Pengujian perangkat lunak menggunakan dua metode yakni :

##### 1. *White Box Testing*

Merupakan cara pengujian dengan melihat modul untuk yang telah dibuat dengan program – program yang ada. Dan menganalisa apakah terjadi kesalahan atau tidak pada penulisan kode program. Pengujian ini dilakukan oleh (*develeoper*) pembuat program. Menurut (Pressman, 2005) pengujian *white box* merupakan teknik pengujian jalur dasar yang digunakan untuk menentukan kompleksitas logis dengan menentukan rangkaian dasar jalur eksekusinya. Tahapan teknik pengujian jalur dasar meliputi dari mulai listing program, grafik alir, kompleksitas siklomatik, jalur program independen dan pengujian basis set.

##### 2. *Black Box Testing*

Menurut (Pressman, 2005), Black-Box Testing adalah metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini berusaha menemukan kesalahan dalam kategori sebagai berikut: fungsi-fungsi yang tidak benar atau hilang, kesalahan interface, kesalahan dalam struktur data atau akses database eksternal, kesalahan kinerja. tabel pengujian *black box* dapat dilihat pada tabel 2.1.

Tabel 2.1 Tabel pengujian *black box*

No	Fitur	Kasus	Hasil	Keterangan
----	-------	-------	-------	------------

---

##### 5. *Operation and Maintenance*

Tahap ini merupakan tahap akhir dari perancangan model *waterfall*. Penerapan program ini dilakukan setelah program yang dibuat oleh penulis selesai dan telah melalui tahap pengujian terlebih dahulu. Selanjutnya dilakukan proses *maintenance* atau perawatan mengatasi masalah *bugs* yang muncul setelah aplikasi diserahkan pada *user*.



### **BAB 3. METODOLOGI PENELITIAN**

Pada bab ini dijelaskan tentang metode-metode yang digunakan selama penelitian dilakukan, seperti jenis penelitian, studi literatur, data dan sumber data penelitian, serta tahapan analisis hingga model perancangan sistem.

#### **3.1 Jenis Penelitian**

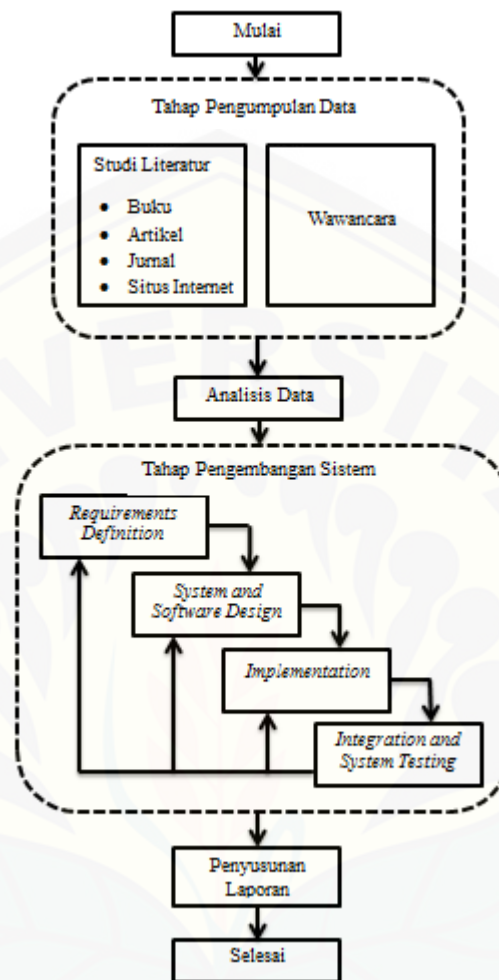
Dalam penelitian ini penulis menggunakan metode penelitian kualitatif dan kuantitatif. Metode kualitatif dalam penelitian ini meliputi tahapan penemuan masalah yang akan diteliti kemudian mengkaji studi literatur yang berkaitan dengan cara penyelesaian masalah yang didapat. Dilanjutkan dengan wawancara kepada pihak terkait penelitian yaitu wawancara kepada Dinas Pekerjaan Umum BINA MARGA Kabupaten Jember. Sedangkan metode kuantitatif dalam penelitian ini meliputi pengolahan data hasil dari tahapan wawancara kepada Dinas Pekerjaan Umum BINA MARGA Kabupaten Jember. Data atribut kriteria dan alternatif jalan dihitung menggunakan sistem perankingan *Simple Additive Weighting* sehingga data mentah tersebut dapat menghasilkan urutan prioritas perbaikan jalan.

#### **3.2 Tempat dan Waktu Penelitian**

Tempat pelaksanaan penelitian yaitu Dinas Pekerjaan Umum BINA MARGA Kabupaten Jember. Waktu penelitian dilakukan selama lima bulan, dimulai pada bulan Maret 2015 sampai bulan Juli 2015

#### **3.3 Tahapan Penelitian**

Penelitian ini akan dilakukan dalam beberapa tahap diantaranya teknik pengumpulan data, tahap analisis dan tahap pengembangan sistem. Tahapan penelitian disini digunakan untuk mencapai tujuan penelitian yaitu untuk menentukan prioritas perbaikan jalan. Tahapan alur penelitian untuk membuat sistem pendukung keputusan penentuan prioritas perbaikan jalan digambarkan dalam diagram alir seperti pada gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian (Sumber:Hasil Analisis, 2015)

### 3.3.1 Teknik Pengumpulan Data

Proses untuk mendapatkan data yang dibutuhkan untuk membangun sistem pendukung keputusan penentuan prioritas perbaikan jalan menggunakan teknik pengumpulan data sebagai berikut:

#### 1. Studi Literatur

Pada tahap ini dilakukan pengumpulan data dan informasi yang diperlukan untuk proses perancangan sistem. Data dan informasi dapat diperoleh dari lokasi penelitian yaitu di DPU Bina Marga Kabupaten Jember. Selain itu, studi literatur juga dapat

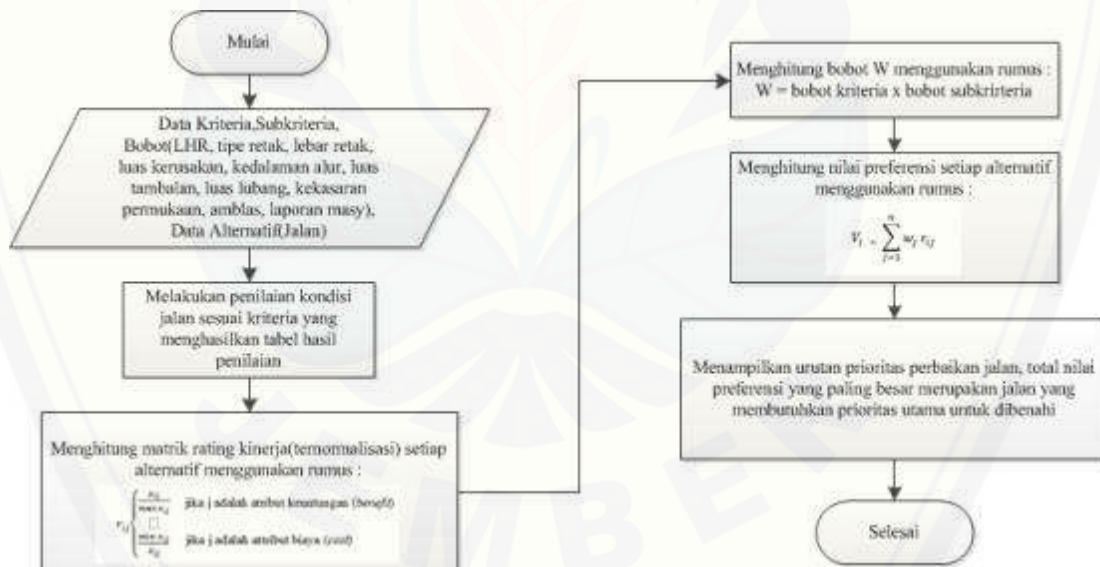
diperoleh dari *paper*, jurnal ilmiah, serta buku-buku referensi yang berkaitan dengan penelitian.

## 2. Wawancara

Metode interview atau wawancara yaitu mencari secara langsung dari pihak yang bersangkutan atau interview secara langsung. Penulis dalam penelitian ini melakukan wawancara secara langsung kepada pihak DPU Bina Marga Kabupaten Jember.

### 3.3.2 Tahap Analisis Data

Tahap analisis dilakukan setelah melakukan pengumpulan data yang diperoleh dari studi literatur dan wawancara mengenai penentuan prioritas perbaikan jalan. Data yang diperoleh akan dianalisa dengan metode *Simple Additive Weighting* (SAW). Alur dari proses metode SAW dapat dilihat pada gambar 3.2



Gambar 3.2 Flowchart metode SAW pada sistem

Tahap pertama alur metode pada gambar 3.2 yaitu menentukan kriteria, subkriteria, bobot dan alternatif(jalan). Setelah mengetahui kriteria yang digunakan kemudian dilanjutkan dengan melakukan penilaian kondisi jalan sesuai dengan

kriteria, penilaian kondisi jalan dilakukan oleh surveyor. Dilanjutkan dengan menghitung matrix rating kinerja tiap alternatif. Setelah mendapatkan hasil dari penghitungan rating kinerja dilanjutkan dengan menghitung bobot W. Kemudian dilanjutkan dengan menghitung nilai preferensi setiap alternatif. Hasil total nilai preferensi kemudian ditampilkan dalam bentuk urutan dari nilai terbesar sampai terkecil. Nilai preferensi yang paling paling besar merupakan jalan yang membutuhkan prioritas untuk dibenahi pertama.

### 3.3.3 Tahap Pengembangan Sistem

Setelah tahap pengumpulan data selesai, kemudian akan dilanjutkan tahap pengembangan sistem dengan menggunakan konsep berbasis objek dengan pemodelan *Unified Modelling Language* (UML). Pemodelan UML yang digunakan pada penelitian ini antara lain, *Business Process*, *Usecase Diagram*, *Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class diagram* dan *Entity Relationship Diagram* (ERD).

Setelah tahap perancangan selesai, dilanjutkan dengan tahap implementasi menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor* (PHP) dengan bantuan *framework Code Igniter* (CI) dan manajemen basisdata menggunakan *DBMS MySQL*. Hasil perancangan dan implementasi kemudian akan ditesting menggunakan *White Box* dan *Black Box*. Ketika ada perbaikan dalam sistem maka proses dari tahapan pengembangan sistem harus diulang sesuai dengan model pengembangan sistem pada penelitian ini yaitu model *waterfall*.

## BAB 4. PENGEMBANGAN SISTEM

Bab ini akan membahas tentang pengembangan sistem untuk membuat sistem pendukung keputusan penentuan prioritas perbaikan jalan menggunakan metode *Simple Additive Weighting* (SAW). Tahapan-tahapan pengembangan dilaksanakan berdasarkan model *waterfall*.

### 4.1 Deskripsi Umum Sistem

Deskripsi umum dari sistem pendukung keputusan penentuan prioritas perbaikan jalan Kabupaten Jember yang dibangun dalam penelitian ini akan dijelaskan lebih detail pada SOP (*statement of purpose*) sistem.

#### 4.1.1 SOP (*statement of purpose*) Sistem

Sistem pendukung keputusan penentuan prioritas perbaikan jalan ini adalah sebuah sistem berbasis web yang dapat digunakan dalam menunjang keputusan dalam penentuan jalan-jalan mana yang membutuhkan prioritas utama untuk diperbaiki. Pengambilan keputusan dilakukan dengan mempertimbangkan kriteria-kriteria dan bobot-bobot kriteria yang digunakan untuk penentuan prioritas. Kriteria-kriteria dan bobot-bobot kriteria diolah dengan menggunakan metode *Simple Additive Weighting* (SAW) sehingga dapat menghasilkan suatu keputusan yaitu urutan prioritas perbaikan jalan. Sistem pendukung keputusan penentuan prioritas perbaikan jalan ini memiliki beberapa fitur antara lain : fitur mengelola data user, mengelola data jalan, mengelola data survei, mengelola data survei detail, melihat laporan masyarakat, menambah laporan masyarakat, analisis prioritas jalan, mengubah bobot kriteria, mengubah bobot subkriteria, mengubah bobot nilai subkriteria. Dengan adanya sistem ini diharapkan dapat membantu perencanaan pemeliharaan jalan dalam hal menentukan prioritas perbaikan jalan.



## 4.2 Analisis Kebutuhan

Analisis kebutuhan perangkat lunak dalam penelitian ini yaitu dengan cara mengidentifikasi permasalahan yang ada untuk kemudian dicatat dan dijadikan bahan untuk mulai membangun sistem pendukung keputusan penentuan prioritas perbaikan jalan. Analisis kebutuhan yang dilakukan meliputi proses pengumpulan data kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Sedangkan kebutuhan nonfungsional merupakan layanan pada sistem yang tidak secara langsung terkait pada fungsi sistem. Berikut adalah kebutuhan fungsional dan non-fungsional sistem :

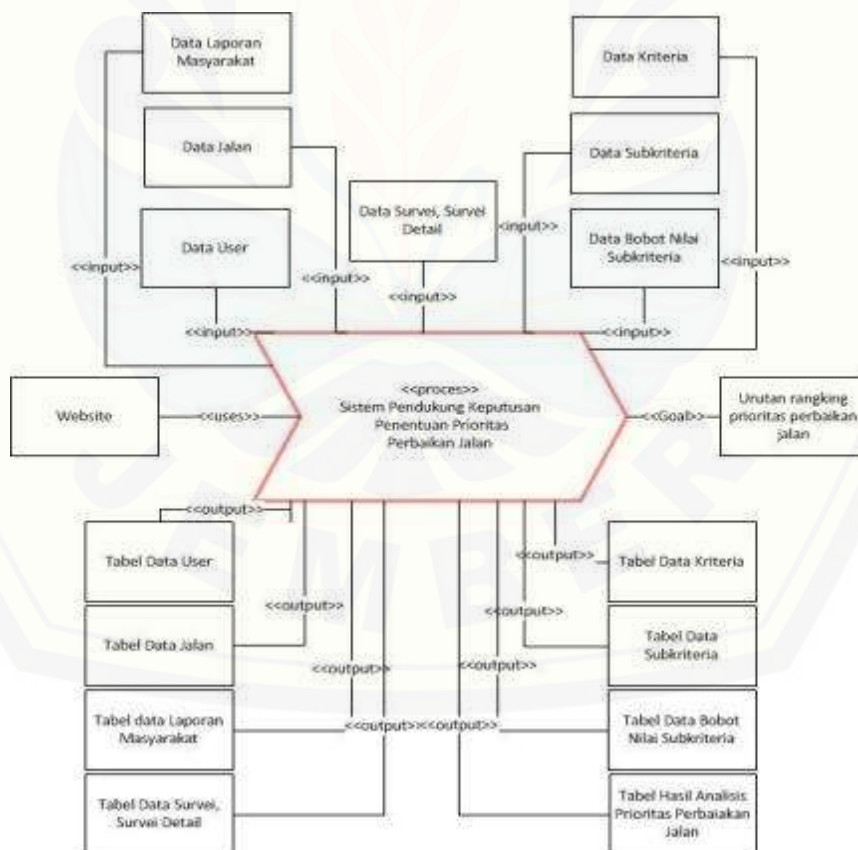
- a. Kebutuhan fungsional dari sistem yang akan dibangun
  1. Sistem dapat mengelola data user.
  2. Sistem dapat mengelola data jalan.
  3. Sistem dapat mengelola data survei.
  4. Sistem dapat mengelola data survei detail.
  5. Sistem dapat menambah laporan masyarakat.
  6. Sistem dapat menampilkan laporan masyarakat.
  7. Sistem dapat menampilkan hasil analisis prioritas jalan berupa tabel hasil penilaian, tabel rating kinerja, tabel nilai preferensi dan urutan prioritas.
  8. Sistem dapat mengubah bobot kriteria.
  9. Sistem dapat mengubah bobot subkriteria.
  10. Sistem dapat mengubah bobot nilai subkriteria.
- b. Kebutuhan non-fungsional dari sistem yang akan dibangun
  1. Sistem berbasis *website*
  2. Sistem memiliki batasan hak akses pengguna dengan menggunakan username dan password
  3. Sistem menggunakan framework Codeigniter

### 4.3 Desain Sistem

Desain sistem adalah tahapan untuk memodelkan sistem pendukung keputusan yang akan dibuat. Dokumen desain sistem yang akan dibuat meliputi *Business Process*, *Usecase Diagram*, *Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class diagram* dan *Entity Relationship Diagram (ERD)*.

#### 4.3.1 Bussines Process

*Business Process* merupakan model atau diagram yang menggambarkan sebuah proses lengkap dengan *resources* dan *information* yang dibutuhkan, event yang mendorong terjadinya proses dan goal yang dituju. *Bussines Process* digunakan untuk menggambarkan inputan data yang dibutuhkan sistem, output dari sistem serta tujuan dari pembuatan sistem. *Bussines Process* sistem dapat dilihat pada gambar 4.1

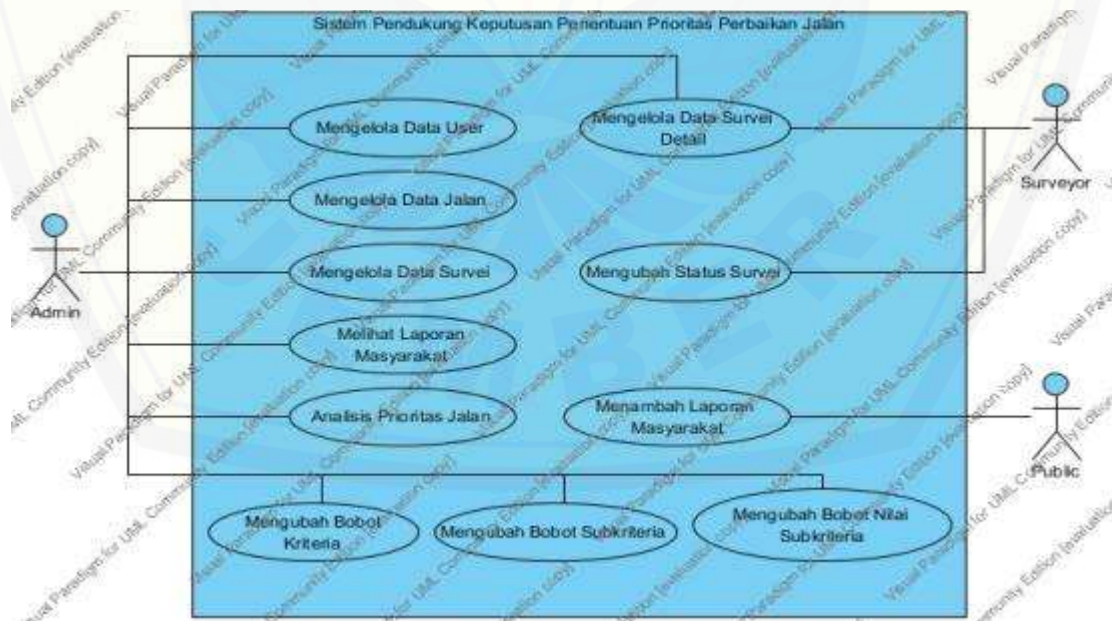


Gambar 4.1 *Business process* sistem

Berdasarkan gambar 4.1 sistem membutuhkan tujuh inputan data dan menghasilkan delapan *output*. Data yang diperlukan sebagai inputan yaitu data user, data jalan, data laporan masyarakat, data survei dan survei detail, data kriteria, data subkriteria dan data bobot nilai subkriteria. Sedangkan data *output* sebagai hasil dari pengolahan sistem yaitu tabel data user, tabel data jalan, tabel data laporan masyarakat, tabel data survei dan survei detail, tabel data kriteria, tabel data subkriteria, tabel data bobot nilai subkriteria dan tabel hasil analisis prioritas perbaikan jalan. Tujuan dari sistem ini adalah urutan ranking prioritas perbaikan jalan.

#### 4.3.2 Usecase Diagram

*Usecase Diagram* adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Usecase Diagram* berfungsi untuk menggambarkan fitur apa saja yang akan dijalankan pada *software* yang akan dibuat. *Usecase Diagram* sistem pendukung keputusan penentuan prioritas perbaikan jalan dapat dilihat pada gambar 4.2.



Gambar 4.2 Usecase diagram sistem



Definisi *usecase* dari *usecase* sistem dapat dilihat pada tabel 4.1. Sedangkan definisi aktor dapat dilihat pada tabel 4.2.

Tabel 4.1 Definisi *usecase*

No	Use case	Deskripsi
<b>1</b>	<b>Mengelola Data User</b>	
	Tambah data user	Proses menambahkan data user kedalam sistem oleh admin
	Ubah data user	Proses mengubah isian data user oleh admin
	Hapus data user	Proses menghapus data user dari sistem oleh admin.
<b>2</b>	<b>Mengelola Data Jalan</b>	
	Tambah data jalan	Proses menambahkan data jalan kedalam sistem oleh admin
	Ubah data jalan	Proses mengubah isian data jalan oleh admin
	Hapus data jalan	Proses menghapus data jalan dari sistem oleh admin.
<b>3</b>	<b>Mengelola Data Survei</b>	
	Tambah data survei	Proses menambahkan data survei kedalam sistem oleh admin
	Ubah data survei	Proses mengubah isian data survei oleh admin
	Hapus data survei	Proses menghapus data survei dari sistem oleh admin.
<b>3</b>	<b>Mengubah Status Survei</b>	
	Ubah status survei	Proses mengubah status survei oleh surveyor
<b>4</b>	<b>Mengelola Data Survei Detail</b>	
	Tambah data survei	Proses menambahkan data survei detail

detail admin	kedalam sistem oleh admin
Hapus data survei detail admin	Proses menghapus data survei detail dari sistem oleh admin
Tambah data survei detail surveyor	Proses menambahkan data survei detail kedalam sistem oleh surveyor
Hapus data survei detail surveyor	Proses menghapus data survei detail dari sistem oleh surveyor
<b>5</b>	<b>Mengubah Bobot Kriteria</b>
Ubah bobot kriteria	Proses mengubah isian bobo kriteria oleh admin
<b>6</b>	<b>Mengubah Bobot Subkriteria</b>
Ubah bobot subkriteria	Proses mengubah isian bobot subkriteria oleh admin
<b>7</b>	<b>Mengubah Bobot Nilai Subkriteria</b>
Ubah bobot nilai subkriteria	Proses mengubah isian bobot nilai subkriteria oleh admin
<b>8</b>	<b>Melihat Laporan Masyarakat</b>
Lihat data laporan masyarakat	proses pengguna melihat data laporan masyarakat yang ada didalam sistem
<b>9</b>	<b>Analisis Prioritas Jalan</b>
Analisis prioritas kerusakan jalan	Proses penghitungan nilai masing kondisi jalan. Kemudian ditampilkan sesuai dengan urutan prioritasnya.
<b>10</b>	<b>Menambah Laporan Masyarakat</b>
Tambah laporan masyarakat	Proses menambahkan laporan masyarakat terkait kerusakan jalan kedalam sistem oleh user publik

Tabel 4.2 Definisi Aktor

No.	Usecase	Deskripsi
1.	Admin	Operator sistem yang mengelola data user, data jalan, data survei dan survei detail, mengubah data kriteria(kriteria, subkriteria, dan bobot nilai subkriteria), melihat adanya laporan masyarakat dan menganalisis urutan prioritas perbaikan jalan.
2.	Surveyor	Operator sistem yang mengelola data survei detail(tambah, hapus) dan mengubah status survei
3.	Publik	User publik atau masyarakat yang dapat menambahkan laporan terkait kerusakan jalan.

#### 4.3.3 Scenario

*Scenario* diagram digunakan untuk menjelaskan atau menceritakan fitur atau isi yang ada di *use case* diagram. *Scenario* berfungsi untuk menggambarkan alur sistem beserta alternatif alur yang akan dijalankan oleh user pada sistem pendukung keputusan penentuan prioritas perbaikan jalan. Berikut adalah skenario dari sistem yang akan dibuat.

Tabel 4.3 Skenario tambah data user

<b>Nama Usecase</b>	: Mengelola data user
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil menambah data user baru

#### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Data User”	2. Menampilkan <i>dropdown</i> menu “Data User”

3. Klik menu “Form User”
4. Menampilkan halaman form user
5. Mengisi form user
6. Klik tombol “Simpan”
7. Menyimpan data ke database
8. Menampilkan *alert success* pada halaman tabel data user

**SKENARIO ALTERNATIF**

- 6a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.
- 7a. Menampilkan *warning message* pada *field* yang salah

Tabel 4.3 merupakan skenario dari *usecase* mengelola data user. Skenario tambah data user menjelaskan alur proses input data user. Skenario utama merupakan alur utama dari proses input user. Sedangkan skenario alternatif merupakan bagian yang menangani exception atau alur alternatif dari proses input data user. Kondisi setelah skenario ini dijalankan adalah admin berhasil menambahkan data user baru.

Tabel 4.4 Skenario ubah data user

<b>Nama Usecase</b>	: Mengelola data user
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil mengubah data user

**SKENARIO UTAMA**

- | <i>Actor</i>  | <i>System</i>                                   |
|---|---|
| 1. Klik menu “Data User”  | 2. Menampilkan <i>dropdown</i> menu “Data User” |
| 3. Klik menu “Tabel Data User”                                      | 4. Menampilkan halaman tabel data user          |
| 5. Klik <i>icon</i> edit “gambar pensil” pada data yang akan diedit | 6. Menampilkan form edit data user              |
| 7. Mengisi form edit data user                                      |   |

- 8. Klik tombol “Simpan”
- 9. Menampilkan *pop up* “Update data?”
- 10. Klik tombol “OK”
- 11. Menyimpan data ke database
- 12. Menampilkan *alert success* pada halaman tabel data user

**SKENARIO ALTERNATIF**

- 8a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.
- 9a. Menampilkan *warning message* pada *field* yang salah

- 8b. Ketika klik tombol “Batal”
- 9b. Menampilkan halaman tabel data user

- 10c. Ketika klik tombol “Cancel” ditampilkan *pop up*
- 11c. Menampilkan form edit data user

Tabel 4.4 merupakan skenario dari *usecase* mengelola data user. Skenario ubah data user menjelaskan alur proses ubah data user. Skenario utama merupakan alur utama dari proses ubah user. Sedangkan skenario alternatif merupakan bagian yang menangani exception atau alur alternatif dari proses ubah data user. Kondisi setelah skenario ini dijalankan adalah admin berhasil mengubah data user.

Tabel 4.5 Skenario hapus data user

<b>Nama Usecase</b>	: Mengelola data user
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil menghapus data user

**SKENARIO UTAMA**

- | <i>Actor</i>                   | <i>System</i>                                   |
|--------------------------------|---|
| 1. Klik menu “Data User”       | 2. Menampilkan <i>dropdown</i> menu “Data User” |
| 3. Klik menu “Tabel Data User” |   |

5. Klik *icon* hapus “gambar sampah” pada data yang akan dihapus
7. Klik tombol “OK”
4. Menampilkan halaman tabel data user
6. Menampilkan *pop up* “Hapus data?”
8. Menghapus data user dari database
9. Menampilkan *alert success* pada halaman tabel data user

**SKENARIO ALTERNATIF**

- 7a. Ketika klik tombol “Cancel” ditampilkan *pop up*
- 8a. Menampilkan halaman tabel data user

Tabel 4.5 merupakan skenario dari *usecase* mengelola data user. Skenario hapus data user menjelaskan alur proses menghapus data user. Skenario utama merupakan alur utama dari proses hapus user. Sedangkan skenario alternatif merupakan bagian yang menangani exception atau alur alternatif dari proses hapus data user. Kondisi setelah skenario ini dijalankan adalah admin berhasil menghapus data user.

Tabel 4.6 Skenario analisis prioritas perbaikan jalan

<b>Nama Usecase</b>	: Analisis prioritas jalan
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login ke sistem
<b>Pasca-Kondisi</b>	: Admin berhasil melihat urutan prioritas perbaikan jalan

**SKENARIO UTAMA**

- | <i>Actor</i>  | <i>System</i>   |
|---|---|
| 1. Klik menu “Prioritas Jalan”                                  | 2. Menampilkan halaman tabel data survei yang sudah selesai |
| 3. Klik tombol “Analisis” pada data survei yang akan dianalisis | 4. Menghitung nilai preferensi semua alternatif             |



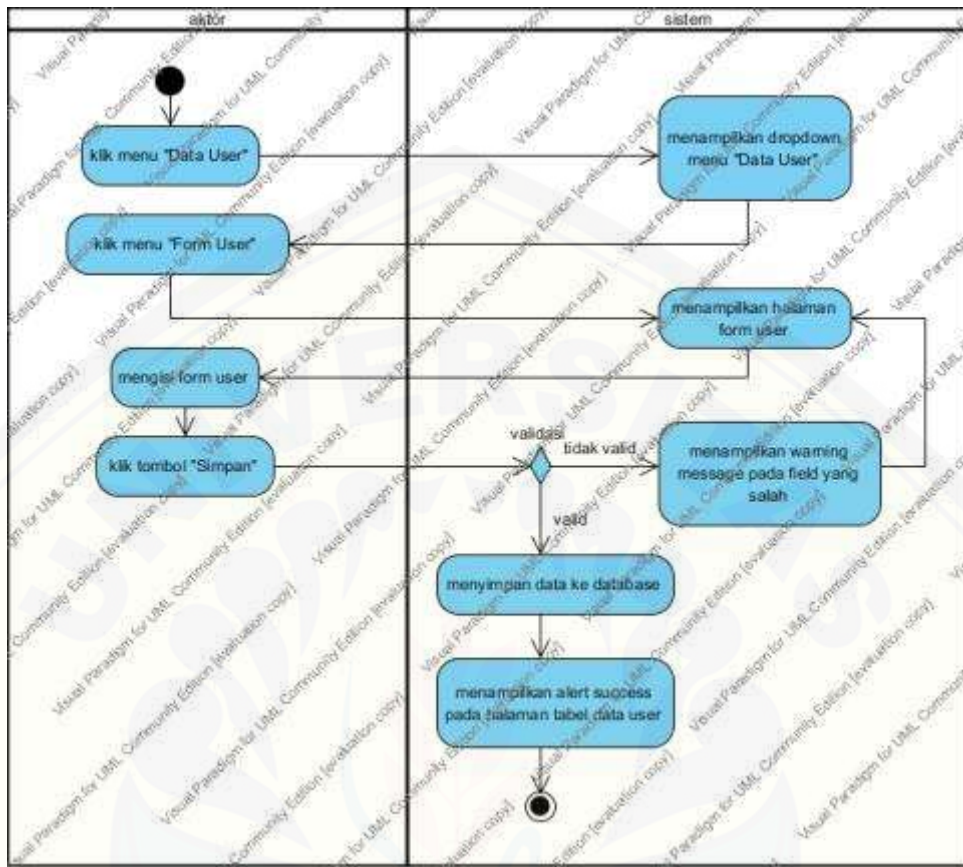
5. Menampilkan tabel penilaian alternatif, tabel matrix rating kinerja (ternormalisasi), tabel hasil nilai preferensi

### SKENARIO ALTERNATIF

Tabel 4.6 merupakan skenario dari *usecase* analisis prioritas jalan. Skenario analisis prioritas perbaikan jalan menjelaskan alur proses analisis prioritas perbaikan jalan. Skenario utama merupakan alur utama dari proses analisis prioritas perbaikan jalan. Sedangkan skenario alternatif merupakan bagian yang menangani exception atau alur alternatif dari proses analisis prioritas perbaikan jalan. Kondisi setelah skenario ini dijalankan adalah admin berhasil melihat urutan prioritas perbaikan jalan.

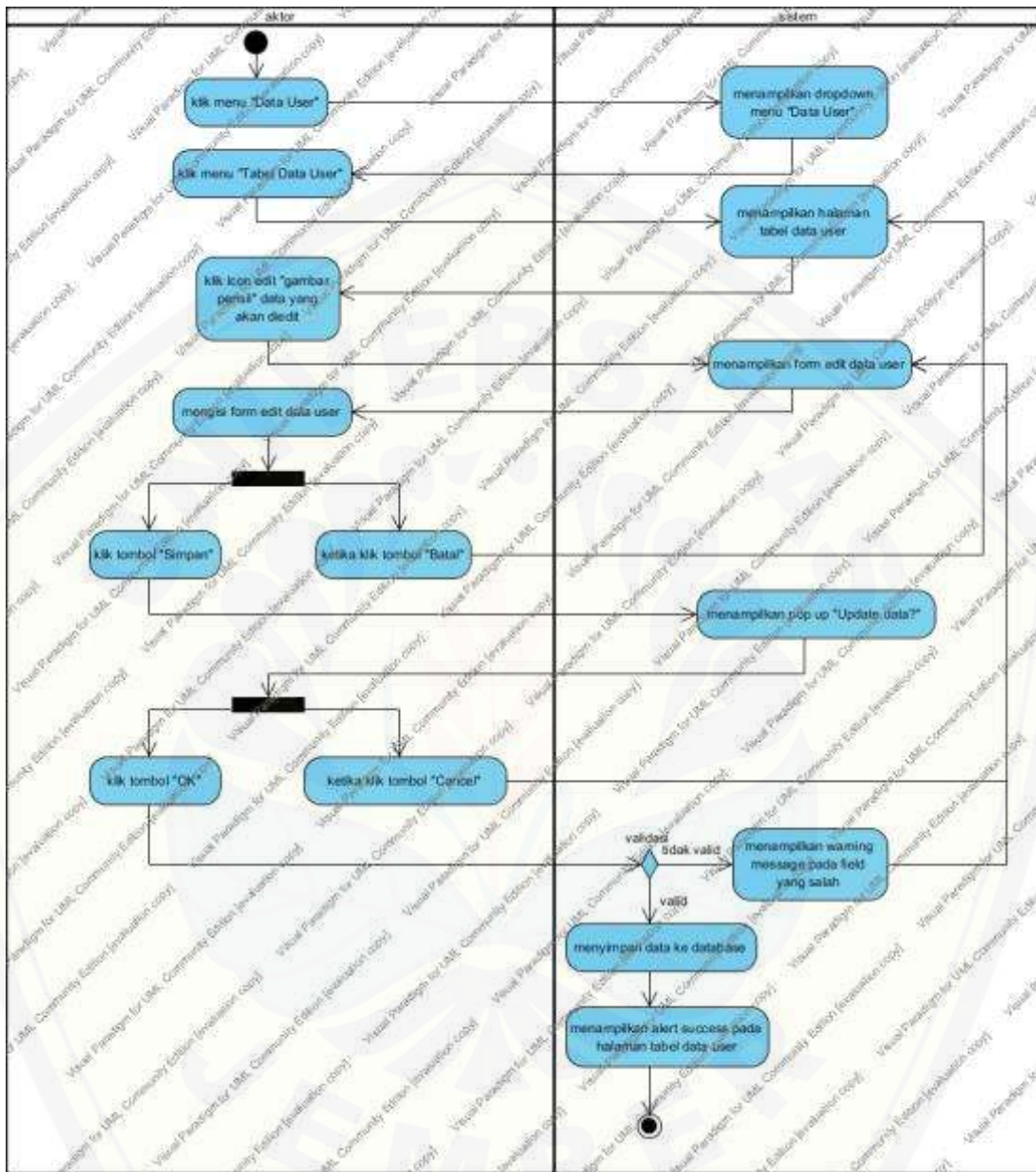
#### 4.3.4 Activity diagram

*Activity diagram* menggambarkan aliran aktivitas dalam sistem pendukung keputusan penentuan prioritas perbaikan jalan yang akan dibangun. *Activity diagram* sistem adalah sebagai berikut :



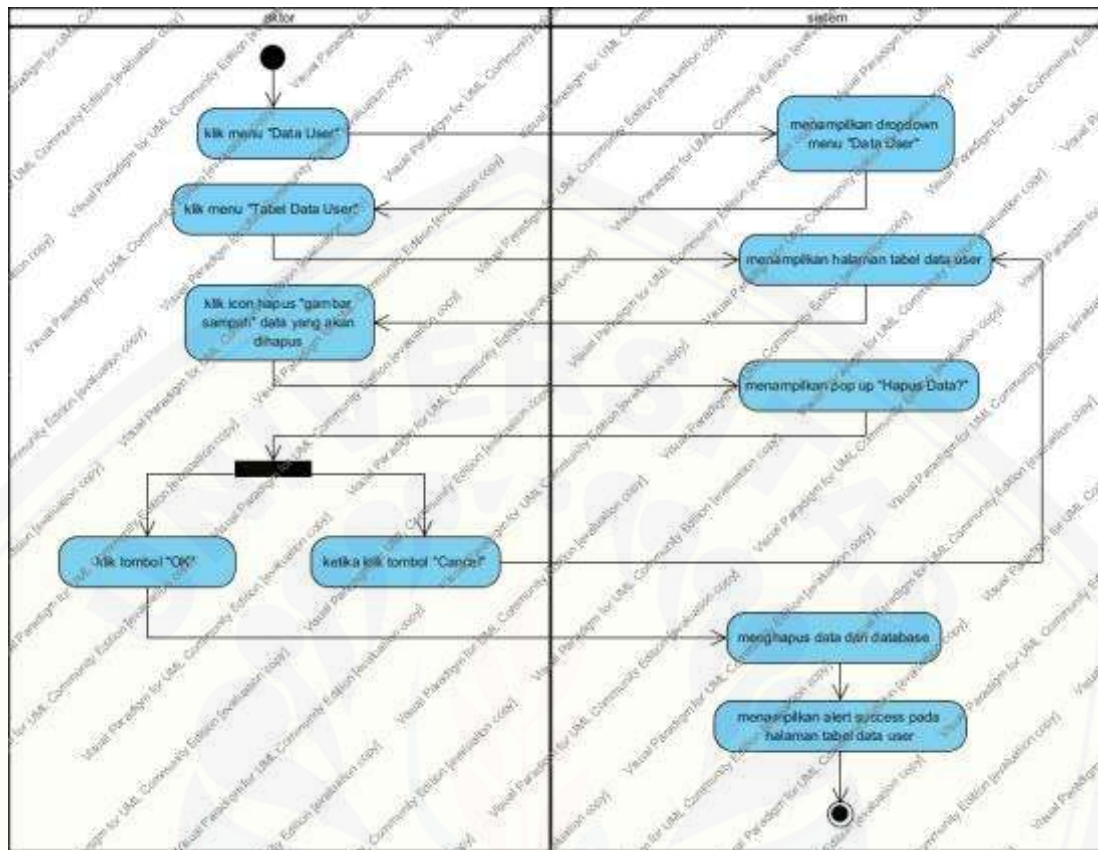
Gambar 4.3 *Activity diagram* tambah data user

Gambar 4.3 menggambarkan alur aktivitas dari proses tambah data user. Admin dapat menambahkan data user baru kedalam sistem, setelah data tersimpan ditampilkan dalam tabel data user.



Gambar 4.4 Activity diagram ubah data user

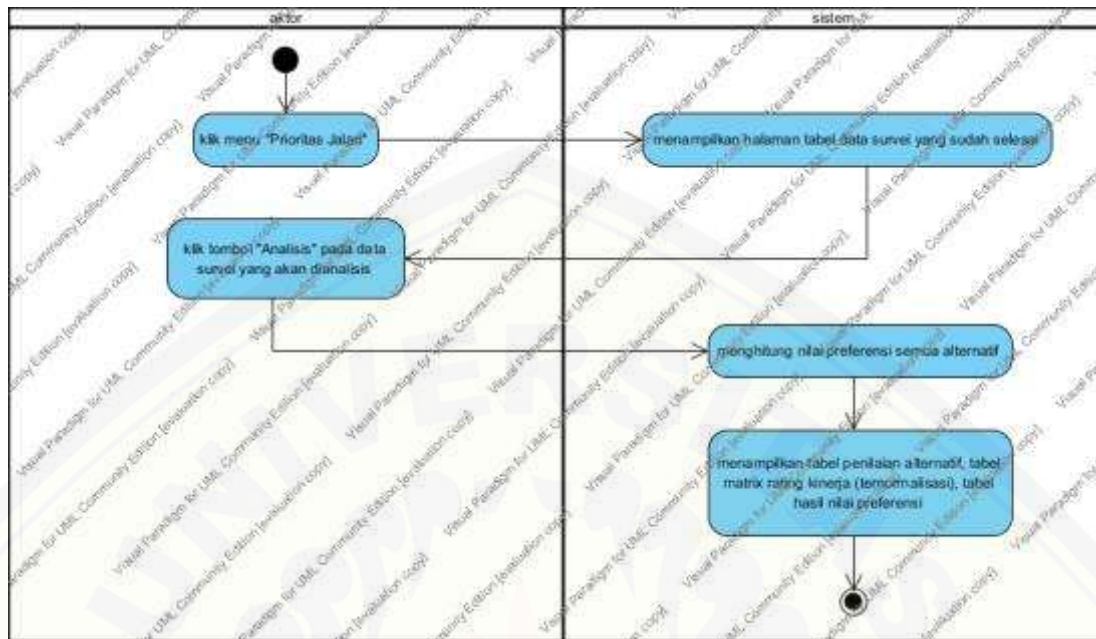
Gambar 4.4 menggambarkan alur aktivitas dari proses ubah data user. Admin dapat mengubah data user sistem, setelah data diubah dan disimpan ditampilkan dalam tabel data user.



Gambar 4.5 Activity diagram hapus data user

Gambar 4.5 menggambarkan alur aktivitas dari proses hapus data user. Admin dapat menghapus data user pada sistem, setelah data dihapus dilanjutkan dengan menampilkan tabel data user.





Gambar 4.6 *Activity diagram* analisis prioritas perbaikan jalan

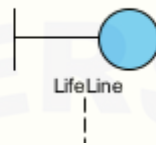
Gambar 4.6 menggambarkan alur aktivitas dari proses analisis prioritas perbaikan jalan. Analisis prioritas perbaikan jalan dilakukan tiap wilayah kecamatan, sehingga setiap wilayah kecamatan memiliki urutan prioritas masing-masing. Hasil analisis ditampilkan dalam tiga tabel yaitu tabel penilaian alternatif, tabel matrik rating kinerja dan tabel hasil nilai preferensi.

#### 4.3.5 *Sequence diagram*

*Sequence diagram* pada sistem pendukung keputusan penentuan prioritas perbaikan jalan ini digunakan untuk menggambarkan interaksi dari masing-masing komponen pada satu fungsi. Interaksi tersebut dilakukan oleh user pada sistem. Di dalam sistem sendiri juga terdapat interaksi yaitu antara view, controller dan model. Pada proses perancangan ini setiap fitur akan digambarkan ke dalam *sequence diagram*. Berikut simbol-simbol dari *sequence diagram* yang digunakan dalam intraksi view, controller dan model :

##### 1. *Boundary*

Boundary merupakan tepi dari sistem, seperti *user interface*, atau suatu alat yang berinteraksi dengan sistem lain. Dalam konsep interaksi view, controller dan model, *boundary* bisa dijadikan gambaran view pada *sequence diagram*. Contoh dari *boundary* dapat dilihat pada Gambar 4.7.



Gambar 4.7 *boundary*

## 2. *Control*

*Control* mengatur aliran dari informasi untuk sebuah skenario. Perilaku dan perilaku bisnis umumnya diatur oleh objek ini. Dalam konsep interaksi view, controller dan model, *control* bisa dijadikan gambaran *controller* pada *sequence diagram*. Contoh dari *control* dapat dilihat pada Gambar 4.8.

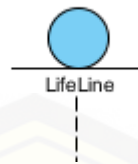


Gambar 4.8 *Control*

## 3. *Entity*

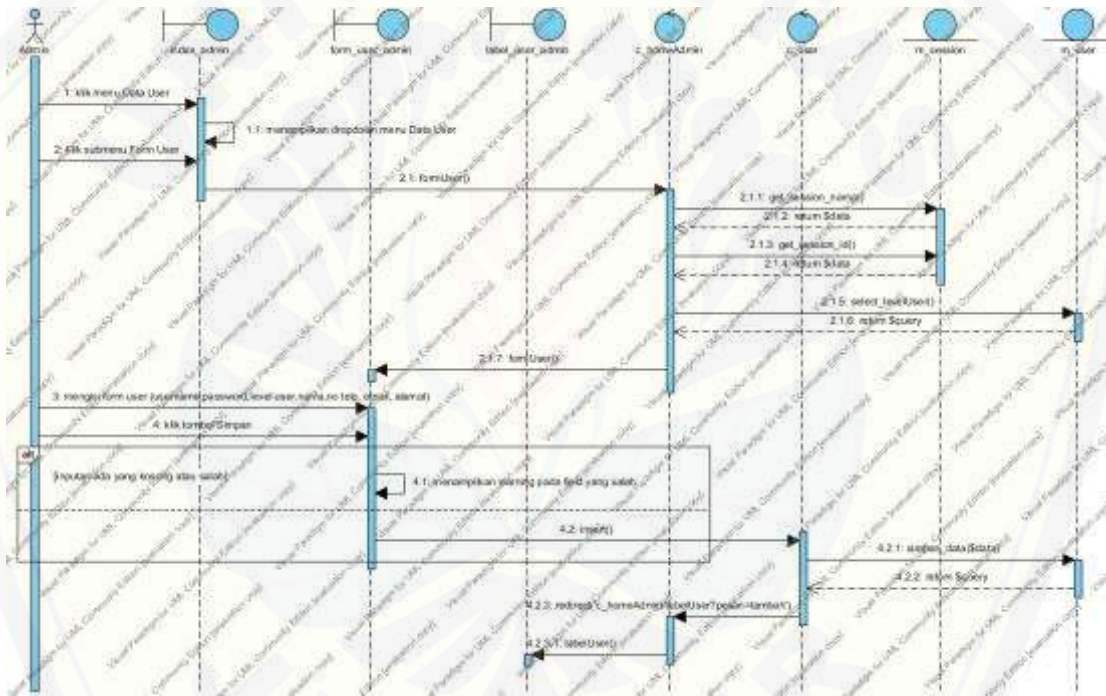
*Entity* merupakan elemen yang bertanggung jawab menyimpan data atau informasi. Disisi lain *entity* juga berinteraksi dengan *query* yang berhubungan dengan database. Dalam konsep interaksi view, controller dan model, *entity* bisa dijadikan gambaran model pada *sequence diagram*. Contoh dari *entity* dapat dilihat pada Gambar 4.9.





Gambar 4.9 Entity

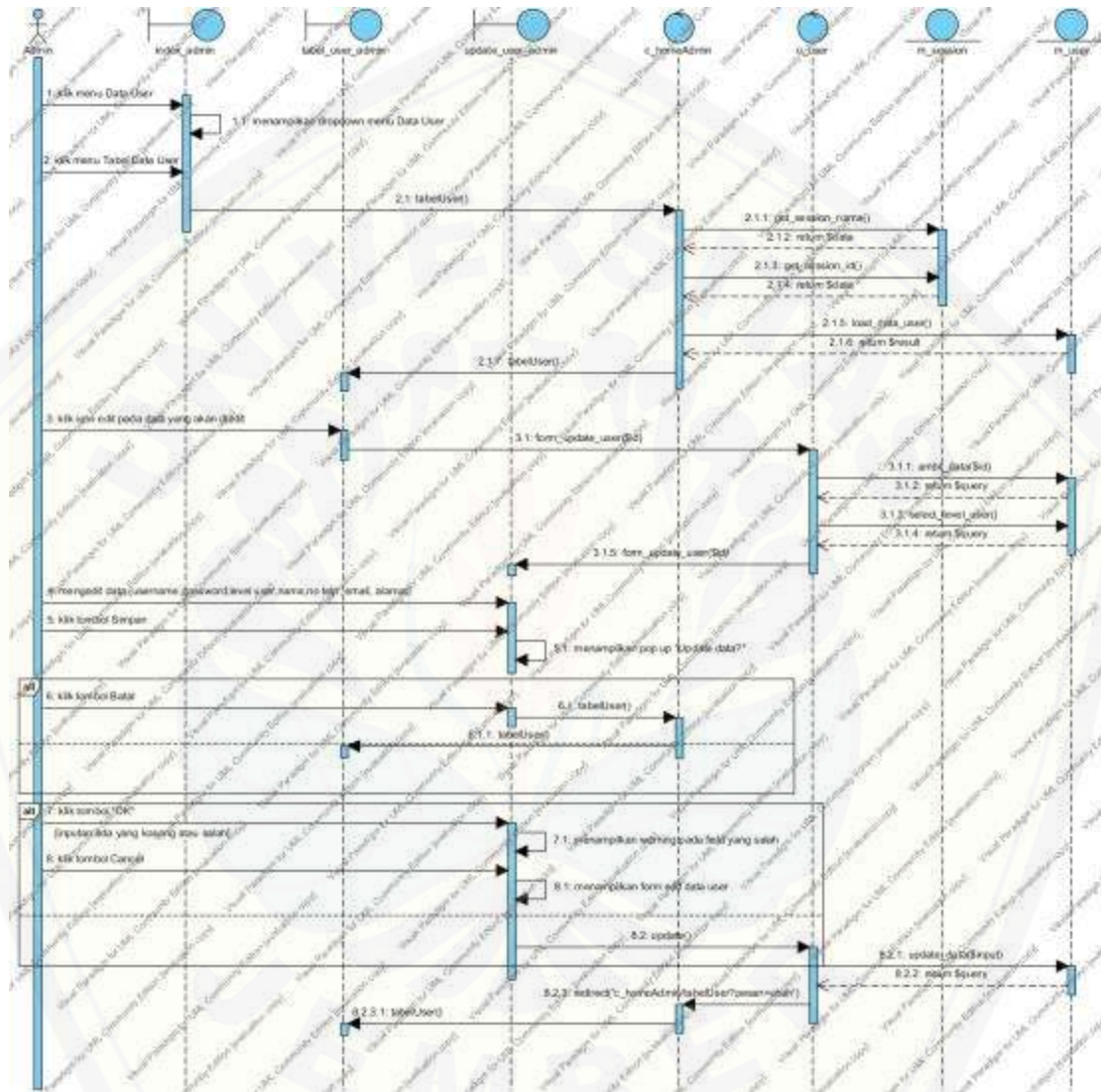
*Sequence Diagram* dari sistem pendukung keputusan penentuan prioritas perbaikan jalan adalah sebagai berikut :



Gambar 4.10 *Sequence diagram* tambah data user

Pada gambar 4.10 merupakan *sequence diagram* untuk menggambarkan skenario dan memodelkan aliran logika pada fitur menambah data user. *Sequence diagram* tambah data user menjelaskan alur MVC dari proses tambah data user. Pada *sequence diagram* tambah data user terdapat tiga *view*, dua *controller* dan dua *model*. *View* pada diagram ini adalah *index\_admin*, *form\_user\_admin* dan *tabel\_user\_admin*.

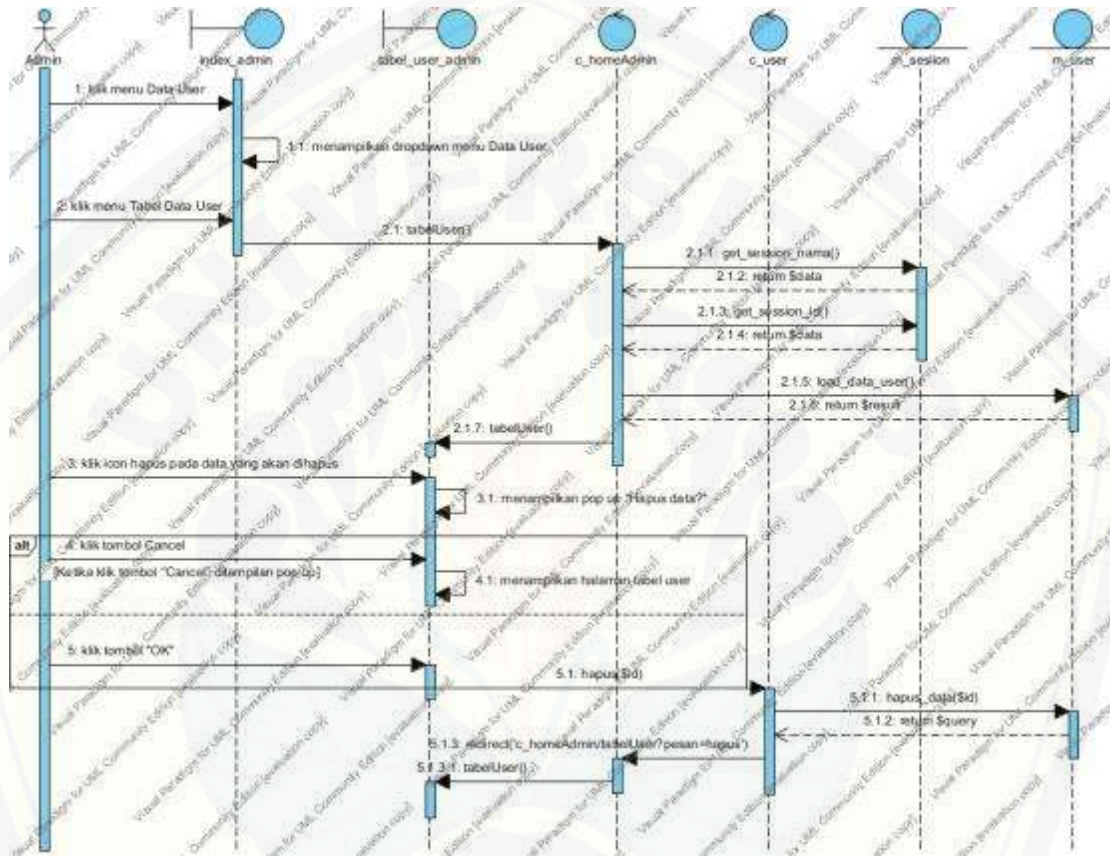
Controller pada diagram ini yaitu c\_homeAdmin dan c\_user sedangkan model pada diagram ini adalah m\_session dan m\_user.



Gambar 4.11 Sequence diagram ubah data user

Pada gambar 4.11 merupakan *sequence diagram* untuk menggambarkan skenario dan memodelkan aliran logika pada fitur ubah data user. *Sequence diagram* ubah data user menjelaskan alur MVC dari proses ubah data user. Pada *sequence diagram* ubah data user terdapat tiga *view*, dua *controller* dan dua *model*. *View* pada

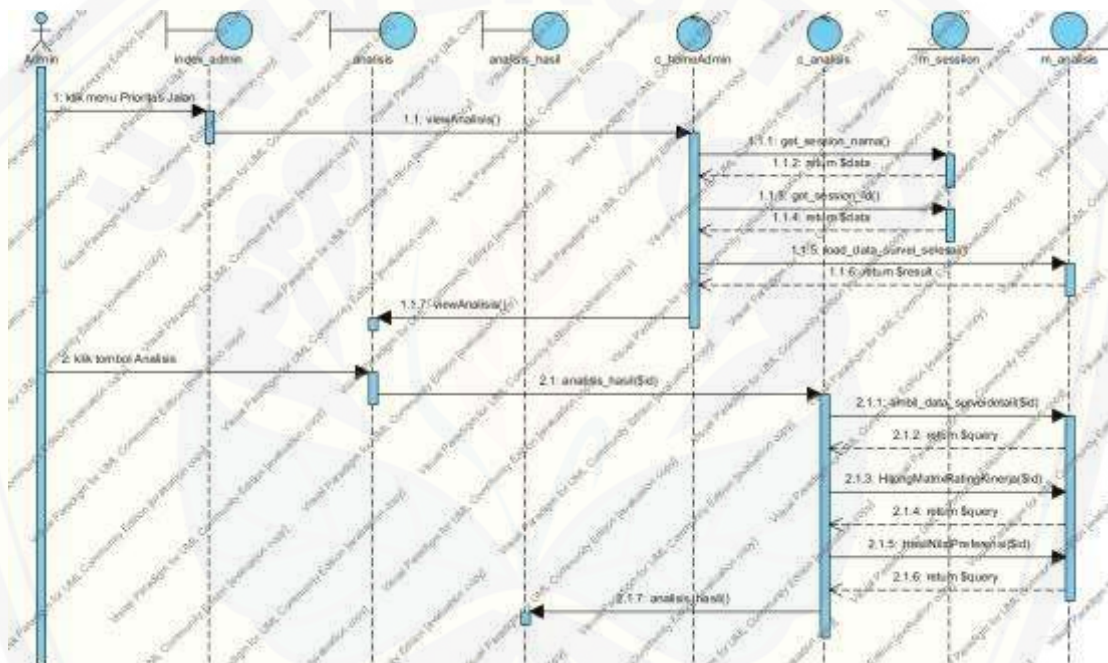
diagram ini adalah `index_admin`, `tabel_user_admin` dan `update_user_admin`. *Controller* pada diagram ini yaitu `c_homeAdmin` dan `c_user` sedangkan *model* pada diagram ini adalah `m_session` dan `m_user`.



Gambar 4.12 Sequence diagram hapus data user



Pada gambar 4.12 merupakan *sequence diagram* untuk menggambarkan skenario dan memodelkan aliran logika pada fitur hapus data user. *Sequence diagram* hapus data user menjelaskan alur MVC dari proses hapus data user. Pada *sequence diagram* hapus data user terdapat dua *view*, dua *controller* dan dua *model*. *View* pada diagram ini adalah *index\_admin*, *tabel\_user\_admin*. *Controller* pada diagram ini yaitu *c\_homeAdmin* dan *c\_user* sedangkan *model* pada diagram ini adalah *m\_session* dan *m\_user*.



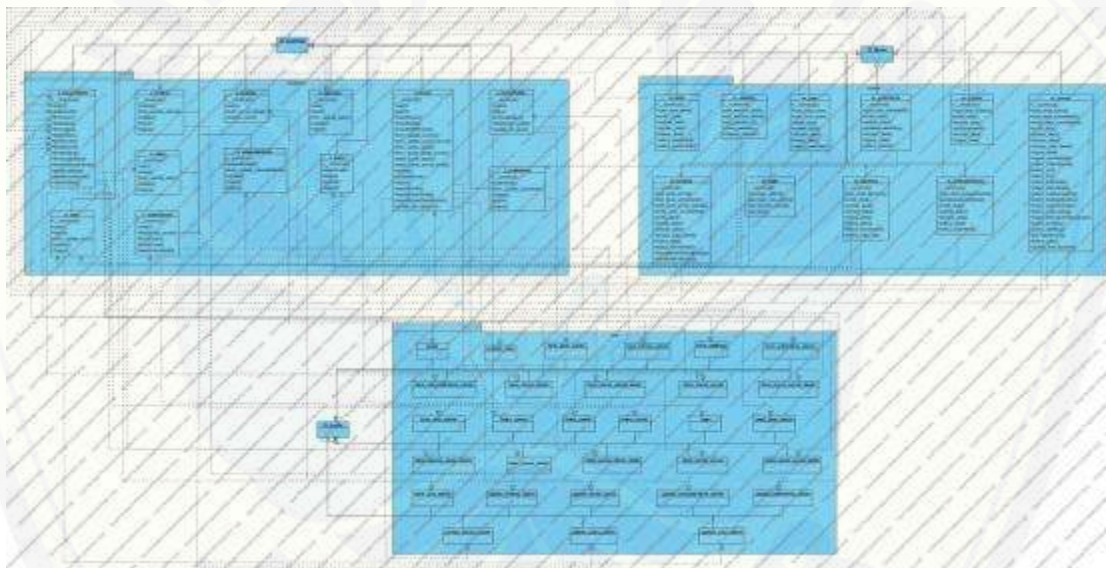
Gambar 4.13 *Sequence diagram* analisis prioritas perbaikan jalan

Pada gambar 4.13 merupakan *sequence diagram* untuk menggambarkan skenario dan memodelkan aliran logika pada fitur analisis prioritas perbaikan jalan. *Sequence diagram* analisis prioritas perbaikan jalan menjelaskan alur MVC dari proses analisis prioritas perbaikan jalan. Pada *sequence diagram* analisis prioritas perbaikan jalan terdapat tiga *view*, dua *controller* dan dua *model*. *View* pada diagram ini adalah *index\_admin*, *analisis*, *analisis\_hasil*. *Controller* pada diagram ini yaitu

c\_homeAdmin dan c\_analisis sedangkan *model* pada diagram ini adalah m\_session dan m\_analisis.

#### 4.3.6 Class diagram

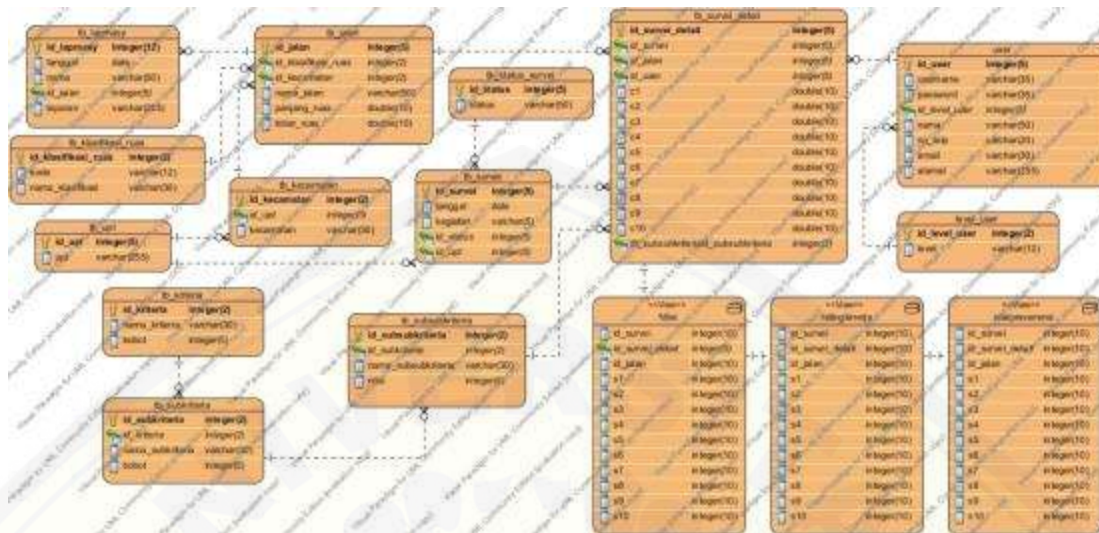
*Class diagram* menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. Selain itu *class diagram* juga menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dirancang sehingga bagaimana caranya setiap *class* saling berkolaborasi untuk mencapai sebuah tujuan. *Class diagram* sistem dapat dilihat pada gambar 4.14. Untuk detailnya terdapat pada lampiran.



Gambar 4.14 *class diagram* sistem

#### 4.3.7 Entity Relationship Diagram ( ERD )

*Entity Relationship Diagram* ( ERD ) pada sistem pendukung keputusan penentuan prioritas perbaikan jalan menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD aplikasi ditunjukkan pada gambar 4.15.



Gambar 4.15 ERD Sistem

Pada gambar 4.15 menjelaskan hubungan antar tabel dalam *database* sistem. Dalam ERD sistem menggunakan lima belas tabel yang semuanya memiliki relasi. Relasi yang digunakan adalah relasi *one to many* atau relasi satu ke banyak. Objek-objek dalam *database* sistem telah melalui proses normalisasi.

#### 4.4 Penulisan Kode Program

Setelah tahap desain selesai, tahap selanjutnya dalam penelitian ini yaitu tahap pengimplementasian desain perancangan ke dalam bahasa pemrograman. Desain sistem dari semua fitur sistem yang telah dibuat menggunakan pemodelan UML akan diimplementasikan kedalam kode program. Penulisan kode program menggunakan bahasa pemrograman *php*, *html*, dan *css* dengan *framework Code Igniter* (CI) dan menggunakan *database* My-SQL.

Sistem pendukung keputusan penentuan prioritas perbaikan jalan menggunakan metode *Simple Additive Weighting* (SAW) ini dibangun menggunakan *framework Code Igniter* (CI). Gambaran baris code dari sistem pendukung keputusan penentuan prioritas perbaikan jalan dapat dilihat pada lampiran.



## 4.5 Pengujian Sistem

Pada penelitian ini penulis menggunakan dua metode pengujian sistem yaitu *Black Box Testing* dan *White Box Testing*. Berikut adalah hasil pengujian sistem :

### 4.5.1 *White Box Testing*

Pengujian *white box* testing terdiri dari listing program, diagram alir, *cyclomatic complexity*, jalur program independen dan *test case*. Pada tahap ini fitur yang diuji adalah sebagai berikut:

#### 1. Pengujian *White Box Testing* Mengelola Data User

Pengujian *White Box Testing* mengelola data user meliputi fitur tambah data user, ubah data user, dan hapus user. Pengujian *White Box Testing* mengelola data user adalah sebagai berikut :

##### a. *Listing* program fitur mengelola data user

```

20 public function insert(){ // yg depan variabel array, yg belakang nama varia
21
22     $input ['username']           = $this->input->post('username');
23     $input ['password']          = $this->input->post('password');
24     $input ['level_user']        = $this->input->post('level_user');
25     $input ['nama']              = $this->input->post('nama');
26     $input ['no_telp']           = $this->input->post('no_telp');
27     $input ['email']             = $this->input->post('email');
28     $input ['alamat']            = $this->input->post('alamat');
29
30     // yg depan atribut database, yg belakang nama variabel array(sama kyak
31     $data = array(
32         'username'           => $input['username'],
33         'password'          => $input['password'],
34         'id_level_user'     => $input['level_user'],
35         'nama'              => $input['nama'],
36         'no_telp'           => $input['no_telp'],
37         'email'             => $input['email'],
38         'alamat'            => $input['alamat']
39     );
40
41
42     $this->m_user->simpan_data($data);
43     redirect('c_homeAdmin/tabelUser?pesan=tambah');
44
45 }

```

Gambar 4.16 *Listing* program tambah data user

```

54 public function update(){
55     $input ['id_user']           = $this->input->post('id_user');
56     $input ['username']        = $this->input->post('username');
57     $input ['password']        = $this->input->post('password');
58     $input ['level_user']      = $this->input->post('level_user');
59     $input ['nama']            = $this->input->post('nama');
60     $input ['no_telp']         = $this->input->post('no_telp');
61     $input ['email']           = $this->input->post('email');
62     $input ['alamat']          = $this->input->post('alamat');
63
64     // yg depan atribut database, yg belakang nama variabel array(sama kayak
65
66     $this->m_user->update_data($input);
67     redirect('c_homeAdmin/tabelUser?pesan=ubah');
68 }

```

Gambar 4.17 Listing program ubah data user

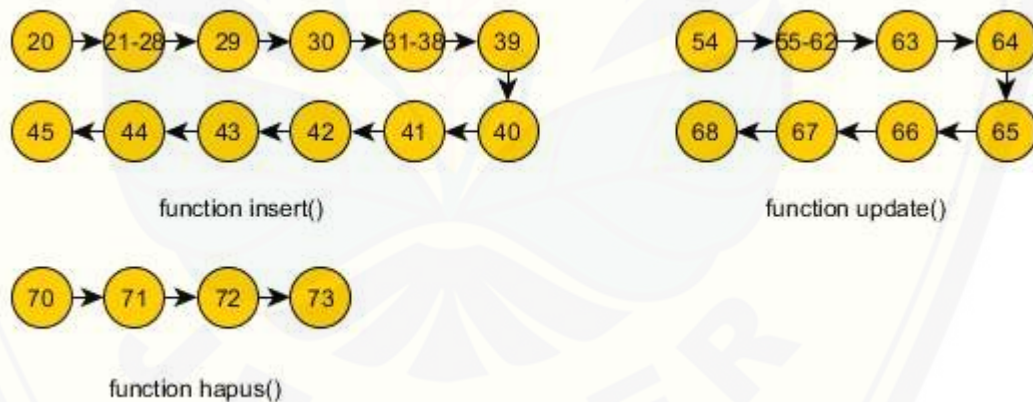
```

70 public function hapus($id=""){
71     $this->m_user->hapus_data($id);
72     redirect('c_homeAdmin/tabelUser?pesan=hapus');
73 }

```

Gambar 4.18 Listing program hapus data user

## b. Diagram alir fitur mengelola data user



Gambar 4.19 Diagram alir fitur mengelola data user

c. Perhitungan *Cyclomatic Complexity* fitur mengelola data user

Perhitungan diagram alir pada fitur mengelola data user menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

$$\text{function insert(): } V(G) = E - N + 2 = 11 - 12 + 2 = 1$$

$$\text{function update(): } V(G) = E - N + 2 = 7 - 8 + 2 = 1$$

*function* hapus() :  $V(G) = E - N + 2 = 3 - 4 + 2 = 1$

d. Pengujian jalur program fitur mengelola data user

Pengujian jalur program fitur mengelola data user berdasarkan diagram alir fitur mengelola data user :

*function* insert() : jalur : 20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-  
35-36-37-38-39-40-41-42-43-44-45

*function* update() : jalur : 54-55-56-57-58-59-60-61-62-63-64-65-66-67-68

*function* hapus() : jalur : 70-71-72-73

e. *Test Case* fitur mengelola data user

Tabel 4.7 *Test case* fitur mengelola data user

<i>Test Case function insert()</i>	
<i>Test Case</i>	Jika simpan data user berhasil
Target yang diharapkan	Menyimpan data kriteria ke <i>database</i>
Hasil pengujian	Benar
Path/Jalur	20-21-22-23-24-25-26-27-28-29-30-31-32-33- 34-35-36-37-38-39-40-41-42-43-44-45
<i>Test Case function update()</i>	
<i>Test Case</i>	Jika ubah data user berhasil
Target yang diharapkan	Mengubah data user di <i>database</i>
Hasil pengujian	Benar
Path/Jalur	54-55-56-57-58-59-60-61-62-63-64-65-66-67- 68
<i>Test Case function hapus()</i>	
<i>Test Case</i>	Jika hapus data user berhasil
Target yang diharapkan	Menghapus data user dari <i>database</i>
Hasil pengujian	Benar
Path/Jalur	70-71-72-73

2. Pengujian *White Box Testing* Mengelola Data Jalan

Pengujian *White Box Testing* mengelola data jalan meliputi fitur tambah data jalan, ubah data jalan, dan hapus data jalan. Pengujian *White Box Testing* mengelola data jalan adalah sebagai berikut :

a. *Listing* program fitur mengelola data jalan

```

21 public function insert(){ // yg depan variabel array, yg belakang nama variabel dr for
22
23     $input ['klasifikasi_ruas']           = $this->input->post('klasifikasi_ruas');
24     $input ['kecamatan']                 = $this->input->post('kecamatan');
25     $input ['nama_jalan']                = $this->input->post('nama_jalan');
26     $input ['panjang_ruas']             = $this->input->post('panjang_ruas');
27     $input ['lebar_ruas']                = $this->input->post('lebar_ruas');
28
29     // yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)
30     $data = array(
31         'id_klasifikasi_ruas'           => $input['klasifikasi_ruas'],
32         'id_kecamatan'                 => $input['kecamatan'],
33         'nama_jalan'                   => $input['nama_jalan'],
34         'panjang_ruas'                 => $input['panjang_ruas'],
35         'lebar_ruas'                   => $input['lebar_ruas']
36     );
37
38
39     $this->m_jalan->simpan_data($data);
40     redirect('c_homeAdmin/tabelJalan?pesan=tambah');
41 }

```

Gambar 4.20 Listing program tambah data jalan

```

42 public function update(){
43     $input ['id_jalan']                 = $this->input->post('id_jalan');
44     $input ['klasifikasi_ruas']         = $this->input->post('klasifikasi_ruas');
45     $input ['kecamatan']                 = $this->input->post('kecamatan');
46     $input ['nama_jalan']               = $this->input->post('nama_jalan');
47     $input ['panjang_ruas']             = $this->input->post('panjang_ruas');
48     $input ['lebar_ruas']               = $this->input->post('lebar_ruas');
49
50     // yg depan atribut database, yg belakang nama variabel array(sama kyak d
51
52     $this->m_jalan->update_data($input);
53     redirect('c_homeAdmin/tabelJalan?pesan=ubah');
54 }

```

Gambar 4.21 Listing program ubah data jalan

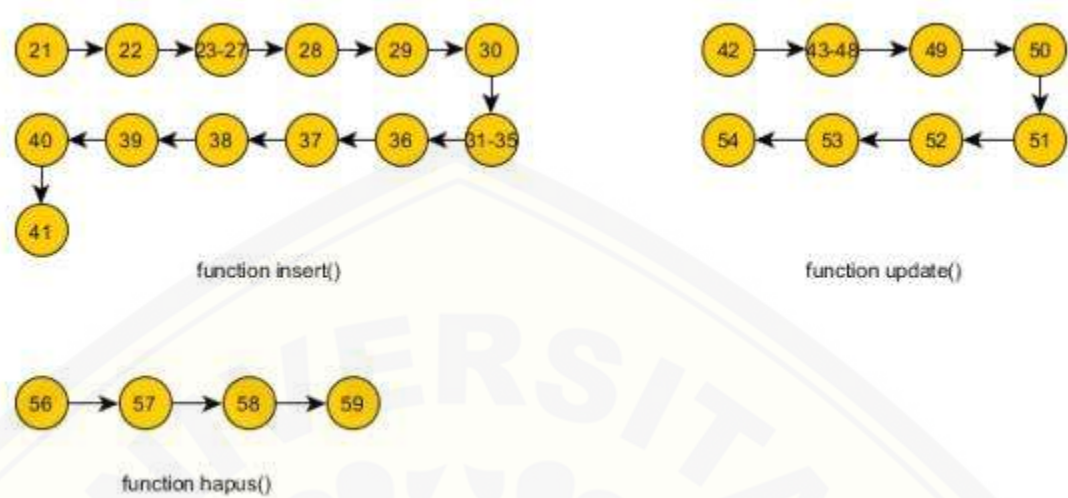
```

56 public function hapus($id=""){
57     $this->m_jalan->hapus_data($id);
58     redirect('c_homeAdmin/tabelJalan?pesan=hapus');
59 }

```

Gambar 4.22 Listing program hapus data jalan

b. Diagram alir fitur mengelola data jalan



Gambar 4.23 Diagram alir fitur mengelola data jalan

c. Perhitungan *Cyclomatic Complexity* fitur mengelola data jalan

Perhitungan diagram alir pada fitur mengelola data jalan menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

$$function\ insert() : V(G) = E - N + 2 = 12 - 13 + 2 = 1$$

$$function\ update() : V(G) = E - N + 2 = 7 - 8 + 2 = 1$$

$$function\ hapus() : V(G) = E - N + 2 = 3 - 4 + 2 = 1$$

d. Pengujian jalur program fitur mengelola data jalan

Pengujian jalur program fitur mengelola data jalan berdasarkan diagram alir fitur mengelola data jalan :

*function insert()* : jalur : 21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41

*function update()* : jalur : 42-43-44-45-46-47-48-49-50-51-52-53-54

*function hapus()* : jalur : 56-57-58-59

e. *Test Case* fitur mengelola data jalan

Tabel 4.8 *Test case* fitur mengelola data jalan

<i>Test Case function insert()</i>	
<i>Test Case</i>	Jika simpan data jalan berhasil



Target yang diharapkan	Menyimpan data jalan ke <i>database</i>
Hasil pengujian	Benar
Path/Jalur	21-22-23-24-25-26-27-28-29-30-31-32-33-34- 34-35-36-37-38-39-40-41

#### Test Case function update()

<i>Test Case</i>	Jika ubah data jalan berhasil
Target yang diharapkan	Mengubah data jalan di <i>database</i>
Hasil pengujian	Benar
Path/Jalur	42-43-44-45-46-47-48-49-50-51-52-53-54

#### Test Case function hapus()

<i>Test Case</i>	Jika hapus data jalan berhasil
Target yang diharapkan	Menghapus data jalan dari <i>database</i>
Hasil pengujian	Benar
Path/Jalur	56-57-58-59

### 3. Pengujian *White Box Testing* Analisis Prioritas Jalan

Pengujian *White Box Testing* analisis prioritas jalan meliputi *function* `ambil_data_surveidetail()`, *function* `HitungMatrikRatingKinerja()`. Pengujian *White Box Testing* analisis prioritas jalan adalah sebagai berikut :

#### a. Listing program fitur analisis prioritas jalan

```

33 function ambil_data_surveidetail($id) {
34     $result = $this->db->query("select id_survei, id_survei_detail, nama_jalan, c.nilai as s1,
35         d.nilai as s2,
36         e.nilai as s3,
37         f.nilai as s4,
38         g.nilai as s5,
39         h.nilai as s6,
40         i.nilai as s7,
41         j.nilai as s8,
42         k.nilai as s9,
43         l.nilai as s10 from tb_survei_detail a
44         JOIN tb_jalan b ON a.id_jalan = b.id_jalan
45         JOIN tb_subsubkriteria c ON a.c1 = c.id_subsubkriteria
46         JOIN tb_subsubkriteria d ON a.c2 = d.id_subsubkriteria
47         JOIN tb_subsubkriteria e ON a.c3 = e.id_subsubkriteria
48         JOIN tb_subsubkriteria f ON a.c4 = f.id_subsubkriteria
49         JOIN tb_subsubkriteria g ON a.c5 = g.id_subsubkriteria
50         JOIN tb_subsubkriteria h ON a.c6 = h.id_subsubkriteria
51         JOIN tb_subsubkriteria i ON a.c7 = i.id_subsubkriteria
52         JOIN tb_subsubkriteria j ON a.c8 = j.id_subsubkriteria
53         JOIN tb_subsubkriteria k ON a.c9 = k.id_subsubkriteria
54         JOIN tb_subsubkriteria l ON a.c10 = l.id_subsubkriteria
55         WHERE id_survei='$id'");
56     return $result;
57 }

```

Gambar 4.24 Listing program *function* `ambil_data_surveidetail()`

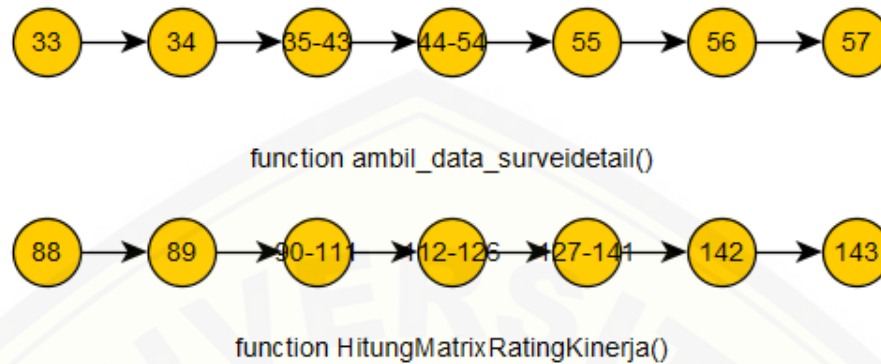
```

88 function HitungMatrikRatingKinerja($id){
89     $query1 = $this->db->query("CREATE VIEW nilai AS
90     select id_survei, id_survei_detail, b.id_jalan AS id_jalan, c.nilai as s1,
91     d.nilai as s2,
92     e.nilai as s3,
93     f.nilai as s4,
94     g.nilai as s5,
95     h.nilai as s6,
96     i.nilai as s7,
97     j.nilai as s8,
98     k.nilai as s9,
99     l.nilai as s10 from tb_survei_detail a
100     JOIN tb_jalan b ON a.id_jalan = b.id_jalan
101     JOIN tb_subsubkriteria c ON a.c1 = c.id_subsubkriteria
102     JOIN tb_subsubkriteria d ON a.c2 = d.id_subsubkriteria
103     JOIN tb_subsubkriteria e ON a.c3 = e.id_subsubkriteria
104     JOIN tb_subsubkriteria f ON a.c4 = f.id_subsubkriteria
105     JOIN tb_subsubkriteria g ON a.c5 = g.id_subsubkriteria
106     JOIN tb_subsubkriteria h ON a.c6 = h.id_subsubkriteria
107     JOIN tb_subsubkriteria i ON a.c7 = i.id_subsubkriteria
108     JOIN tb_subsubkriteria j ON a.c8 = j.id_subsubkriteria
109     JOIN tb_subsubkriteria k ON a.c9 = k.id_subsubkriteria
110     JOIN tb_subsubkriteria l ON a.c10 = l.id_subsubkriteria
111     WHERE id_survei='$id'");
112     $query2 = $this->db->query("SELECT MAX(s1) as m1, MAX(s2) as m2, MAX(s3) as m3,
113     MAX(s4) as m4, MAX(s5) as m5, MAX(s6) as m6, MAX(s7) as m7, MAX(s8) as m8,
114     MAX(s9) as m9, MAX(s10) as m10 from nilai");
115     foreach ($query2->result_array() as $max) {
116         $nilaimax1 = $max['m1'];
117         $nilaimax2 = $max['m2'];
118         $nilaimax3 = $max['m3'];
119         $nilaimax4 = $max['m4'];
120         $nilaimax5 = $max['m5'];
121         $nilaimax6 = $max['m6'];
122         $nilaimax7 = $max['m7'];
123         $nilaimax8 = $max['m8'];
124         $nilaimax9 = $max['m9'];
125         $nilaimax10 = $max['m10'];
126     }
127     $query3 = $this->db->query("CREATE VIEW ratingkinerja AS
128     SELECT id_survei, id_survei_detail, id_jalan,
129     s1/$nilaimax1 as s1,
130     s2/$nilaimax2 as s2,
131     s3/$nilaimax3 as s3,
132     s4/$nilaimax4 as s4,
133     s5/$nilaimax5 as s5,
134     s6/$nilaimax6 as s6,
135     s7/$nilaimax7 as s7,
136     s8/$nilaimax8 as s8,
137     s9/$nilaimax9 as s9,
138     s10/$nilaimax10 as s10 FROM nilai");
139
140     $query4 = $this->db->query("SELECT D.name_jalan, s1,s2,s3,s4,s5,s6,s7,s8,s9,s10 FROM ratingkinerja a
141     JOIN tb_jalan b ON a.id_jalan = b.id_jalan");
142     return $query4;
143 }

```

Gambar 4.25 Listing program *function* HitungMatrikRatingKinerja()

b. Diagram alir fitur analisis prioritas jalan



Gambar 4.26 Diagram alir fitur analisis prioritas jalan

c. Perhitungan *Cyclomatic Complexity* fitur analisis prioritas jalan

Perhitungan diagram alir pada fitur analisis prioritas jalan menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

$$\text{function ambil\_data\_surveidetail() : } V(G) = E - N + 2 = 6 - 7 + 2 = 1$$

$$\text{function HitungMatrikRatingKinerja() : } V(G) = E - N + 2 = 6 - 7 + 2 = 1$$

## d. Pengujian jalur program fitur analisis prioritas jalan

Pengujian jalur program fitur analisis prioritas jalan berdasarkan diagram alir fitur analisis prioritas jalan :

*function* ambil\_data\_surveidetail() : jalur : 33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57

*function* HitungMatrikRatingKinerja() : jalur : 88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143

e. *Test Case* fitur analisis prioritas jalanTabel 4.9 *Test case* fitur analisis prioritas jalan

<i>Test Case</i> function ambil_data_surveidetail()
---

<i>Test Case</i>	Jika simpan data survei detail berhasil
Target yang diharapkan	Mengambil data dari <i>database</i>
Hasil pengujian	Benar
Path/Jalur	33-34-35-36-37-38-39-40-41-42-43-44-45-46- 47-48-49-50-51-52-53-54-55-56-57

#### *Test Case* function HitungMatrikRatingKinerja()

<i>Test Case</i>	Jika menghitung matrik rating kinerja berhasil
Target yang diharapkan	Menghitung matrik rating kinerja
Hasil pengujian	Benar
Path/Jalur	88-89-90-91-92-93-94-95-96-97-98-99-100- 101-102-103-104-105-106-107-108-109-110- 111-112-113-114-115-116-117-118-119-120- 121-123-124-125-126-127-128-129-130-131- 132-133-134-135-136-137-138-139-140-141- 142-143

#### 4. Pengujian *White Box Testing* menambahkan data laporan masyarakat

Pengujian *White Box Testing* menambahkan data laporan masyarakat terdapat pada *function* insert(). Pengujian *White Box Testing* menambahkan data laporan masyarakat adalah sebagai berikut :

##### a. Listing program fitur menambahkan data laporan masyarakat

```

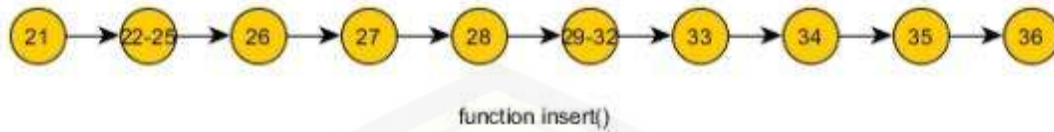
21 public function insert(){ // yg depan variabel aray, yg belakang nama variabel dr form
22     $input ['tanggal']     = $this->input->post('tgl');
23     $input ['nama']       = $this->input->post('nama');
24     $input ['id_jalan']   = $this->input->post('id_jalan');
25     $input ['laporan']    = $this->input->post('laporan');
26
27     // yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)
28     $data = array(
29         'tanggal'         => $input['tanggal'],
30         'nama'           => $input['nama'],
31         'id_jalan'       => $input['id_jalan'],
32         'laporan'        => $input['laporan']
33     );
34     $this->m_lapmasy->simpan_data($data);
35     redirect('c_homePublic/formLapMasy?pesan=berhasil');
36 }

```

Gambar 4.27 Listing program fitur menambahkan data laporan masyarakat



b. Diagram alir fitur menambahkan data laporan masyarakat



Gambar 4.28 Diagram alir fitur menambahkan data laporan masyarakat

c. Perhitungan *Cyclomatic Complexity* fitur menambahkan data laporan masyarakat

Perhitungan diagram alir pada fitur menambahkan data laporan masyarakat menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

$$function\ insert() : V(G) = E - N + 2 = 9 - 10 + 2 = 1$$

d. Pengujian jalur program fitur menambahkan data laporan masyarakat

Pengujian jalur program fitur menambahkan data laporan masyarakat berdasarkan diagram alir fitur menambahkan data laporan masyarakat :

$$function\ insert() : jalur : 21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36$$

e. *Test Case* fitur menambahkan data laporan masyarakat

Tabel 4.10 *Test case* fitur menambahkan data laporan masyarakat

<i>Test Case function insert()</i>	
<i>Test Case</i>	Jika berhasil menambahkan laporan
Target yang diharapkan	Menyimpan data laporan masyarakat kedalam <i>database</i>
Hasil pengujian	Benar
Path/Jalur	21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36

#### 4.5.2 *Black Box Testing*

Pengujian *black box* menitik beratkan pada fungsionalitas sistem. Pengujian *black box* merupakan pengujian yang dilakukan oleh user dan aplikasi yang dibangun pada penelitian ini akan diuji dengan mengujikan langsung *running aplikasi* dan melakukan kegiatan pengujian dengan menganalisis proses input dan output yang



dihasilkan aplikasi. Dokumentasi hasil pengujian dapat dilihat pada tabel-tabel dibawah ini.

Tabel 4.11 Pengujian *black box* fitur mengelola data user

No	Fitur	Kasus	Hasil	Keterangan
1.	Tambah data user (user admin)	<ul style="list-style-type: none"> <li>Ketika klik tombol simpan dan semua <i>field</i> sudah terisi dengan benar</li> <li>Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> </ul>	<ul style="list-style-type: none"> <li>Menyimpan data dan menampilkan alert success pada halaman tabel data user</li> <li>Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> </ul>	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
2	Ubah data user (user admin)	<ul style="list-style-type: none"> <li>Ketika klik tombol simpan</li> <li>Ketika klik tombol OK pada <i>pop up</i> dan semua <i>field</i> sudah terisi dengan benar</li> <li>Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> <li>Ketika klik tombol batal</li> <li>Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan <i>pop up</i> "Update data?"</li> <li>Menyimpan data dan menampilkan alert success pada halaman tabel data user</li> <li>Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> <li>Menampilkan halaman tabel data user</li> <li>Menampilkan halaman form edit data user</li> </ul>	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal
3	Hapus data	<ul style="list-style-type: none"> <li>Klik <i>icon</i> hapus "gambar</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan <i>pop</i></li> </ul>	

user (user admin)	sampah” pada data yang akan dihapus	<i>up</i> “Hapus data?”	[ √ ] Berhasil [ ] Gagal
	<ul style="list-style-type: none"> <li>• Ketika klik tombol OK pada <i>pop up</i></li> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menghapus data dan menampilkan alert success pada halaman tabel data user</li> <li>• Menampilkan halaman tabel data user</li> </ul>	

Tabel 4.11 merupakan hasil pengujian *Black Box Testing* pada fitur mengelola data user. Hasil pengujian menunjukkan bahwa hasil yang sesuai dengan yang diharapkan oleh pengguna.

Tabel 4.12 Pengujian *black box* fitur mengelola data jalan

No	Fitur	Kasus	Hasil	Keterangan
1.	Tambah data jalan (user admin)	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan dan semua <i>field</i> sudah terisi dengan benar</li> <li>• Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> </ul>	<ul style="list-style-type: none"> <li>• Menyimpan data dan menampilkan alert success pada halaman tabel data jalan</li> <li>• Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> </ul>	[ √ ] Berhasil [ ] Gagal
2	Ubah data jalan (user admin)	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan</li> <li>• Ketika klik tombol OK pada <i>pop up</i> dan semua <i>field</i> sudah terisi dengan</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>pop up</i> “Update data?”</li> <li>• Menyimpan data dan menampilkan alert success pada</li> </ul>	[ √ ] Berhasil

	benar	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> <li>• Ketika klik tombol batal</li> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<p>halaman tabel data jalan <input type="checkbox"/> Gagal</p> <ul style="list-style-type: none"> <li>• Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> <li>• Menampilkan halaman tabel data jalan</li> <li>• Menampilkan halaman form edit data jalan</li> </ul>
3	Hapus data jalan (user admin)	<ul style="list-style-type: none"> <li>• Klik <i>icon</i> hapus “gambar sampah” pada data yang akan dihapus</li> <li>• Ketika klik tombol OK pada <i>pop up</i></li> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<p>Menampilkan <i>pop up</i> “Hapus data?” <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal</p> <ul style="list-style-type: none"> <li>• Menghapus data dan menampilkan alert success pada halaman tabel data jalan</li> <li>• Menampilkan halaman tabel data jalan</li> </ul>

Tabel 4.12 merupakan hasil pengujian *Black Box Testing* pada fitur mengelola data jalan. Hasil pengujian menunjukkan bahwa hasil yang sesuai dengan yang diharapkan oleh pengguna. Pengujian *black box testing* selengkapnya dapat dilihat pada lampiran.

## BAB 6. PENUTUP

Pada bab ini merupakan bagian akhir di dalam penulisan skripsi, berisi tentang kesimpulan dan saran. Kesimpulan yang ditulis merupakan kesimpulan dari hasil penelitian yang telah dilakukan dan saran yang diharapkan dapat digunakan sebagai acuan untuk penelitian selanjutnya.

### 6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Metode *Simple Additive Weighting* (SAW) dapat diterapkan dalam penentuan prioritas perbaikan jalan. Kriteria yang dibutuhkan yaitu lalulintas harian rata-rata, kondisi jalan dan laporan masyarakat. Hasil dari pengujian alternatif menggunakan metode ini menunjukkan bahwa jalan-jalan yang menjadi prioritas utama merupakan jalan dengan tingkat lalulintas harian rata-rata yang tinggi dan kondisi jalan yang rusak berat, sehingga kriteria yang paling menentukan pada kasus ini yaitu LHR dan kondisi jalan daripada laporan masyarakat.
2. Sistem pendukung keputusan penentuan prioritas perbaikan jalan kabupaten jember menggunakan metode *Simple Additive Weighting* (SAW) memiliki tiga hak akses yaitu admin, surveyor dan publik. Peran utama admin dalam sistem ini yaitu menganalisis urutan prioritas perbaikan jalan dari hasil survei kondisi masing-masing jalan yang diinputkan oleh surveyor. Adanya laporan kerusakan jalan oleh *user* publik dapat mempengaruhi penilaian penentuan prioritas perbaikan jalan. Sistem pendukung keputusan penentuan prioritas perbaikan jalan yang dibangun sesuai dengan kebutuhan dari penelitian dan dapat membantu DPU Bina Marga Kabupaten Jember untuk menentukan urutan prioritas perbaikan jalan. Pengembangan sistem ini sudah sesuai dengan alur model *waterfall* sehingga dokumentasi pengembangan dapat terorganisir dengan baik.

## 6.2 Saran

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

1. Melakukan penerapan metode lain untuk penentuan prioritas perbaikan jalan agar dapat dilakukan perbandingan antar metode.
2. Dapat ditambahkan fitur untuk manajemen anggaran proyek perbaikan jalan sehingga dapat menghitung rancangan pembiayaan suatu proyek.
3. Pengembangan lebih lanjut sistem ini dapat dilakukan dengan membuat aplikasi berbasis *mobile* agar lebih memudahkan petugas lapangan dalam manajemen data proyek.



**DAFTAR PUSTAKA**

- Asmoro, D. (1990). *Tata Cara Penyusunan Program Pemeliharaan Jalan Kota*. Jakarta: Direktorat Pembinaan Jalan Kota, Direktorat Jendral Bina Marga.
- Darmastuti, D. (2013). *Implementasi Metode Simple Additive Weighting Dalam Sistem Informasi Lowongan Kerja Berbasis WEB Untuk Rekomendasi Pencari Kerja Terbaik*. Tanjungpura: Skripsi : Jurusan Teknik Elektro Fakultas Teknik Universitas Tanjungpura.
- Davis, G. B. (1991). *Kerangka Dasar Sistem Informasi Manajemen Bagian 1*. Jakarta: PT Pustaka Binamas Pressindo.
- Faiz, A. H. (2009). Metode Analitical Hierarchy Process untuk Menentukan Prioritas Penanganan Jalan di Wilayah Balai Pemeliharaan Jalan Mojokerto. *Jurnal Aplikasi, Volumel 6*, 29-40.
- Indonesia, P. R. (2004). *Undang - Undang Nomor 38 Tahun 2004 tentang Jalan*. Jakarta: Sekretariat Negara Republik Indonesia.
- Keen, H. d. (1996). *Information Technology: Tomorrow's Advantage Today*. Hammond: Mcgraw-Hill College.
- Kusumadewi, S. (2006). *Fuzzy Multi Attribute Decision Making*. Yogyakarta: Graha Ilmu.
- Morton, P. G. (1978). *Decision Support Systems: An Organizational Perspective*. Reading MA: Addison-Wesley.
- Muhyuzir, T. D. (2001). *Analisa Perancangan Sistem Pengolahan Data*. Jakarta: PT. Alex Media Komputindo.
- Pressman, R. S. (2005). *Rekayasa Perangkat Lunak*. Yogyakarta: ANDI.
- Purwati, Y. (2013). *Sistem Pendukung Keputusan Berbasis Web Untuk Pemilihan Jodoh Islami Dengan Menggunakan Metode SIMPLE ADDITIVE WEIGHTING(SAW)*. Yogyakarta: Skripsi : Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta.

Sommerville, I. (2011). *Software Engineering 9th Edition*. United State of America: Addison-Wesley Publishing Company Inc.

Turban, E. (2001). *Decision Support System*. Yogyakarta: Penerbit Indonesia ANDI.



## LAMPIRAN A. PERANCANGAN SISTEM

A.1 Perancangan *Scenario*1. *Scenario* tambah data jalan

---

<b>Nama Usecase</b>	: Mengelola data jalan
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil menambah data jalan baru

---

**SKENARIO UTAMA**


---

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Jalan”	2. Menampilkan <i>dropdown</i> menu “Data Jalan”
3. Klik menu “Form Jalan”	4. Menampilkan halaman form jalan
5. Mengisi form jalan	7. Menyimpan data ke database
6. Klik tombol “Simpan”	8. Menampilkan <i>alert success</i> pada halaman tabel data jalan

---

**SKENARIO ALTERNATIF**


---

6a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.	7a. Menampilkan <i>warning message</i> pada <i>field</i> yang salah
---	---

---

2. *Scenario* ubah data jalan

---

<b>Nama Usecase</b>	: Mengelola data jalan
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil mengubah data jalan

---

**SKENARIO UTAMA**


---

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Jalan”	2. Menampilkan <i>dropdown</i> menu “Data Jalan”

3. Klik menu “Tabel Data Jalan”
4. Menampilkan halaman tabel data jalan
5. Klik *icon* edit “gambar pensil” pada data yang akan diedit
6. Menampilkan form edit data jalan
7. Mengisi form edit data jalan
8. Klik tombol “Simpan”
9. Menampilkan *pop up* “Update data?”
10. Klik tombol “OK”
11. Menyimpan data ke database
12. Menampilkan *alert success* pada halaman tabel data jalan

**SKENARIO ALTERNATIF**

- 8a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.
  - 9a. Menampilkan *warning message* pada *field* yang salah
- 8b. Ketika klik tombol “Batal”
  - 9b. Menampilkan halaman tabel data jalan
- 10c. Ketika klik tombol “Cancel” ditampilkan *pop up*
  - 11c. Menampilkan form edit data jalan

3. *Scenario* hapus data jalan

<b>Nama Usecase</b>	: Mengelola data jalan
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil menghapus data jalan

**SKENARIO UTAMA**

- | <i>Actor</i>              | <i>System</i>                                    |
|---------------------------|--|
| 1. Klik menu “Data Jalan” | 2. Menampilkan <i>dropdown</i> menu “Data Jalan” |
| 3. Klik menu “Tabel Data  |  |

Jalan”

5. Klik *icon* hapus “gambar sampah” pada data yang akan dihapus
7. Klik tombol “OK”
4. Menampilkan halaman tabel data jalan
6. Menampilkan *pop up* “Hapus data?”
8. Menghapus data jalan dari database
9. Menampilkan *alert success* pada halaman tabel data jalan

#### SKENARIO ALTERNATIF

8a. Ketika klik tombol “Cancel” ditampilkan *pop up*

9a. Menampilkan halaman tabel data jalan

4. *Scenario* tambah data survei

<b>Nama Usecase</b>	: Mengelola data survei
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil menambah data survei baru

#### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Survei”	2. Menampilkan <i>dropdown</i> menu “Data Survei”
3. Klik menu “Form Survei”	4. Menampilkan halaman form survei
5. Mengisi form survei	7. Menyimpan data ke database
6. Klik tombol “Simpan”	8. Menampilkan <i>alert success</i> pada halaman tabel data survei

#### SKENARIO ALTERNATIF

6a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.

7a. Menampilkan *warning message* pada *field* yang salah



5. *Scenario* ubah data survei

<b>Nama Usecase</b>	: Mengelola data survei
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil mengubah data survei

**SKENARIO UTAMA**

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Survei”	2. Menampilkan <i>dropdown</i> menu “Data Survei”
3. Klik menu “Tabel Data Survei”	4. Menampilkan halaman tabel data survei
5. Klik <i>icon</i> edit “gambar pensil” pada data yang akan diedit	6. Menampilkan form edit data survei
7. Mengisi form edit data survei	9. Menampilkan <i>pop up</i> “Update data?”
8. Klik tombol “Simpan”	11. Menyimpan data ke database
10. Klik tombol “OK”	12. Menampilkan <i>alert success</i> pada halaman tabel data survei

**SKENARIO ALTERNATIF**

8a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.	9a. Menampilkan <i>warning message</i> pada <i>field</i> yang salah
8b. Ketika klik tombol “Batal”	9b. Menampilkan halaman tabel data survei
10c. Ketika klik tombol “Cancel” ditampilkan <i>pop up</i>	11c. Menampilkan form edit data survei

6. *Scenario* hapus data survei

---

<b>Nama Usecase</b>	: Mengelola data survei
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil menghapus data survei

#### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Survei”	2. Menampilkan <i>dropdown</i> menu “Data Survei”
3. Klik menu “Tabel Data Survei”	4. Menampilkan halaman tabel data survei
5. Klik <i>icon</i> hapus “gambar sampah” pada data yang akan dihapus	6. Menampilkan <i>pop up</i> “Hapus data?”
7. Klik tombol “OK”	8. Menghapus data survei dari database
	9. Menampilkan <i>alert success</i> pada halaman tabel data survei

#### SKENARIO ALTERNATIF

8a. Ketika klik tombol “Cancel” ditampilkan <i>pop up</i>	9a. Menampilkan halaman tabel data survei
---	---

---

#### 7. *Scenario* tambah laporan masyarakat

---

<b>Nama Usecase</b>	: Menambah laporan masyarakat
<b>Aktor</b>	: Publik
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Publik mengakses sistem
<b>Pasca-Kondisi</b>	: Publik berhasil menambah data laporan masyarakat

#### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Laporkan?”	2. Menampilkan halaman form laporan masyarakat
3. Mengisi form laporan masyarakat	

4. Klik tombol “Simpan”
5. Menyimpan data ke database
6. Menampilkan *alert success* pada halaman form laporan masyarakat

#### SKENARIO ALTERNATIF

- 4a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.
- 5a. Menampilkan *warning* pada *field* yang salah

#### 8. Scenario tambah data survey detail admin

<b>Nama Usecase</b>	: Mengelola data survei detail
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil menambah data survei detail baru

#### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Survei”	2. Menampilkan <i>dropdown</i> menu “Data Survei”
3. Klik menu “Tabel Data Survei”	4. Menampilkan halaman tabel data survei
5. Klik <i>link</i> “input survei detail”	6. Menampilkan halaman form survei detail
7. Mengisi form survei detail	9. Menyimpan data ke database
8. Klik tombol “Simpan”	10. Menampilkan <i>alert success</i> pada halaman tabel data survei detail

#### SKENARIO ALTERNATIF

- 8a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.
- 9a. Menampilkan *warning message* pada *field* yang salah

8b. Ketika klik tombol “Batal”

9b. Menampilkan halaman tabel data survei

9. *Scenario* hapus data survey detail admin

<b>Nama Usecase</b>	: Mengelola data survei detail
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil menghapus data survei detail

### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Survei”	2. Menampilkan <i>dropdown</i> menu “Data Survei”
3. Klik menu “Tabel Data Survei”	4. Menampilkan halaman tabel data survei
5. Klik <i>icon</i> cari “gambar kaca pembesar” pada data yang akan dipilih	6. Menampilkan halaman tabel data survei detail
7. Klik <i>icon</i> hapus “gambar sampah” pada data yang akan dihapus	8. Menampilkan <i>pop up</i> “Hapus data?”
9. Klik tombol “OK”	10. Menghapus data dari database
	11. Menampilkan <i>alert success</i> pada halaman tabel data survei detail

### SKENARIO ALTERNATIF

9a. Ketika klik tombol “Cancel” ditampilkan *pop up*

10a. Menampilkan halaman tabel data survei detail

10. *Scenario* tambah data survey detail surveyor

<b>Nama Usecase</b>	: Mengelola data survei detail
---------------------	--------------------------------

<b>Aktor</b>	: Surveyor
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Surveyor login kedalam sistem
<b>Pasca-Kondisi</b>	: Surveyor berhasil menambah data survei detail baru

### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Survei”	2. Menampilkan <i>dropdown</i> menu “Data Survei”
3. Klik menu “Tabel Data Survei”	4. Menampilkan halaman tabel data survei
5. Klik <i>link</i> “input survei detail”	6. Menampilkan halaman form survei detail
7. Mengisi form survei detail	9. Menyimpan data ke database
8. Klik tombol “Simpan”	10. Menampilkan <i>alert success</i> pada halaman tabel data survei detail

### SKENARIO ALTERNATIF

8a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.	9a. Menampilkan <i>warning message</i> pada <i>field</i> yang salah
8b. Ketika klik tombol “Batal”	9b. Menampilkan halaman tabel data survei

### 11. *Scenario* hapus data survei detail surveyor

<b>Nama Usecase</b>	: Mengelola data survei detail
<b>Aktor</b>	: Surveyor
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Surveyor login kedalam sistem
<b>Pasca-Kondisi</b>	: Surveyor berhasil menghapus data survei detail

### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
--------------	---------------



1. Klik menu “Data Survei”
2. Menampilkan *dropdown* menu “Data Survei”
3. Klik menu “Tabel Data Survei”
4. Menampilkan halaman tabel data survei
5. Klik *icon* cari “gambar kaca pembesar” pada data yang akan dipilih
6. Menampilkan halaman tabel data survei detail
7. Klik *icon* hapus “gambar sampah” pada data yang akan dihapus
8. Menampilkan *pop up* “Hapus data?”
9. Klik tombol “OK”
10. Menghapus data dari database
11. Menampilkan *alert success* pada halaman tabel data survei detail

**SKENARIO ALTERNATIF**

- 8a. Ketika klik tombol “Cancel” ditampilkan *pop up*
- 9a. Menampilkan halaman tabel data survei detail

12. *Scenario* ubah status survei

- |                      |                                       |
|----------------------|---------------------------------------|
| <b>Nama Usecase</b>  | : Mengubah status survei              |
| <b>Aktor</b>         | : Admin                               |
| <b>Triger</b>        | : -                                   |
| <b>Pra-Kondisi</b>   | : Admin login kedalam sistem          |
| <b>Pasca-Kondisi</b> | : Admin berhasil mengubah data survei |

**SKENARIO UTAMA**

- | <i>Actor</i>  | <i>System</i>                                     |
|---|---|
| 1. Klik menu “Data Survei”                              | 2. Menampilkan <i>dropdown</i> menu “Data Survei” |
| 3. Klik menu “Tabel Data Survei”                        | 4. Menampilkan halaman tabel data survei          |
| 5. Klik <i>icon</i> edit “gambar pensil” pada data yang |   |

akan diedit

- 7. Mengisi form edit data survei
- 8. Klik tombol “Simpan”
- 10. Klik tombol “OK”
- 6. Menampilkan form edit data survei
- 9. Menampilkan *pop up* “Update data?”
- 11. Menyimpan data ke database
- 12. Menampilkan *alert success* pada halaman tabel data survei

**SKENARIO ALTERNATIF**

- 8a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.
  - 9a. Menampilkan *warning message* pada *field* yang salah
- 8b. Ketika klik tombol “Batal”
  - 9b. Menampilkan halaman tabel data survei
- 10c. Ketika klik tombol “Cancel” ditampilkan *pop up*
  - 11c. Menampilkan form edit data survei

13. *Scenario* ubah bobot kriteria

<b>Nama Usecase</b>	: Mengubah bobot kriteria
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil mengubah data kriteria

**SKENARIO UTAMA**

- | <i>Actor</i>                 | <i>System</i>                                       |
|------------------------------|---|
| 1. Klik menu “Data Kriteria” | 2. Menampilkan <i>dropdown</i> menu “Data Kriteria” |
| 3. Klik menu “Kriteria”      | 4. Menampilkan halaman tabel data kriteria          |
| 5. Klik tombol “Edit Bobot”  | 6. Menampilkan form edit bobot kriteria             |
| 7. Mengisi form edit bobot   |   |

- kriteria
8. Klik tombol “Simpan”
  10. Klik tombol “OK”
  9. Menampilkan *pop up* “Update data?”
  11. Menyimpan data ke database
  12. Menampilkan *alert success* pada halaman tabel data kriteria

#### SKENARIO ALTERNATIF

- 8a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.
  - 9a. Menampilkan *warning message* pada *field* yang salah
- 8b. Ketika klik tombol “Batal”
  - 9b. Menampilkan halaman tabel data kriteria
- 10c. Ketika klik tombol “Cancel” ditampilkan *pop up*
  - 11c. Menampilkan form edit bobot kriteria
- 10d. Ketika klik tombol “OK” dan jumlah total bobot melebihi 100%
  - 11d. Menampilkan *alert error* pada halaman tabel data kriteria
- 10e. Ketika klik tombol “OK” dan jumlah total bobot kurang dari 100%
  - 11e. Menampilkan *alert error* pada halaman tabel data kriteria

#### 14. Scenario ubah bobot subkriteria

<b>Nama Usecase</b>	: Mengubah bobot subkriteria
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil mengubah data subkriteria

#### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Kriteria”	2. Menampilkan <i>dropdown</i> menu “Data Kriteria”
3. Klik menu “Subkriteria”	4. Menampilkan halaman tabel data subkriteria
5. Klik tombol “Edit Bobot Kriteria Kondisi Jalan”	6. Menampilkan form edit bobot subkriteria
7. Mengisi form edit bobot subkriteria	
8. Klik tombol “Simpan”	9. Menampilkan <i>pop up</i> “Update data?”
10. Klik tombol “OK”	11. Menyimpan data ke database
	12. Menampilkan <i>alert success</i> pada halaman tabel data subkriteria

#### SKENARIO ALTERNATIF

8a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.	9a. Menampilkan <i>warning message</i> pada <i>field</i> yang salah
8b. Ketika klik tombol “Batal”	9b. Menampilkan halaman tabel data subkriteria
10c. Ketika klik tombol “Cancel” ditampilkan <i>pop up</i>	11c. Menampilkan form edit bobot subkriteria
10d. Ketika klik tombol “Simpan” dan jumlah total bobot melebihi 100%	11d. Menampilkan <i>alert error</i> pada halaman tabel data kriteria
10e. Ketika klik tombol	

“Simpan” dan jumlah total bobot kurang dari 100%

11e. Menampilkan *alert error* pada halaman tabel data kriteria

15. *Scenario* ubah bobot nilai subkriteria

<b>Nama Usecase</b>	: Mengubah data bobot nilai subkriteria
<b>Aktor</b>	: Admin
<b>Triger</b>	: -
<b>Pra-Kondisi</b>	: Admin login kedalam sistem
<b>Pasca-Kondisi</b>	: Admin berhasil mengubah data bobot nilai subkriteria

### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Data Kriteria”	2. Menampilkan <i>dropdown</i> menu “Data Kriteria”
3. Klik menu “Bobot Nilai”	4. Menampilkan halaman tabel data bobot nilai subkriteria
5. Klik <i>icon</i> edit “gambar pensil” pada data yang akan diedit	6. Menampilkan form edit data bobot nilai subkriteria
7. Mengisi form edit data bobot nilai subkriteria	9. Menampilkan <i>pop up</i> “Update data?”
8. Klik tombol “Simpan”	11. Menyimpan data ke database
10. Klik tombol “OK”	12. Menampilkan <i>alert success</i> pada halaman tabel data bobot nilai subkriteria

### SKENARIO ALTERNATIF

8a. Ketika klik tombol “Simpan” dan inputan ada yang kosong atau salah.	9a. Menampilkan <i>warning message</i> pada <i>field</i> yang salah
---	---



8b. Ketika klik tombol “Batal”

9b. Menampilkan halaman tabel data bobot nilai subkriteria

10c. Ketika klik tombol “Cancel” ditampilkan *pop up*

11c. Menampilkan form edit data bobot nilai subkriteria

16. *Scenario* lihat data laporan masyarakat

<b>Nama Usecase</b>	:	Melihat laporan masyarakat
<b>Aktor</b>	:	Admin
<b>Triger</b>	:	-
<b>Pra-Kondisi</b>	:	Admin login kedalam sistem
<b>Pasca-Kondisi</b>	:	Admin berhasil melihat laporan masyarakat

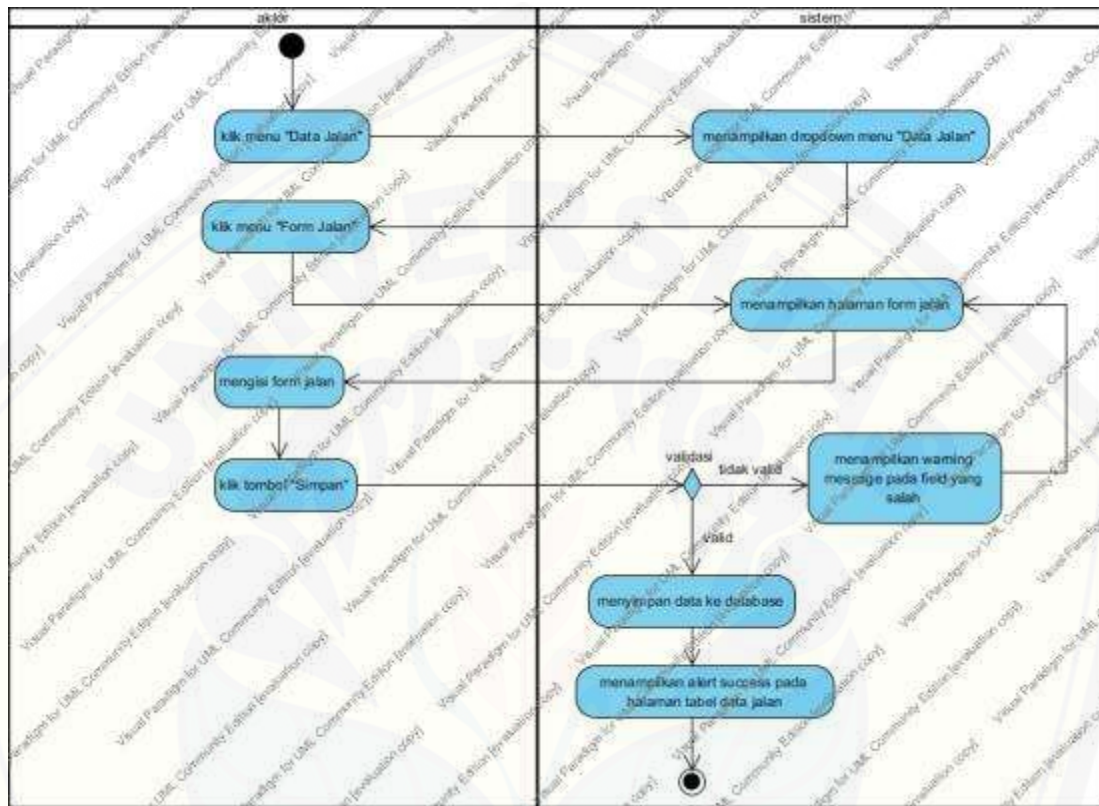
#### SKENARIO UTAMA

<i>Actor</i>	<i>System</i>
1. Klik menu “Laporan Masyarakat”	2. Menampilkan halaman tabel data laporan masyarakat

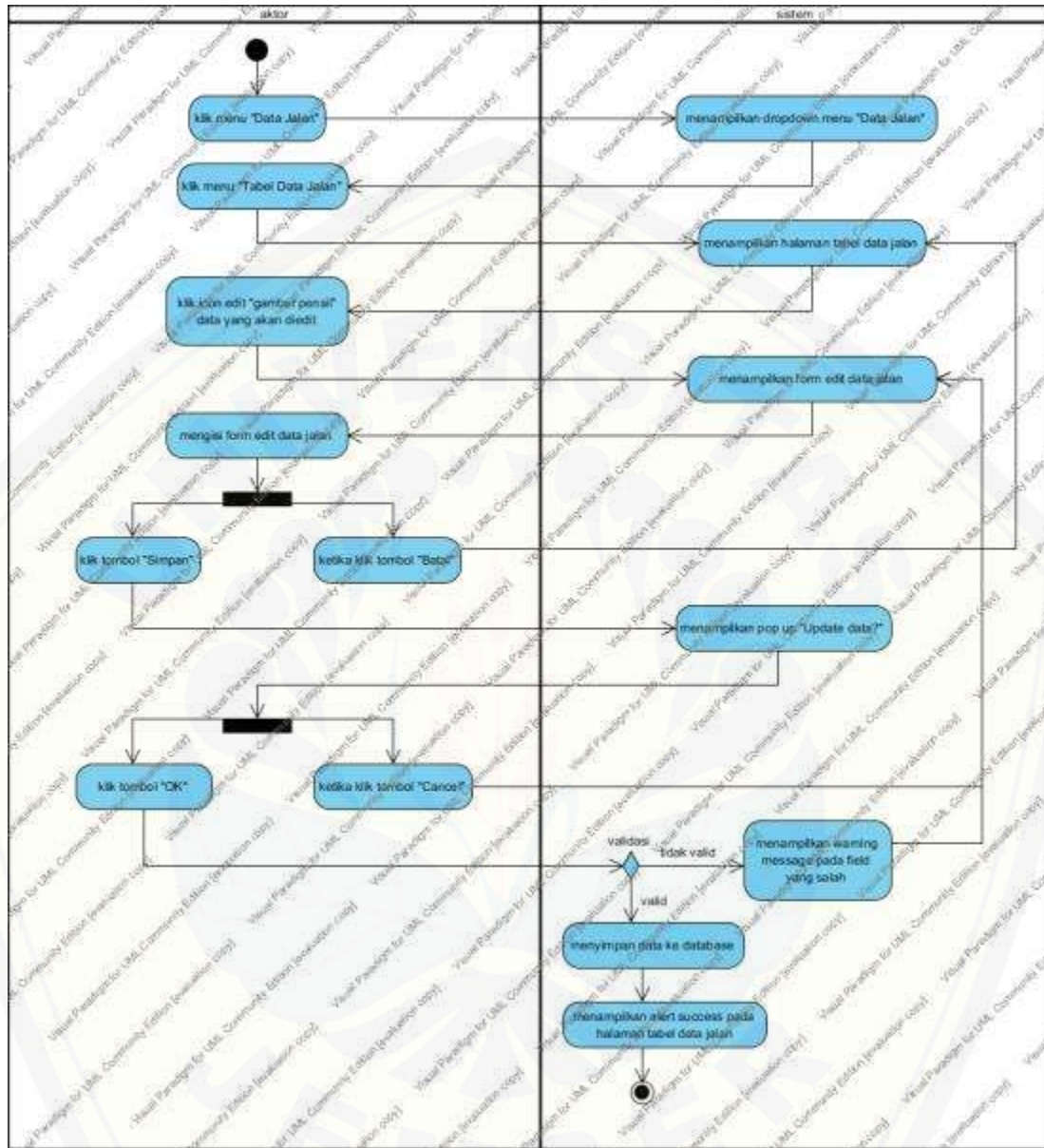
#### SKENARIO ALTERNATIF

## A.2 Perancangan Activity Diagram

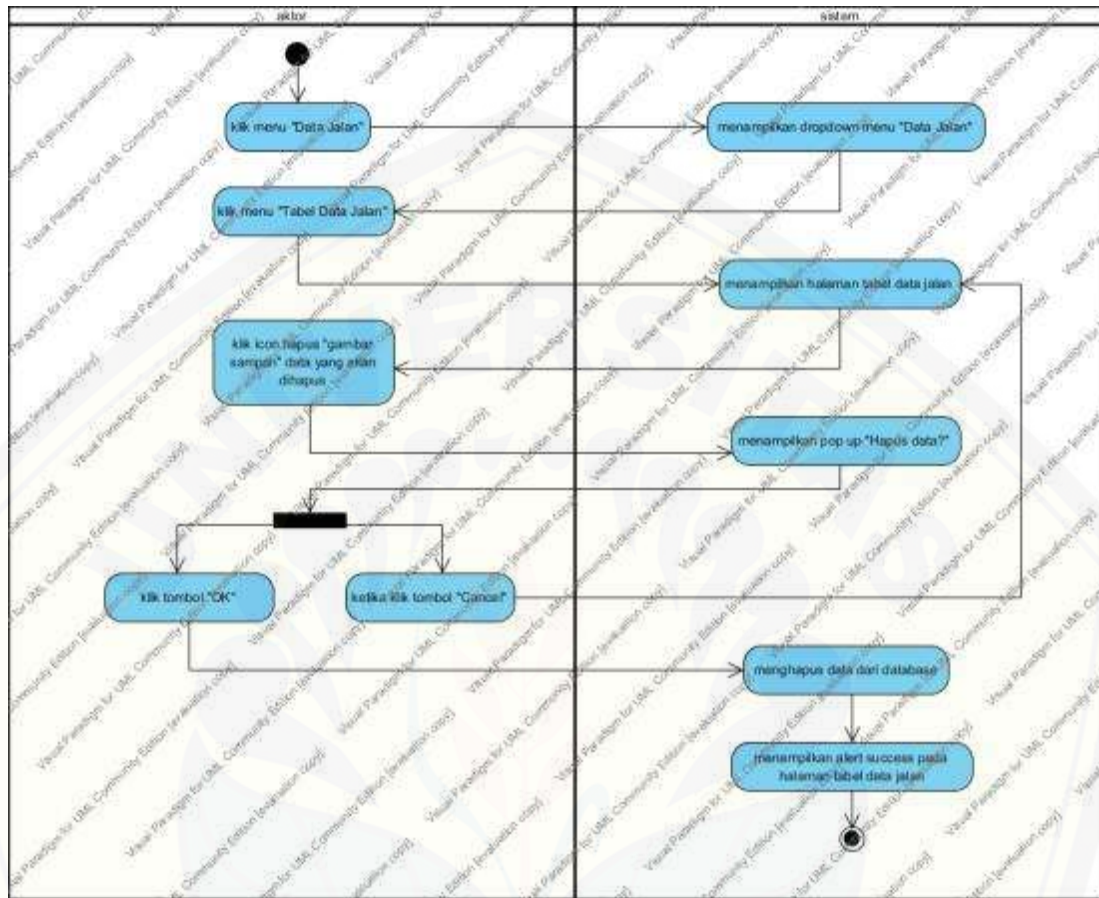
### 1. Activity diagram tambah data jalan



### 2. Activity diagram ubah data jalan

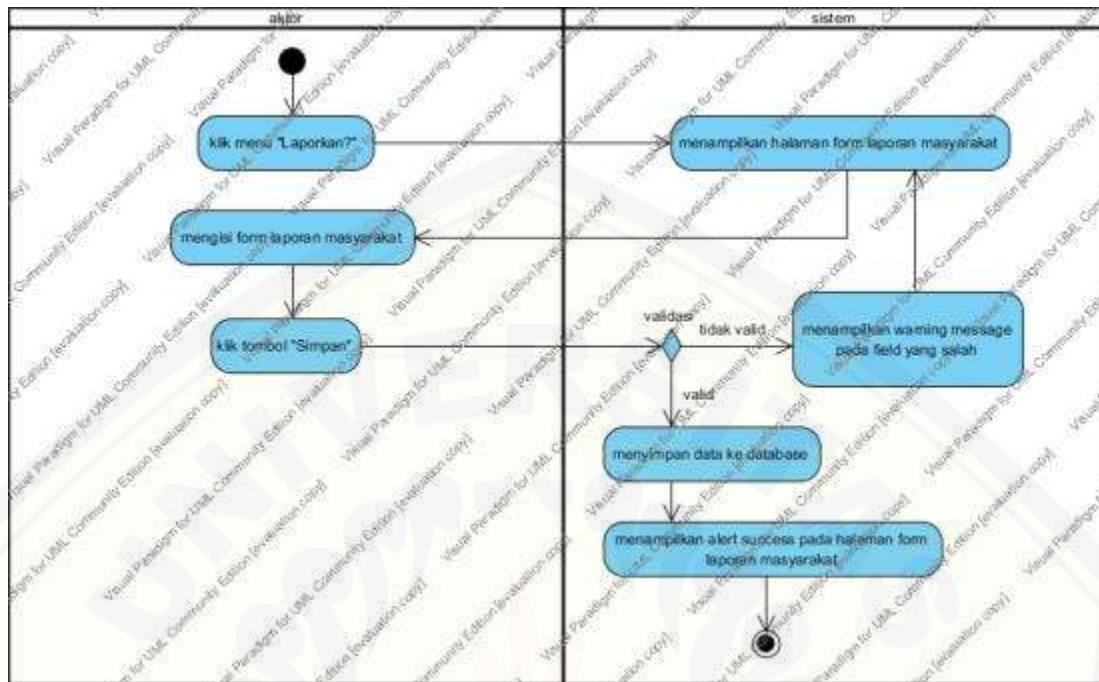


3. Activity diagram hapus data jalan



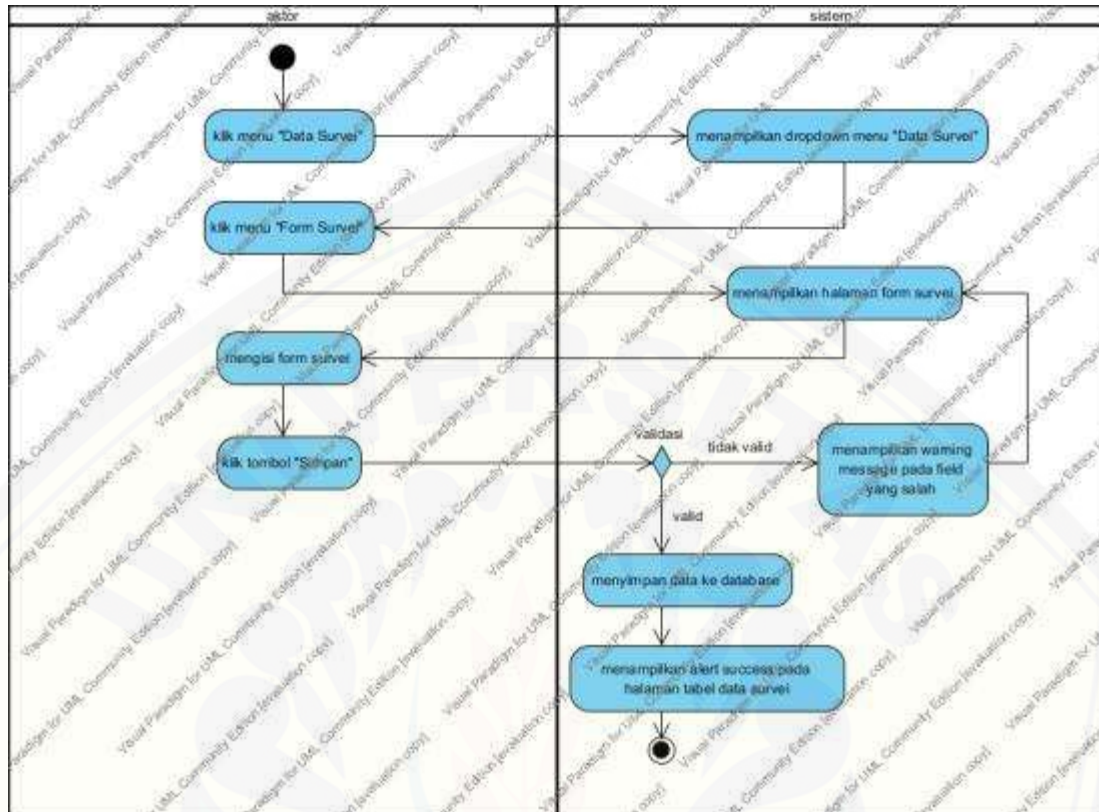
4. Activity diagram tambah laporan masyarakat



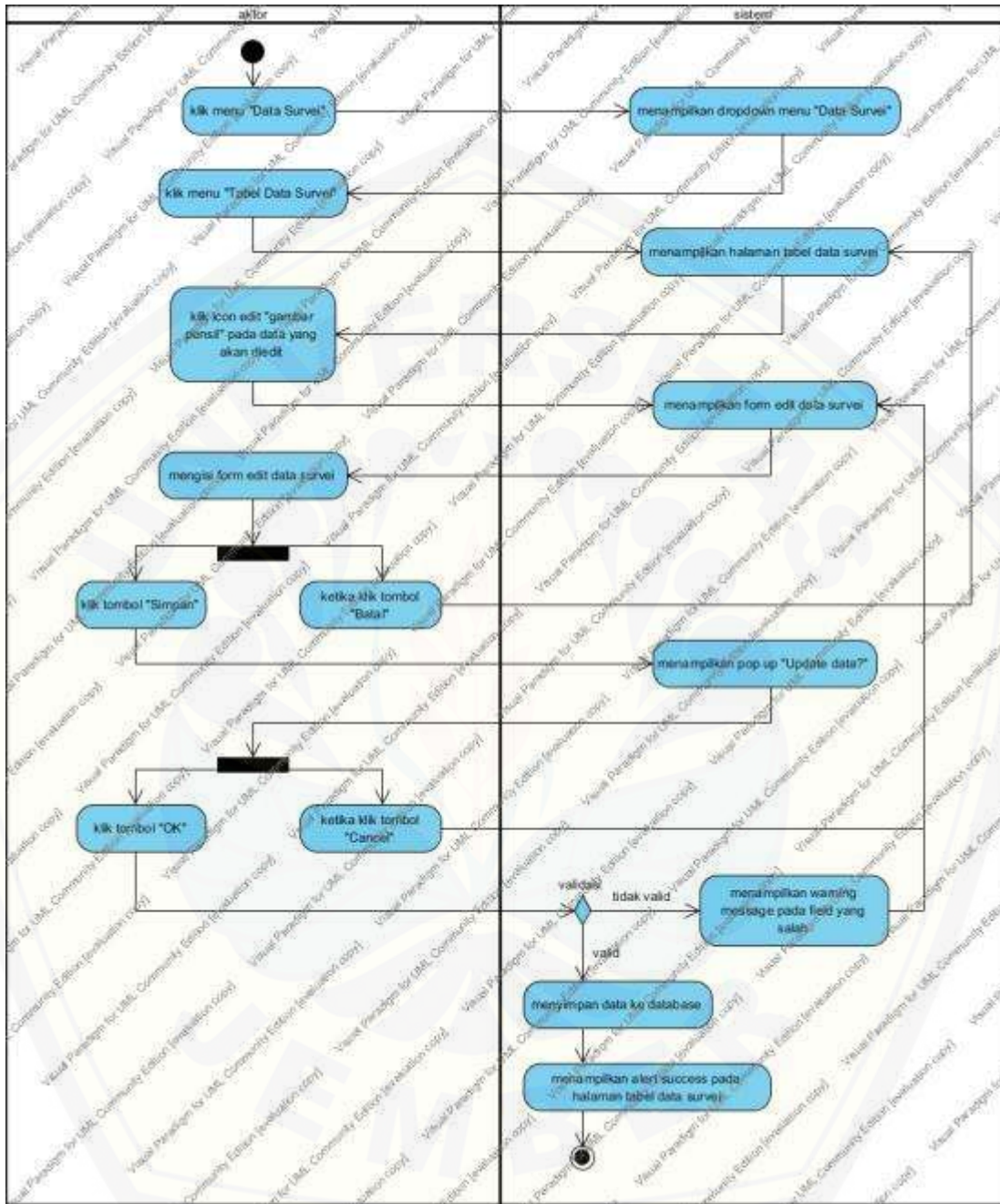


5. *Activity diagram* tambah data survei

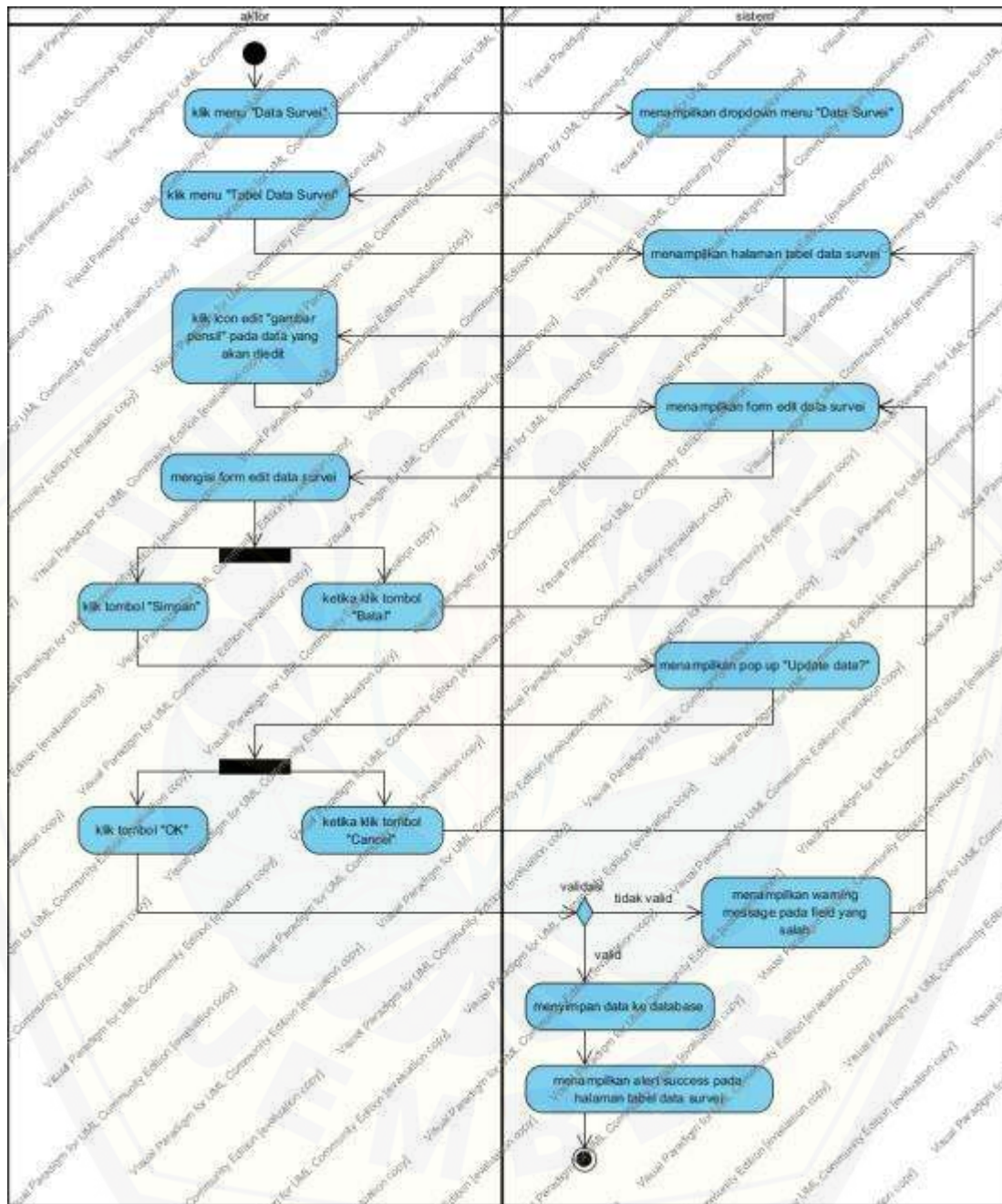




6. Activity diagram ubah data survei

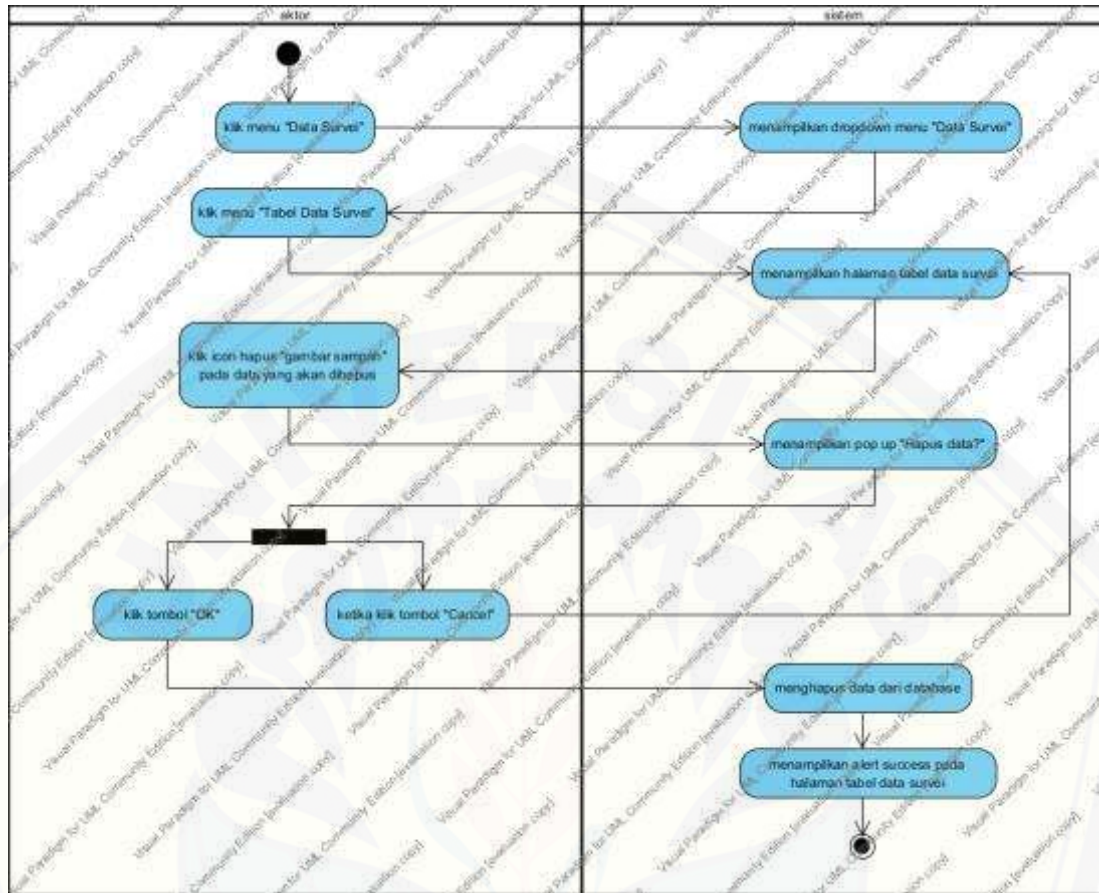


7. Activity diagram ubah status survei

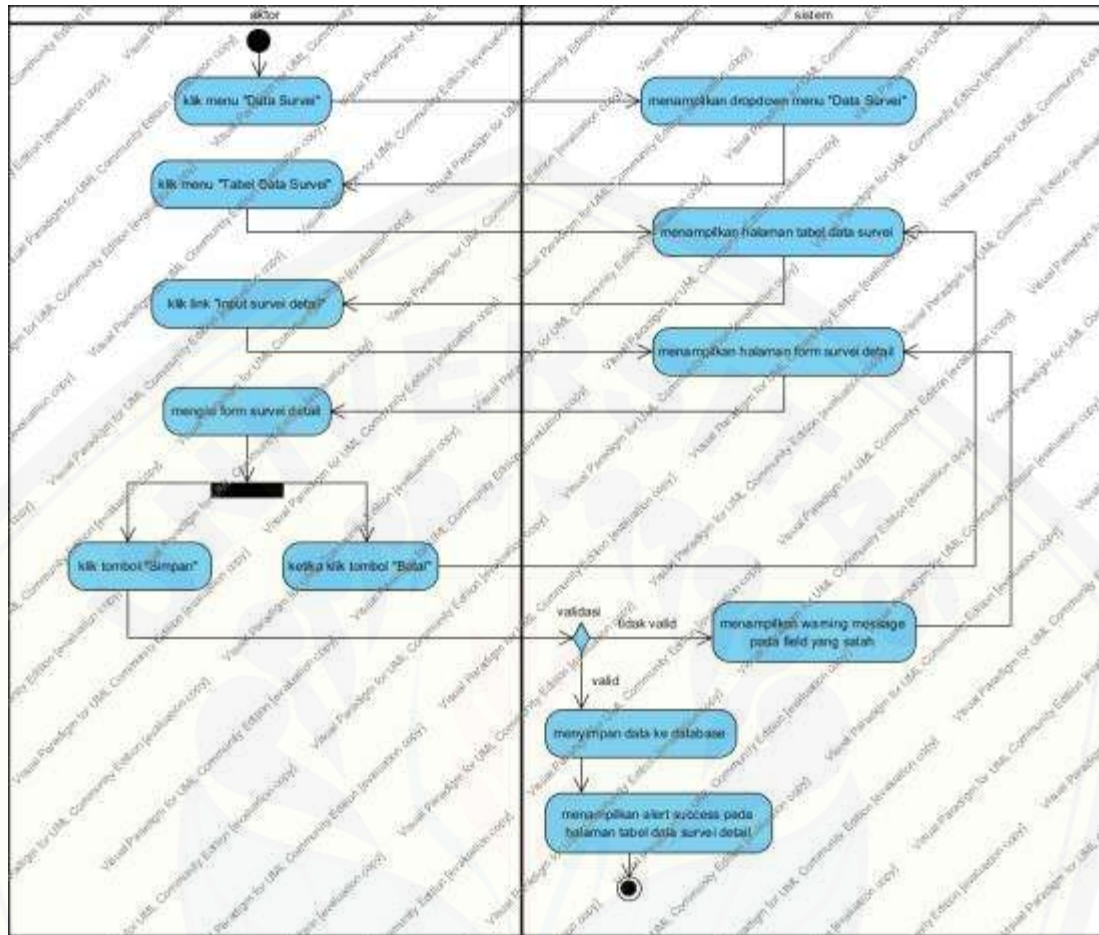


8. Activity diagram hapus data survei



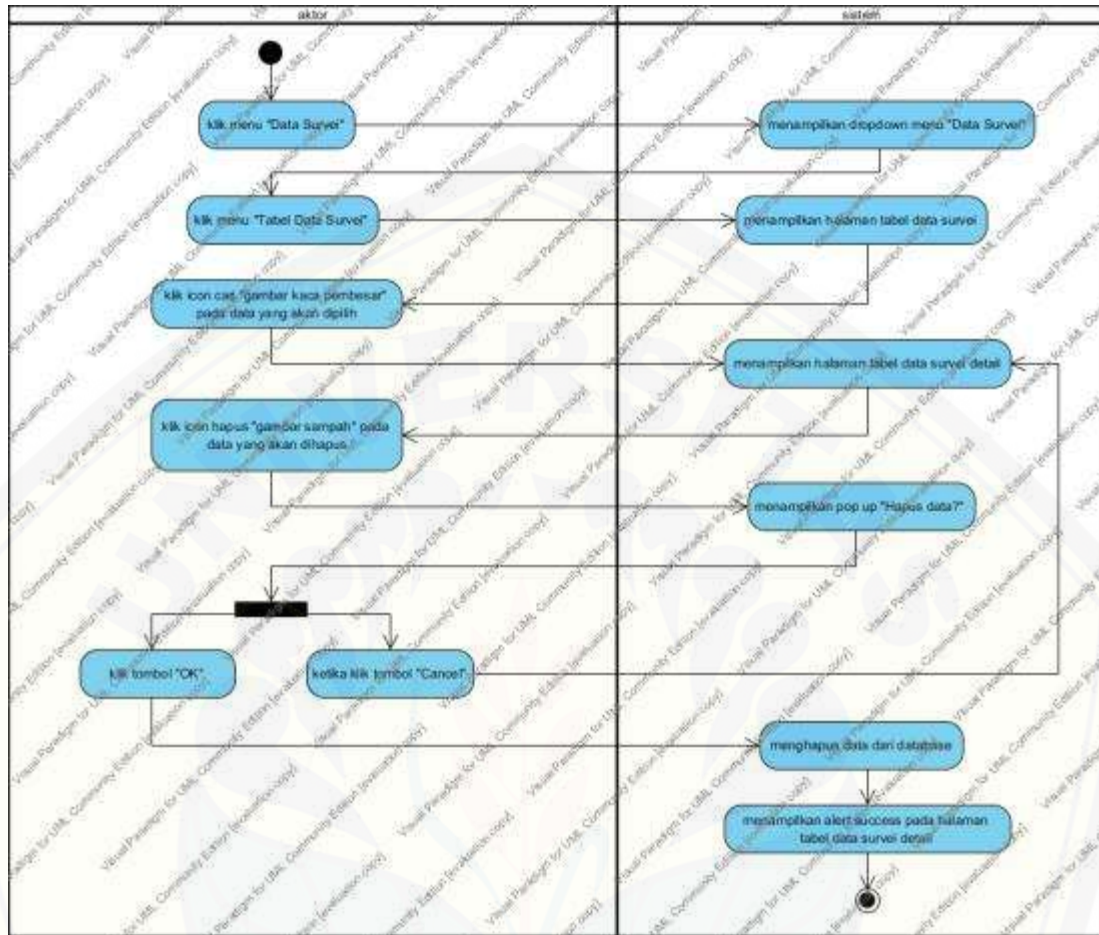


9. Activity diagram tambah data survei detail admin

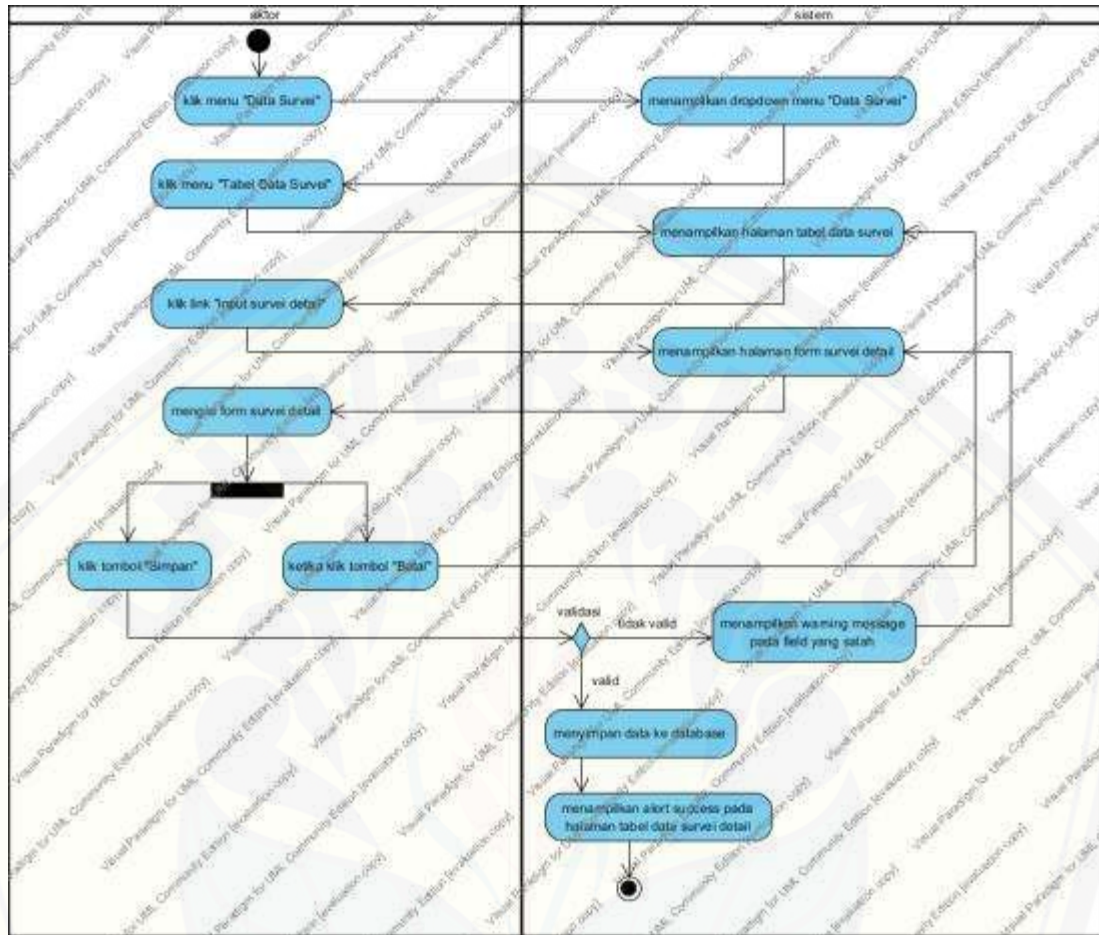


10. Activity diagram hapus data survei detail admin

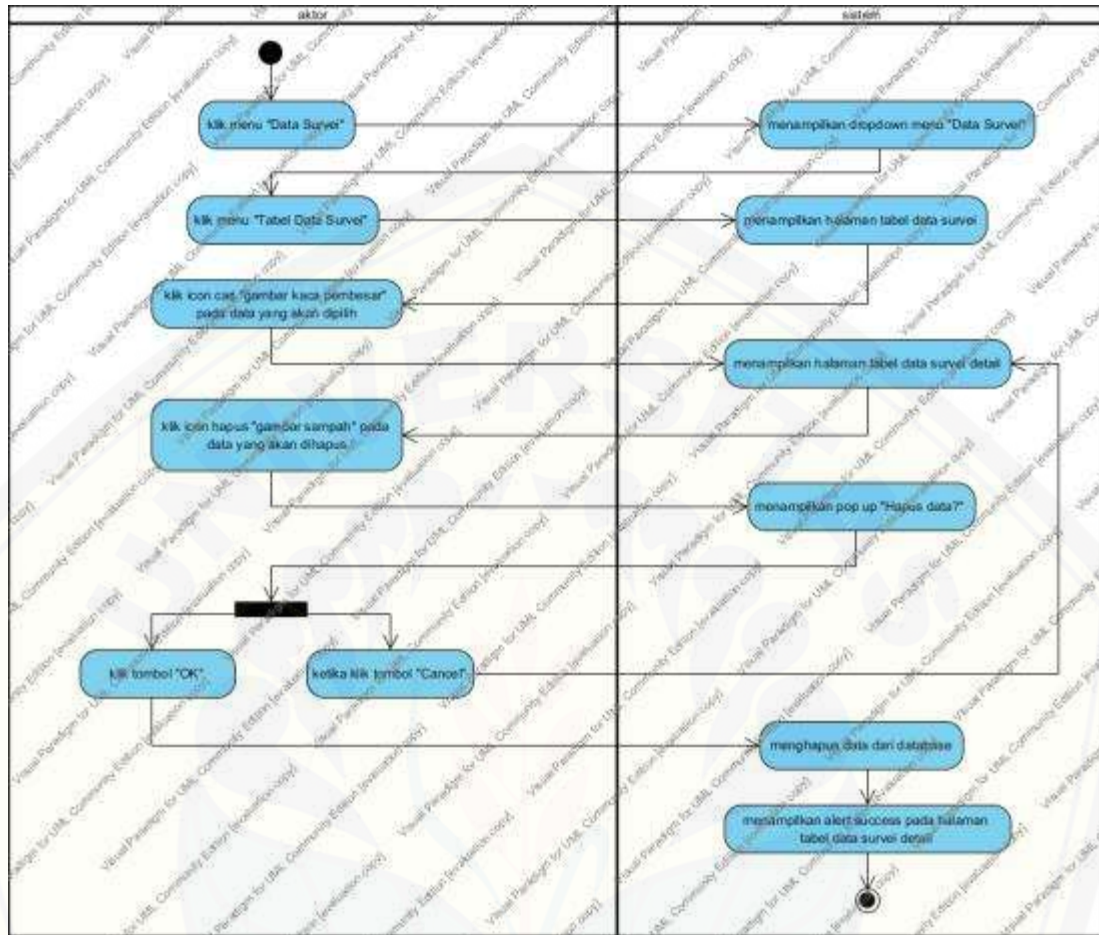




11. Activity diagram tambah data survei detail surveyor

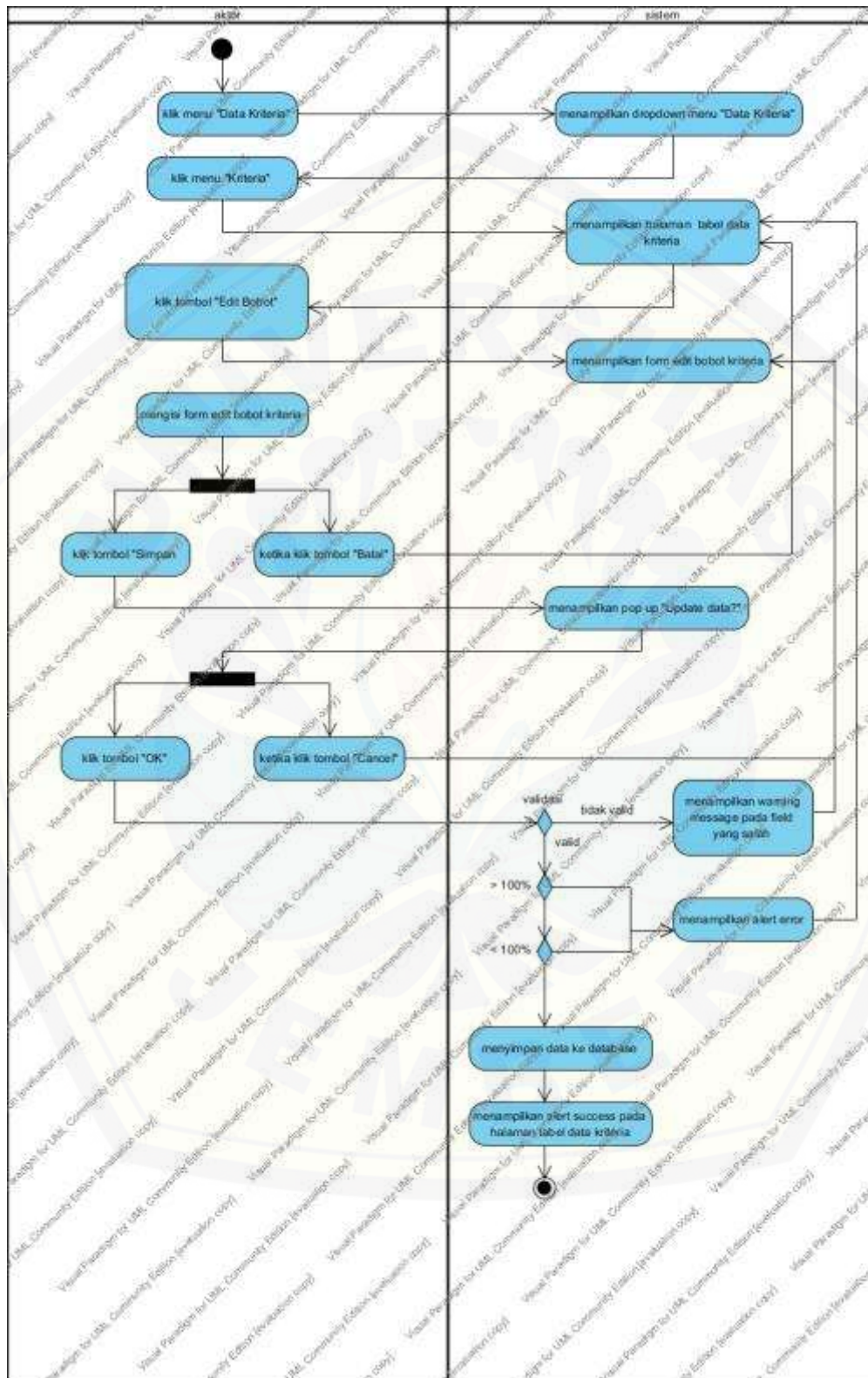


12. Activity diagram hapus data survei detail surveyor



13. Activity diagram ubah bobot kriteria

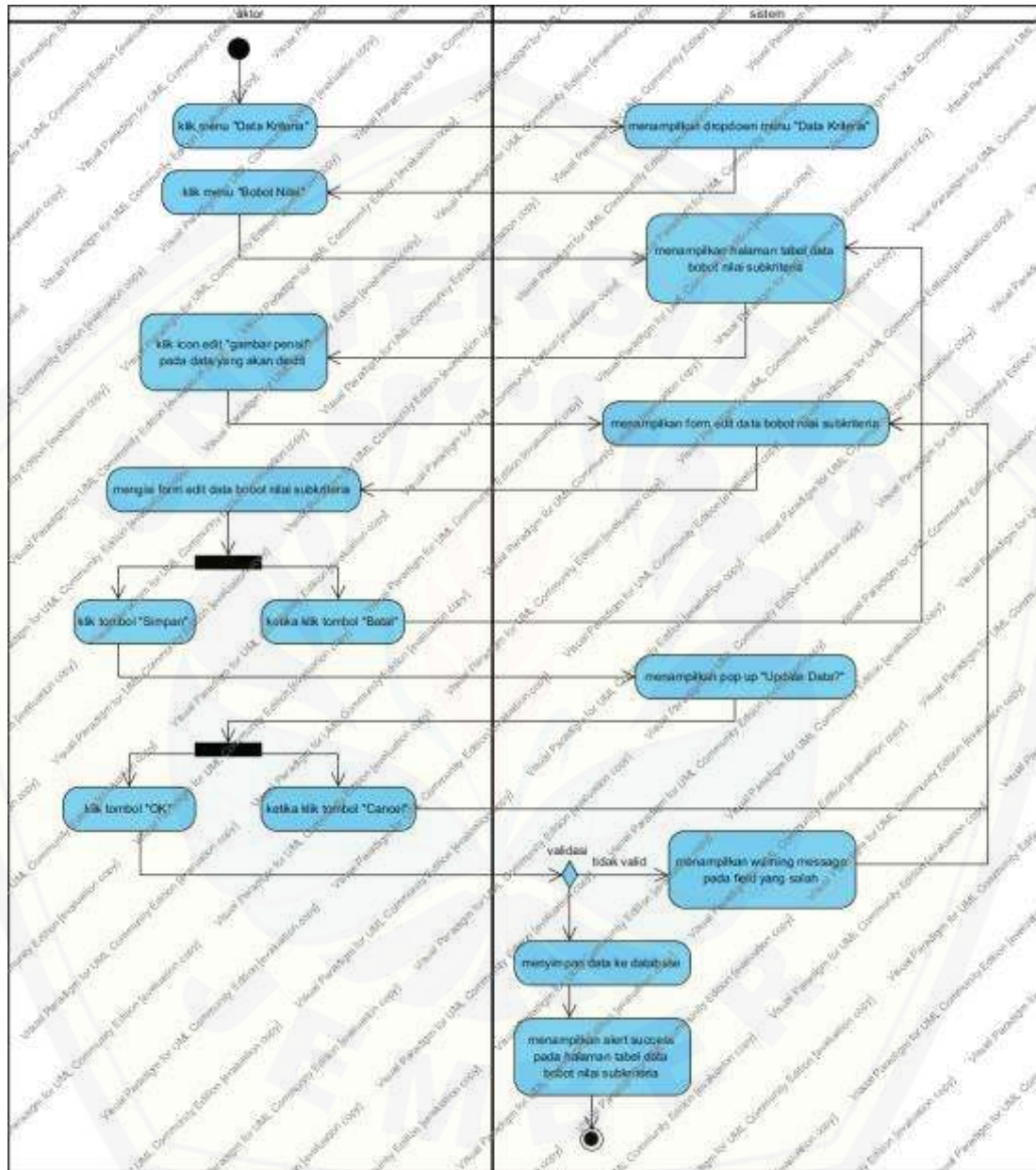








15. Activity diagram ubah bobot nilai subkriteria

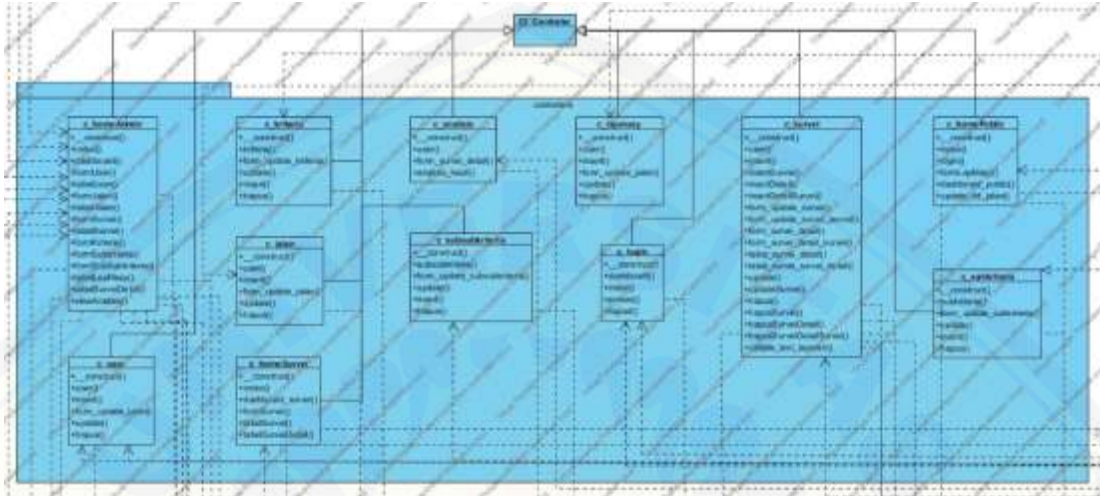


16. Activity diagram lihat data laporan masyarakat

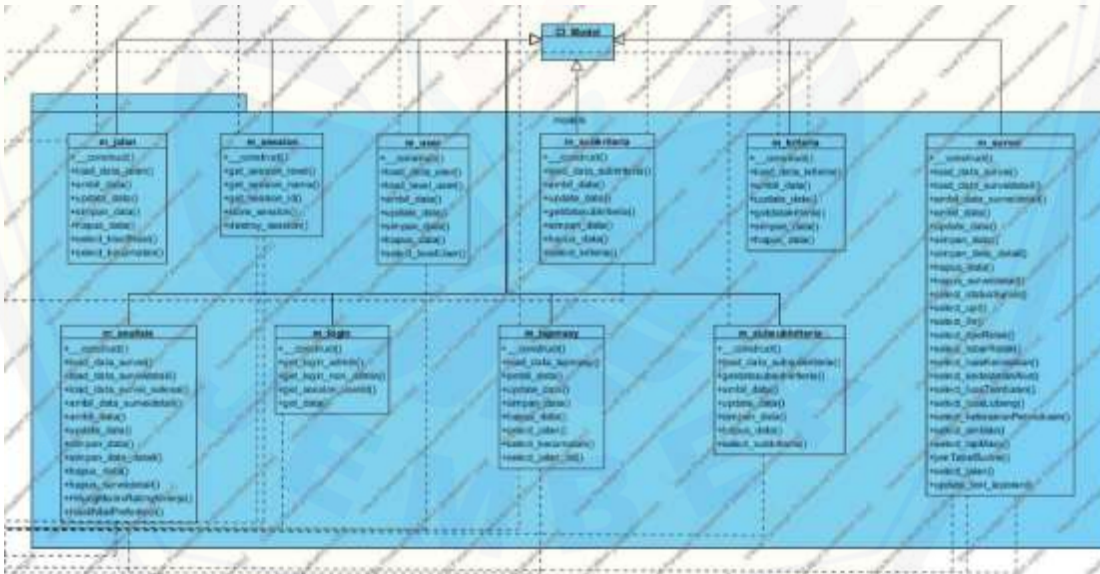


### A.4 Perancangan Class Diagram

#### 1. Controller



#### 2. Model



#### 3. View





2	Ubah data survei	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan</li> <li>• Ketika klik tombol OK pada <i>pop up</i> dan semua <i>field</i> sudah terisi dengan benar</li> <li>• Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> <li>• Ketika klik tombol batal</li> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>pop up</i> “Update data?”</li> <li>• Menyimpan data dan menampilkan alert success pada halaman tabel data survei <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal</li> <li>• Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> <li>• Menampilkan halaman tabel data survei</li> <li>• Menampilkan halaman form edit data survei</li> </ul>
3	Hapus data user	<ul style="list-style-type: none"> <li>• Klik <i>icon</i> hapus “gambar sampah” pada data yang akan dihapus</li> <li>• Ketika klik tombol OK pada <i>pop up</i></li> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>pop up</i> “Hapus data?” <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal</li> <li>• Menghapus data dan menampilkan alert success pada halaman tabel data survei</li> <li>• Menampilkan halaman tabel data survei</li> </ul>

2. *Black box testing* fitur mengelola data survei detail admin

No	Fitur	Kasus	Hasil	Keterangan
----	-------	-------	-------	------------



1. Tambah data survei detail	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan dan semua <i>field</i> sudah terisi dengan benar</li> </ul>	<ul style="list-style-type: none"> <li>• Menyimpan data dan menampilkan alert success pada halaman tabel data survei detail [ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal</li> </ul>
	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> </ul>
	<ul style="list-style-type: none"> <li>• Ketika klik tombol batal</li> </ul>	<ul style="list-style-type: none"> <li>• menampilkan halaman tabel data survei</li> </ul>
2. Hapus data survei detail	<ul style="list-style-type: none"> <li>• Klik <i>icon</i> hapus “gambar sampah” pada data yang akan dihapus</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>pop up</i> “Hapus data?” [ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal</li> </ul>
	<ul style="list-style-type: none"> <li>• Ketika klik tombol OK pada <i>pop up</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menghapus data dan menampilkan alert success pada halaman tabel data survei detail</li> </ul>
	<ul style="list-style-type: none"> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan halaman tabel data survei detail</li> </ul>

### 3. *Black box testing* fitur mengelola data survei detail surveyor

No	Fitur	Kasus	Hasil	Keterangan
1.	Tambah data survei detail	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan dan semua <i>field</i> sudah terisi dengan benar</li> </ul>	<ul style="list-style-type: none"> <li>• Menyimpan data dan menampilkan alert success pada halaman tabel data survei detail</li> </ul>	[ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal

	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> <li>• Ketika klik tombol batal</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> <li>• menampilkan halaman tabel data survei</li> </ul>
2. Hapus data survei detail	<ul style="list-style-type: none"> <li>• Klik <i>icon</i> hapus “gambar sampah” pada data yang akan dihapus</li> <li>• Ketika klik tombol OK pada <i>pop up</i></li> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>pop up</i> “Hapus data?” [ <input checked="" type="checkbox"/> ] Berhasil [ <input type="checkbox"/> ] Gagal</li> <li>• Menghapus data dan menampilkan alert success pada halaman tabel data survei detail</li> <li>• Menampilkan halaman tabel data survei detail</li> </ul>

4. *Black box testing* fitur mengubah bobot kriteria

No	Fitur	Kasus	Hasil	Keterangan
1.	Ubah bobot kriteria	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan</li> <li>• Ketika klik tombol OK pada <i>pop up</i> dan semua <i>field</i> sudah terisi dengan benar</li> <li>• Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>pop up</i> “Update data?”</li> <li>• Menyimpan data dan menampilkan alert success pada halaman tabel data kriteria</li> <li>• Menampilkan <i>warning message</i> pada <i>field</i> yang</li> </ul>	<p>[ <input checked="" type="checkbox"/> ] Berhasil</p> <p>[ <input type="checkbox"/> ] Gagal</p>

---

<p>salah</p> <ul style="list-style-type: none"> <li>• Ketika klik tombol batal</li> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<p>salah</p> <ul style="list-style-type: none"> <li>• Menampilkan kembali halaman tabel data kriteria</li> <li>• Menampilkan kembali halaman form edit data kriteria</li> </ul>
---	---

---

5. *Black box testing* fitur mengubah bobot subkriteria

No	Fitur	Kasus	Hasil	Keterangan
1.	Ubah bobot subkriteria	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan</li> <li>• Ketika klik tombol OK pada <i>pop up</i> dan semua <i>field</i> sudah terisi dengan benar</li> <li>• Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> <li>• Ketika klik tombol batal</li> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>pop up</i> "Update data?"</li> <li>• Menyimpan data dan menampilkan alert success pada halaman tabel data subkriteria</li> <li>• Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> <li>• Menampilkan kembali halaman tabel data subkriteria</li> <li>• Menampilkan kembali halaman form edit data subkriteria</li> </ul>	<p>[ √ ] Berhasil</p> <p>[ ] Gagal</p>

---

6. *Black box testing* fitur mengubah bobot nilai subkriteria

No	Fitur	Kasus	Hasil	Keterangan
1.	Ubah bobot nilai subkriteria	<ul style="list-style-type: none"> <li>• Ketika klik tombol simpan</li> <li>• Ketika klik tombol OK pada <i>pop up</i> dan semua <i>field</i> sudah terisi dengan benar</li> <li>• Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> <li>• Ketika klik tombol batal</li> <li>• Ketika klik tombol <i>cancel</i> ditampilkan <i>pop up</i></li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>pop up</i> “Update data?”</li> <li>• Menyimpan data dan menampilkan alert success pada halaman tabel data bobot nilai subkriteria</li> <li>• Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> <li>• Menampilkan halaman tabel data bobot nilai subkriteria</li> <li>• Menampilkan halaman form edit data bobot nilai subkriteria</li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>

7. *Black box testing* fitur melihat laporan masyarakat

No	Fitur	Kasus	Hasil	Keterangan
1.	Lihat data laporan masyarakat	<ul style="list-style-type: none"> <li>• Ketika klik menu laporan masyarakat</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan halaman tabel data laporan masyarakat</li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>

8. *Black box testing* fitur analisis prioritas jalan

No	Fitur	Kasus	Hasil	Keterangan
1.	Analisis prioritas jalan	<ul style="list-style-type: none"> <li>Ketika klik tombol analisis pada data survei yang akan dianalisis</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan tabel data alternatif, matrix kinerja (ternormalisasi), tabel hasil nilai preferensi</li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>

9. *Black box testing* fitur menambah laporan masyarakat

No	Fitur	Kasus	Hasil	Keterangan
1.	Tambah laporan masyarakat	<ul style="list-style-type: none"> <li>Ketika klik tombol simpan dan semua <i>field</i> sudah terisi dengan benar</li> <li>Ketika klik tombol simpan dan inputan/<i>field</i> ada yang kosong atau salah</li> </ul>	<ul style="list-style-type: none"> <li>Menyimpan data dan menampilkan alert success pada halaman form laporan</li> <li>Menampilkan <i>warning message</i> pada <i>field</i> yang salah</li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ <input type="checkbox"/> ] Gagal</li> </ul>



## LAMPIRAN C. KODE PROGRAM

### 1. Kode Program *Controller* analisis (c\_analisis)

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class c_analisis extends CI_Controller {

    public function __construct() {

        parent::__construct();
    }
}
```

```
$this->load->model('m_user');

$this->load->model('m_login');

$this->load->model('m_analisis');

$this->load->model('m_session');

}

public function user(){

    $session = $this->m_login->get_session_userid();

    $sid = $session['session_userid'];

}

public function form_survei_detail($id){

    $session = $this->m_session->get_session_id();

    $sid = $session['session_id'];

    $data['survei_detail']=$this->m_analisis->ambil_data($id);

    $this->load->view('form_survei_admin_detail',$data);

}

public function analisis_hasil($id){

    $data['survei_detail']=$this->m_analisis->ambil_data_surveidetail($id);

    $data['matrix_kinerja']=$this->m_analisis->HitungMatrixRatingKinerja($id);

    $data['hasil']=$this->m_analisis->HasilNilaiPreferensi($id);

    $this->load->view('analisis_hasil',$data);

}

}

?>
```

## 2. Kode Program *Model* analisis (m\_analisis)

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

Class m_analisis extends CI_Model {

function __construct() {
    parent::__construct();
}

function load_data_survei(){

    $result = $this->db->query("select * from tb_survei a, tb_status_survei b, tb_upt c

        where a.id_status = b.id_status AND a.id_upt = c.id_upt

        order by id_survei ASC");

    return $result;

}

function load_data_surveidetail(){

    $result = $this->db->query("select * from tb_survei_detail a, tb_jalan b

        where a.id_jalan = b.id_jalan

        order by id_survei ASC");

    return $result;

}

function load_data_survei_selesai(){

    $result = $this->db->query("SELECT *

        FROM tb_survei a
```

```
JOIN tb_status_survei b ON a.id_status = b.id_status

JOIN tb_upt c ON a.id_upt = c.id_upt

WHERE a.id_status =2

ORDER BY id_survei ASC");

return $result;
}

function ambil_data_surveidetail($id){
$result = $this->db->query("select id_survei, id_survei_detail, nama_jalan, c.nilai as s1,
    d.nilai as s2,
    e.nilai as s3,
    f.nilai as s4,
    g.nilai as s5,
    h.nilai as s6,
    i.nilai as s7,
    j.nilai as s8,
    k.nilai as s9,
    l.nilai as s10 from tb_survei_detail a
    JOIN tb_jalan b ON a.id_jalan = b.id_jalan
    JOIN tb_subsubkriteria c ON a.c1 = c.id_subsubkriteria
    JOIN tb_subsubkriteria d ON a.c2 = d.id_subsubkriteria
    JOIN tb_subsubkriteria e ON a.c3 = e.id_subsubkriteria
    JOIN tb_subsubkriteria f ON a.c4 = f.id_subsubkriteria
    JOIN tb_subsubkriteria g ON a.c5 = g.id_subsubkriteria
```

```
JOIN tb_subsubkriteria h ON a.c6 = h.id_subsubkriteria

JOIN tb_subsubkriteria i ON a.c7 = i.id_subsubkriteria

JOIN tb_subsubkriteria j ON a.c8 = j.id_subsubkriteria

JOIN tb_subsubkriteria k ON a.c9 = k.id_subsubkriteria

JOIN tb_subsubkriteria l ON a.c10 = l.id_subsubkriteria

WHERE id_survei='$id');

return $result;

}

function ambil_data($id){

    return $this->db->query("select * from tb_survei where id_survei='$id'");

}

function update_data($input) {

    $query = $this->db->query("UPDATE tb_survei SET `tahun`='$input[tahun]',

    `kegiatan`='$input[kegiatan]',

    `id_status`='$input[status]',

    `id_upt`='$input[upt]'

    WHERE `id_survei`='$input[id_survei]'");

    return $query;

}

function simpan_data($data){

    return $this->db->insert('tb_survei',$data);

}
```



```
function simpan_data_detail($data){

    return $this->db->insert('tb_survei_detail',$data);

}

public function hapus_data($id){

    $query = $this->db->query("DELETE from tb_survei where id_survei='$id'");

    $query = $this->db->query("DELETE from tb_survei_detail where id_survei='$id'");

    return $query;

}

public function hapus_surveidetil($id){

    $query = $this->db->query("DELETE from tb_survei_detail where id_survei_detail='$id'");

    return $query;

}

function HitungMatrixRatingKinerja($id){

    $query1 = $this->db->query("CREATE VIEW nilai AS

    select id_survei, id_survei_detail, b.id_jalan AS id_jalan, c.nilai as s1,

    d.nilai as s2,

    e.nilai as s3,

    f.nilai as s4,

    g.nilai as s5,

    h.nilai as s6,

    i.nilai as s7,

    j.nilai as s8,

    k.nilai as s9,
```

```
l.nilai as s10 from tb_survei_detail a

JOIN tb_jalan b ON a.id_jalan = b.id_jalan

JOIN tb_subsubkriteria c ON a.c1 = c.id_subsubkriteria

JOIN tb_subsubkriteria d ON a.c2 = d.id_subsubkriteria

JOIN tb_subsubkriteria e ON a.c3 = e.id_subsubkriteria

JOIN tb_subsubkriteria f ON a.c4 = f.id_subsubkriteria

JOIN tb_subsubkriteria g ON a.c5 = g.id_subsubkriteria

JOIN tb_subsubkriteria h ON a.c6 = h.id_subsubkriteria

JOIN tb_subsubkriteria i ON a.c7 = i.id_subsubkriteria

JOIN tb_subsubkriteria j ON a.c8 = j.id_subsubkriteria

JOIN tb_subsubkriteria k ON a.c9 = k.id_subsubkriteria

JOIN tb_subsubkriteria l ON a.c10 = l.id_subsubkriteria

WHERE id_survei='$id');

$query2 = $this->db->query("SELECT MAX(s1) as m1, MAX(s2) as m2, MAX(s3) as m3,
MAX(s4) as m4, MAX(s5) as m5, MAX(s6) as m6, MAX(s7) as m7, MAX(s8) as m8,
MAX(s9) as m9, MAX(s10) as m10 from nilai");

foreach ($query2->result_array() as $max) {

    $nilaimax1 = $max['m1'];

    $nilaimax2 = $max['m2'];

    $nilaimax3 = $max['m3'];

    $nilaimax4 = $max['m4'];

    $nilaimax5 = $max['m5'];

    $nilaimax6 = $max['m6'];
```

```
$nilaimax7 = $max['m7'];

$nilaimax8 = $max['m8'];

$nilaimax9 = $max['m9'];

$nilaimax10 = $max['m10'];

}

$query3 = $this->db->query("CREATE VIEW ratingkinerja AS

SELECT id_survei, id_survei_detail, id_jalan,

s1/$nilaimax1 as s1,

s2/$nilaimax2 as s2,

s3/$nilaimax3 as s3,

s4/$nilaimax4 as s4,

s5/$nilaimax5 as s5,

s6/$nilaimax6 as s6,

s7/$nilaimax7 as s7,

s8/$nilaimax8 as s8,

s9/$nilaimax9 as s9,

s10/$nilaimax10 as s10 FROM nilai");

$query4 = $this->db->query("SELECT b.nama_jalan, s1,s2,s3,s4,s5,s6,s7,s8,s9,s10 FROM ratingkinerja a

JOIN tb_jalan b ON a.id_jalan = b.id_jalan");

return $query4;

}

function HasilNilaiPreferensi($id){
```

```
$bobot1 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a, tb_subkriteria
b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from tb_subkriteria)");

$bobot2 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a, tb_subkriteria
b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from tb_subkriteria)+1");

$bobot3 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a, tb_subkriteria
b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from tb_subkriteria)+2");

$bobot4 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a, tb_subkriteria
b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from tb_subkriteria)+3");

$bobot5 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a, tb_subkriteria
b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from tb_subkriteria)+4");

$bobot6 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a, tb_subkriteria
b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from tb_subkriteria)+5");

$bobot7 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a, tb_subkriteria
b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from tb_subkriteria)+6");

$bobot8 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a, tb_subkriteria
b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from tb_subkriteria)+7");

$bobot9 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a, tb_subkriteria
b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from tb_subkriteria)+8");

$bobot10 = $this->db->query("SELECT a.bobot as bot1, b.bobot as bot2 FROM `tb_kriteria` a,
tb_subkriteria b where a.id_kriteria = b.id_kriteria and b.id_subkriteria = (select min(id_subkriteria) from
tb_subkriteria)+9");

foreach ($bobot1->result_array() as $k) {

    $bot1x =(double)($k['bot1']/100) * (double)($k['bot2']/100);

}

foreach ($bobot2->result_array() as $k) {

    $bot2x =(double)($k['bot1']/100) * (double)($k['bot2']/100);
```

```
}  
  
foreach ($bobot3->result_array() as $k) {  
  
    $bot3x =(double)($k['bot1']/100) * (double)($k['bot2']/100);  
  
}  
  
foreach ($bobot4->result_array() as $k) {  
  
    $bot4x =(double)($k['bot1']/100) * (double)($k['bot2']/100);  
  
}  
  
foreach ($bobot5->result_array() as $k) {  
  
    $bot5x =(double)($k['bot1']/100) * (double)($k['bot2']/100);  
  
}  
  
foreach ($bobot6->result_array() as $k) {  
  
    $bot6x =(double)($k['bot1']/100) * (double)($k['bot2']/100);  
  
}  
  
foreach ($bobot7->result_array() as $k) {  
  
    $bot7x =(double)($k['bot1']/100) * (double)($k['bot2']/100);  
  
}  
  
foreach ($bobot8->result_array() as $k) {  
  
    $bot8x =(double)($k['bot1']/100) * (double)($k['bot2']/100);  
  
}  
  
foreach ($bobot9->result_array() as $k) {  
  
    $bot9x =(double)($k['bot1']/100) * (double)($k['bot2']/100);  
  
}
```



```
foreach ($bobot10->result_array() as $k) {  
  
    $bot10x =(double)($k['bot1']/100) * (double)($k['bot2']/100);  
  
}  
  
$bobot1 = $bot1x;  
  
$bobot2 = $bot2x;  
  
$bobot3 = $bot3x;  
  
$bobot4 = $bot4x;  
  
$bobot5 = $bot5x;  
  
$bobot6 = $bot6x;  
  
$bobot7 = $bot7x;  
  
$bobot8 = $bot8x;  
  
$bobot9 = $bot9x;  
  
$bobot10 = $bot10x;  
  
$query1 = $this->db->query("CREATE VIEW nilaipreferensi AS SELECT id_survei, id_survei_detail,  
id_jalan,  
  
s1 * $bobot1 as s1,  
  
s2 * $bobot2 as s2,  
  
s3 * $bobot3 as s3,  
  
s4 * $bobot4 as s4,  
  
s5 * $bobot5 as s5,  
  
s6 * $bobot6 as s6,  
  
s7 * $bobot7 as s7,  
  
s8 * $bobot8 as s8,
```

```
s9 * $bobot9 as s9,  
  
s10 * $bobot10 as s10 FROM ratingkinerja a  
  
");  
  
$query2 = $this->db->query("SELECT b.nama_jalan, s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,  
(s1+s2+s3+s4+s5+s6+s7+s8+s9+s10) as tot  
  
FROM nilaipreferensi a JOIN tb_jalan b ON a.id_jalan = b.id_jalan  
  
WHERE id_survei='$id' order by tot desc");  
  
$query3 = $this->db->query("DROP VIEW nilai");  
  
$query4 = $this->db->query("DROP VIEW ratingkinerja");  
  
$query5 = $this->db->query("DROP VIEW nilaipreferensi");  
  
return $query2;  
  
}  
  
}  
  
?>
```

### 3. Kode Program *Controller* laporan masyarakat (c\_lapMasy)

```
<?php  
  
if (!defined('BASEPATH'))  
  
    exit('No direct script access allowed');  
  
class c_lapmasy extends CI_Controller {  
  
    public function __construct() {  
  
        parent::__construct();  
  
        $this->load->model('m_user');
```

```
$this->load->model('m_login');

$this->load->model('m_lapmasy');

$this->load->model('m_session');

}

public function user(){

    $session = $this->m_login->get_session_userid();

    $id = $session['session_userid'];

}

public function insert(){ // yg depan variabel array, yg belakang nama variabel dr form

    $input ['tanggal']    = $this->input->post('tgl');

    $input ['nama']       = $this->input->post('nama');

    $input ['id_jalan']   = $this->input->post('id_jalan');

    $input ['laporan']    = $this->input->post('laporan');

    // yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

    $data = array(

        'tanggal'    => $input['tanggal'],

        'nama'       => $input['nama'],

        'id_jalan'   => $input['id_jalan'],

        'laporan'    => $input['laporan']

    );

    $this->m_lapmasy->simpan_data($data);

    redirect('c_homePublic/dashboard_public');

}
```

```
}  
?>
```

#### 4. Kode Program *Model* laporan masyarakat (m\_lapmasy)

```
<?php  
  
if (!defined('BASEPATH'))  
    exit('No direct script access allowed');  
  
Class m_lapmasy extends CI_Model {  
  
    function __construct() {  
        parent::__construct();  
    }  
  
    function load_data_lapmasy(){  
        $result = $this->db->query("select * from tb_lapmasy a, tb_jalan b  
            where a.id_jalan = b.id_jalan order by id_lapmasy ASC");  
        return $result;  
    }  
  
    function simpan_data($data){  
        return $this->db->insert('tb_lapmasy',$data);  
    }  
  
    public function hapus_data($id){
```

```
$query = $this->db->query("DELETE from tb_lapmasy where id_lapmasy='$id'");

return $query;

}

public function select_jalan(){

    $query = $this->db->query("select * from tb_jalan");

    return $query;

}

public function select_kecamatan(){

    $query = $this->db->query("select * from tb_kecamatan");

    return $query;

}

public function select_jalan_dd($id_kec){

    $query = $this->db->query("select * from tb_jalan WHERE id_kecamatan='$id_kec'");

    return $query;

}

}

?>
```

##### 5. Kode Program *Controller* survei (c\_survei)

```
<?php

if (!defined('BASEPATH'))

    exit('No direct script access allowed');
```



```
class c_survei extends CI_Controller {

    public function __construct() {

        parent::__construct();

        $this->load->model('m_user');

        $this->load->model('m_login');

        $this->load->model('m_survei');

        $this->load->model('m_session');

    }

    public function user(){

        $session = $this->m_login->get_session_userid();

        $id = $session['session_userid'];

    }

    public function insert(){ // yg depan variabel array, yg belakang nama variabel dr form

        $input ['tanggal']      = $this->input->post('tgl');

        $input ['kegiatan']     = $this->input->post('kegiatan');

        $input ['status']       = $this->input->post('status');

        $input ['upt']          = $this->input->post('upt');
```

```
// yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

$data = array(

    'tanggal' => $input['tanggal'],

    'kegiatan' => $input['kegiatan'],

    'id_status' => $input['status'],

    'id_upt' => $input['upt']

);

$this->m_survei->simpan_data($data);

redirect('c_homeAdmin/tabelSurvei');

}

public function insertSurvei(){ // yg depan variabel array, yg belakang nama variabel dr form

    $input ['tahun']          = $this->input->post('tahun');

    $input ['kegiatan']       = $this->input->post('kegiatan');

    $input ['status']         = $this->input->post('status');

    $input ['upt']            = $this->input->post('upt');

    // yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

    $data = array(
```

```
'tahun' => $input['tahun'],

'kegiatan' => $input['kegiatan'],

'id_status' => $input['status'],

'id_upt' => $input['upt']

);

$this->m_survei->simpan_data($data);

redirect('c_homeSurvei/tabelSurvei');

}

public function insertDetail() { // yg depan variabel array, yg belakang nama variabel dr form

    $input ['id_survei'] = $this->input->post('id_survei');

    $input ['id_user'] = $this->input->post('id_user');

    $input ['id_jalan'] = $this->input->post('nama_jalan');

    $input ['c1'] = $this->input->post('lhr');

    $input ['c2'] = $this->input->post('tipe_retak');

    $input ['c3'] = $this->input->post('lebar_retak');

    $input ['c4'] = $this->input->post('luas_kerusakan');

    $input ['c5'] = $this->input->post('kedalaman_alur');

    $input ['c6'] = $this->input->post('luas_tambalan');
```

```
$input ['c7']      = $this->input->post('luas_lubang');

$input ['c8']      = $this->input->post('kekasaran_permukaan');

$input ['c9']      = $this->input->post('ambblas');

$input ['c10']     = $this->input->post('lapmasy');

// yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

$data = array(

    'id_survei'     => $input['id_survei'],

    'id_user'      => $input['id_user'],

    'id_jalan'     => $input['id_jalan'],

    'c1'           => $input['c1'],

    'c2'           => $input['c2'],

    'c3'           => $input['c3'],

    'c4'           => $input['c4'],

    'c5'           => $input['c5'],

    'c6'           => $input['c6'],

    'c7'           => $input['c7'],

    'c8'           => $input['c8'],

    'c9'           => $input['c9'],

    'c10'          => $input['c10']

);
```

```
$this->m_survei->simpan_data_detail($data);

redirect('c_homeAdmin/tabelSurveiDetail');

}

public function insertDetailSurvei() { // yg depan variabel array, yg belakang nama variabel dr form

    $input ['id_survei']      = $this->input->post('id_survei');
    $input ['id_user']       = $this->input->post('id_user');
    $input ['id_jalan']      = $this->input->post('nama_jalan');
    $input ['c1']            = $this->input->post('lhr');
    $input ['c2']            = $this->input->post('tipe_retak');
    $input ['c3']            = $this->input->post('lebar_retak');
    $input ['c4']            = $this->input->post('luas_kerusakan');
    $input ['c5']            = $this->input->post('kedalaman_alur');
    $input ['c6']            = $this->input->post('luas_tambalan');
    $input ['c7']            = $this->input->post('luas_lubang');
    $input ['c8']            = $this->input->post('kekasaran_permukaan');
    $input ['c9']            = $this->input->post('ambblas');
    $input ['c10']           = $this->input->post('lapmasy');

    // yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

    $data = array(

        'id_survei'  => $input['id_survei'],

        'id_user'   => $input['id_user'],
```



```
'id_jalan'    => $input['id_jalan'],

'c1'         => $input['c1'],

'c2'         => $input['c2'],

'c3'         => $input['c3'],

'c4'         => $input['c4'],

'c5'         => $input['c5'],

'c6'         => $input['c6'],

'c7'         => $input['c7'],

'c8'         => $input['c8'],

'c9'         => $input['c9'],

'c10'        => $input['c10']

);

$this->m_survei->simpan_data_detail($data);

redirect('c_homeSurvei/tabelSurveiDetail');

}

public function form_update_survei($id){

    $data['status'] = $this->m_survei->select_statusSurvei();

    $data['upt'] = $this->m_survei->select_upt();

    $data['update_survei']=$this->m_survei->ambil_data($id);

    $this->load->view('update_survei_admin',$data);
```

```
}  
  
public function form_update_survei_survei($id){  
  
    $data['status'] = $this->m_survei->select_statusSurvei();  
  
    $data['upt'] = $this->m_survei->select_upt();  
  
    $data['update_survei']=$this->m_survei->ambil_data($id);  
  
    $this->load->view('update_survei_survei',$data);  
  
}  
  
public function form_survei_detail($id){  
  
    $session = $this->m_session->get_session_id();  
  
    $sids = $session['session_id'];  
  
    $data['id_user']=$sids;  
  
    $data['lhr'] = $this->m_survei->select_lhr();  
  
    $data['tr'] = $this->m_survei->select_tipeRetak();  
  
    $data['lr'] = $this->m_survei->select_lebarRetak();  
  
    $data['lk'] = $this->m_survei->select_luasKerusakan();  
  
    $data['ka'] = $this->m_survei->select_kedalamanAlur();  
  
    $data['lt'] = $this->m_survei->select_luasTambalan();  
  
    $data['ll'] = $this->m_survei->select_luasLubang();  
  
    $data['kp'] = $this->m_survei->select_kekasaranPermukaan();  
  
    $data['a'] = $this->m_survei->select_amblas();  
  
    $data['lm'] = $this->m_survei->select_lapMasy();  
  
    $data['survei_detail']=$this->m_survei->ambil_data($id);  
  
    $data['join']=$this->m_survei->joinTabelSurvei();  
  
}
```

```
foreach($data['join']->result_array() as $row2)

foreach($data['survei_detail']->result_array() as $row)

if($row["id_upt"]== $row2["id_upt"]){

    $id_upt = $row['id_upt'];

    $data['jalan'] = $this->m_survei->select_jalan($id_upt);

}

else{

}

$this->load->view('form_survei_admin_detail',$data);

}

public function form_survei_detail_survei($id){

    $session = $this->m_session->get_session_id();

    $sids = $session['session_id'];

    $data['id_user']=$sids;

    $data['lhr'] = $this->m_survei->select_lhr();

    $data['tr'] = $this->m_survei->select_tipeRetak();

    $data['lr'] = $this->m_survei->select_lebarRetak();

    $data['lk'] = $this->m_survei->select_luasKerusakan();

    $data['ka'] = $this->m_survei->select_kedalamanAlur();

    $data['lt'] = $this->m_survei->select_luasTambalan();

    $data['ll'] = $this->m_survei->select_luasLubang();
```

```
$data['kp'] = $this->m_survei->select_kekasaranPermukaan();

$data['a'] = $this->m_survei->select_amblas();

$data['lm'] = $this->m_survei->select_lapMasy();

$data['survei_detail']=$this->m_survei->ambil_data($id);

$data['join']=$this->m_survei->joinTabelSurvei();

foreach($data['join']->result_array() as $row2)
foreach($data['survei_detail']->result_array() as $row)

if($row["id_upt"]== $row2["id_upt"]){

    $id_upt = $row['id_upt'];

    $data['jalan'] = $this->m_survei->select_jalan($id_upt);

}

else{

}

$this->load->view('form_survei_survei_detail',$data);

}

public function tabel_survei_detail($id){

    $data['data_survei_detail']=$this->m_survei->ambil_data_surveidetail($id);

    $this->load->view('tabel_survei_admin_detail',$data);

}
```

```
public function tabel_survei_survei_detail($id){

    $data['data_survei_detail']=$this->m_survei->ambil_data_surveidetail($id);

    $this->load->view('tabel_survei_survei_detail',$data);

}

public function update(){

    $input ['id_survei']      = $this->input->post('id_survei');

    $input ['tanggal']       = $this->input->post('tgl');

    $input ['kegiatan']      = $this->input->post('kegiatan');

    $input ['status']        = $this->input->post('status');

    $input ['upt']           = $this->input->post('upt');

    // yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

    $this->m_survei->update_data($input);

    redirect('c_homeAdmin/tabelSurvei');

}

public function updateSurvei(){

    $input ['id_survei']      = $this->input->post('id_survei');

    $input ['tanggal']       = $this->input->post('tgl');

    $input ['kegiatan']      = $this->input->post('kegiatan');

    $input ['status']        = $this->input->post('status');
```



```
$input ['upt'] = $this->input->post('upt');

// yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

$this->m_survei->update_data($input);

redirect('c_homeSurvei/tabelSurvei');
}

public function hapus($id=""){
    $this->m_survei->hapus_data($id);

    redirect('c_homeAdmin/tabelSurvei');
}

public function hapusSurvei($id=""){
    $this->m_survei->hapus_data($id);

    redirect('c_homeSurvei/tabelSurvei');
}

public function hapusSurveiDetail($id=""){
    $this->m_survei->hapus_surveidetail($id);

    redirect('c_homeAdmin/tabelSurveiDetail');
}

public function hapusSurveiDetailSurvei($id=""){
    $this->m_survei->hapus_surveidetail($id);

    redirect('c_homeSurvei/tabelSurveiDetail');
```

```
}

function update_text_laporan($id_jalan){

    $q = $this->m_survei->update_text_laporan($id_jalan);

    if($q->num_rows(>0){

        echo '47';

    }else{

        echo '48';

    }

}

}

}

?>
```

#### 6. Kode Program *Model* survei (m\_survei)

```
<?php

if (!defined('BASEPATH'))

    exit('No direct script access allowed');

Class m_survei extends CI_Model {

    function __construct() {
```

```
parent::__construct();

}

function load_data_survei(){

$result = $this->db->query("select * from tb_survei a, tb_status_survei b, tb_upt c

    where a.id_status = b.id_status AND a.id_upt = c.id_upt

    order by id_survei ASC");

return $result;

}

function load_data_surveidetail(){

$result = $this->db->query("select id_survei, id_survei_detail, nama_jalan, c.nama_subsubkriteria as s1,

    d.nama_subsubkriteria as s2,

    e.nama_subsubkriteria as s3,

    f.nama_subsubkriteria as s4,

    g.nama_subsubkriteria as s5,

    h.nama_subsubkriteria as s6,

    i.nama_subsubkriteria as s7,

    j.nama_subsubkriteria as s8,

    k.nama_subsubkriteria as s9,

    l.nama_subsubkriteria as s10 from tb_survei_detail a

    JOIN tb_jalan b ON a.id_jalan = b.id_jalan

    JOIN tb_subsubkriteria c ON a.c1 = c.id_subsubkriteria

    JOIN tb_subsubkriteria d ON a.c2 = d.id_subsubkriteria
```

```
JOIN tb_subsubkriteria e ON a.c3 = e.id_subsubkriteria

JOIN tb_subsubkriteria f ON a.c4 = f.id_subsubkriteria

JOIN tb_subsubkriteria g ON a.c5 = g.id_subsubkriteria

JOIN tb_subsubkriteria h ON a.c6 = h.id_subsubkriteria

JOIN tb_subsubkriteria i ON a.c7 = i.id_subsubkriteria

JOIN tb_subsubkriteria j ON a.c8 = j.id_subsubkriteria

JOIN tb_subsubkriteria k ON a.c9 = k.id_subsubkriteria

JOIN tb_subsubkriteria l ON a.c10 = l.id_subsubkriteria

ORDER BY id_survei ASC");

return $result;

}

function ambil_data_surveidetail($id){

$result = $this->db->query("select id_survei, id_survei_detail, nama_jalan, c.nama_subsubkriteria as s1,

d.nama_subsubkriteria as s2,

e.nama_subsubkriteria as s3,

f.nama_subsubkriteria as s4,

g.nama_subsubkriteria as s5,

h.nama_subsubkriteria as s6,

i.nama_subsubkriteria as s7,

j.nama_subsubkriteria as s8,

k.nama_subsubkriteria as s9,

l.nama_subsubkriteria as s10 from tb_survei_detail a

JOIN tb_jalan b ON a.id_jalan = b.id_jalan
```

```
JOIN tb_subsubkriteria c ON a.c1 = c.id_subsubkriteria

JOIN tb_subsubkriteria d ON a.c2 = d.id_subsubkriteria

JOIN tb_subsubkriteria e ON a.c3 = e.id_subsubkriteria

JOIN tb_subsubkriteria f ON a.c4 = f.id_subsubkriteria

JOIN tb_subsubkriteria g ON a.c5 = g.id_subsubkriteria

JOIN tb_subsubkriteria h ON a.c6 = h.id_subsubkriteria

JOIN tb_subsubkriteria i ON a.c7 = i.id_subsubkriteria

JOIN tb_subsubkriteria j ON a.c8 = j.id_subsubkriteria

JOIN tb_subsubkriteria k ON a.c9 = k.id_subsubkriteria

JOIN tb_subsubkriteria l ON a.c10 = l.id_subsubkriteria

WHERE id_survei='$id'';

return $result;
}

function ambil_data($id){
    return $this->db->query("select * from tb_survei where id_survei='$id''");
}

function update_data($input) {

$query = $this->db->query("UPDATE tb_survei SET `tanggal`='$input[tanggal]',
`kegiatan`='$input[kegiatan]',
`id_status`='$input[status]',
`id_upt`='$input[upt]'
```



```
WHERE `id_survei`='{$input[id_survei]}');

return $query;

}

function simpan_data($data){

return $this->db->insert('tb_survei',$data);

}

function simpan_data_detail($data){

return $this->db->insert('tb_survei_detail',$data);

}

public function hapus_data($id){

$query = $this->db->query("DELETE from tb_survei where id_survei='$id'");

$query = $this->db->query("DELETE from tb_survei_detail where id_survei='$id'");

return $query;

}

public function hapus_surveidetail($id){

$query = $this->db->query("DELETE from tb_survei_detail where id_survei_detail='$id'");

return $query;

}

public function select_statusSurvei(){

$query = $this->db->query("SELECT * FROM tb_status_survei");

return $query;

}
```

```
public function select_upt(){

    $query = $this->db->query("SELECT * FROM tb_upt");

    return $query;

}

public function select_lhr(){

    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=1");

    return $query;

}

public function select_tipeRetak(){

    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=2");

    return $query;

}

public function select_lebarRetak(){

    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=3");

    return $query;

}

public function select_luasKerusakan(){

    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=4");

    return $query;

}

public function select_kedalamanAlur(){

    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=5");

    return $query;

}
```

```
}  
  
public function select_luasTambalan(){  
  
    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=6");  
  
    return $query;  
  
}  
  
public function select_luasLubang(){  
  
    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=7");  
  
    return $query;  
  
}  
  
public function select_kekasaranPermukaan(){  
  
    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=8");  
  
    return $query;  
  
}  
  
public function select_amblas(){  
  
    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=9");  
  
    return $query;  
  
}  
  
public function select_lapMasy(){  
  
    $query = $this->db->query("select * from tb_subsubkriteria where id_subkriteria=10");  
  
    return $query;  
  
}  
  
public function joinTabelSurvei(){  
  
    $query = $this->db->query("SELECT * from tb_survei c JOIN tb_kecamatan a
```

```
        ON c.id_upt = a.id_upt JOIN tb_jalan b
        ON a.id_kecamatan = b.id_kecamatan");

return $query;
}

public function select_jalan($id_upt){
    $query = $this->db->query("SELECT * from tb_kecamatan a JOIN tb_jalan b
        ON a.id_kecamatan = b.id_kecamatan where id_upt='$id_upt'");
    return $query;
}

public function update_text_laporan($id_jalan){
    $bulan = 6;
    $waktulalu = mktime(0, 0, 0, date("m") - $bulan, date("d"), date("Y"));
    $waktu1 = date("Y-m-d", $waktulalu);
    $waktu2 = date("Y-m-d");
    $query = $this->db->query("SELECT * from tb_lapmasy where id_jalan='$id_jalan' AND
        (tanggal BETWEEN '$waktu1' AND '$waktu2')");
    return $query;
}
}

?>
```

## 7. Kode Program *Controller* user (c\_user)

```
<?php

if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class c_user extends CI_Controller {

    public function __construct() {
        parent::__construct();

        $this->load->model('m_user');

        $this->load->model('m_login');

        $this->load->model('m_session');
    }

    public function user(){

        $session = $this->m_login->get_session_userid();

        $id = $session['session_userid'];

    }

    public function insert(){ // yg depan variabel array, yg belakang nama variabel dr form

        $input ['username']      = $this->input->post('username');
```



```
$input ['password']      = $this->input->post('password');

$input ['level_user']    = $this->input->post('level_user');

$input ['nama']          = $this->input->post('nama');

$input ['no_telp']       = $this->input->post('no_telp');

$input ['email']         = $this->input->post('email');

$input ['alamat']        = $this->input->post('alamat');

// yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

$data = array(

    'username'           => $input['username'],

    'password'           => $input['password'],

    'id_level_user'      => $input['level_user'],

    'nama'                => $input['nama'],

    'no_telp'            => $input['no_telp'],

    'email'              => $input['email'],

    'alamat'             => $input['alamat']

);

$this->m_user->simpan_data($data);

redirect('c_homeAdmin/tabelUser');

}
```

```
public function form_update_user($id){

    $data['update_user']=$this->m_user->ambil_data($id);

    $data['lu'] = $this->m_user->select_levelUser();

    $this->load->view('update_user_admin',$data);

}

public function update(){

    $input ['id_user']      = $this->input->post('id_user');

    $input ['username']    = $this->input->post('username');

    $input ['password']    = $this->input->post('password');

    $input ['level_user']  = $this->input->post('level_user');

    $input ['nama']        = $this->input->post('nama');

    $input ['no_telp']     = $this->input->post('no_telp');

    $input ['email']       = $this->input->post('email');

    $input ['alamat']      = $this->input->post('alamat');

    // yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

    $this->m_user->update_data($input);

    redirect('c_homeAdmin/tabelUser');

}
```

```
public function hapus($id=""){  
  
    $this->m_user->hapus_data($id);  
  
    redirect('c_homeAdmin/tabelUser');  
  
}  
  
}  
  
?>
```

#### 8. Kode Program *Model* user (m\_user)

```
<?php  
  
if (!defined('BASEPATH'))  
    exit('No direct script access allowed');  
  
Class m_user extends CI_Model {  
  
    function __construct() {  
        parent::__construct();  
    }  
  
    function load_data_user(){  
  
        $result = $this->db->query("select id_user, username, password, nama, no_telp, email, alamat,  
a.id_level_user, level from tb_level_user a JOIN tb_user b
```

```
ON a.id_level_user = b.id_level_user order by `id_user` ASC");

return $result;

}

function load_level_user(){

$result = $this->db->query("select * from tb_level_user order by `id_level_user` ASC");

return $result;

}

function ambil_data($id){

return $this->db->query("select id_user, username, password, nama, no_telp, email, alamat, a.id_level_user,
level from tb_level_user a JOIN tb_user b

ON a.id_level_user = b.id_level_user where id_user='$id'");

}

function update_data($input) {

$query = $this->db->query("UPDATE tb_user SET `username`='{$input[username]}',
`password`='{$input[password]}',
`id_level_user`='{$input[level_user]}',
`nama`='{$input[nama]}',
`no_telp`='{$input[no_telp]}',
`email`='{$input[email]}',
`alamat`='{$input[alamat]}

WHERE `id_user`='{$input[id_user]}");

return $query;

}
```

```
}  
  
function simpan_data($data){  
    return $this->db->insert('tb_user',$data);  
}  
  
public function hapus_data($id){  
    $query = $this->db->query("DELETE from tb_user where id_user=$id");  
    return $query;  
}  
  
public function select_levelUser(){  
    $query = $this->db->query("select * from tb_level_user");  
    return $query;  
}  
}  
  
?>
```

#### 9. Kode Program *Controller* jalan (c\_jalan)

```
<?php  
  
if (!defined('BASEPATH'))  
    exit('No direct script access allowed');
```



```
class c_jalan extends CI_Controller {

    public function __construct() {

        parent::__construct();

        $this->load->model('m_user');

        $this->load->model('m_login');

        $this->load->model('m_jalan');

        $this->load->model('m_session');

    }

    public function user(){

        $session = $this->m_login->get_session_userid();

        $id = $session['session_userid'];

    }

    public function insert(){ // yg depan variabel array, yg belakang nama variabel dr form

        $input ['klasifikasi_ruas']      = $this->input->post('klasifikasi_ruas');

        $input ['kecamatan']              = $this->input->post('kecamatan');

        $input ['nama_jalan']              = $this->input->post('nama_jalan');

        $input ['panjang_ruas']           = $this->input->post('panjang_ruas');

        $input ['lebar_ruas']              = $this->input->post('lebar_ruas');
```

```
// yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)

$data = array(

    'id_klasifikasi_ruas'    => $input['klasifikasi_ruas'],

    'id_kecamatan'         => $input['kecamatan'],

    'nama_jalan'           => $input['nama_jalan'],

    'panjang_ruas'        => $input['panjang_ruas'],

    'lebar_ruas'           => $input['lebar_ruas']

);

$this->m_jalan->simpan_data($data);

redirect('c_homeAdmin/tabelJalan');

}

public function update(){

    $input ['id_jalan']      = $this->input->post('id_jalan');

    $input ['klasifikasi_ruas'] = $this->input->post('klasifikasi_ruas');

    $input ['kecamatan']     = $this->input->post('kecamatan');

    $input ['nama_jalan']    = $this->input->post('nama_jalan');

    $input ['panjang_ruas']  = $this->input->post('panjang_ruas');

    $input ['lebar_ruas']    = $this->input->post('lebar_ruas');

}

// yg depan atribut database, yg belakang nama variabel array(sama kyak diatas)
```

```
$this->m_jalan->update_data($input);

redirect('c_homeAdmin/tabelJalan');

}

public function hapus($id=""){

$this->m_jalan->hapus_data($id);

redirect('c_homeAdmin/tabelJalan');

}

public function form_update_jalan($id){

$data['kr']= $this->m_jalan->select_klasifikasi();

$data['k']= $this->m_jalan->select_kecamatan();

$data['update_jalan']=$this->m_jalan->ambil_data($id);

$this->load->view('update_jalan_admin',$data);

}

}

?>
```

#### 10. Kode Program *Model* jalan (m\_jalan)

```
<?php
```

```
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

Class m_jalan extends CI_Model {

    function __construct() {
        parent::__construct();
    }

    function load_data_jalan(){

        $result = $this->db->query("select * from tb_jalan a, tb_kecamatan b, tb_klasifikasi_ruas c

            where a.id_kecamatan = b.id_kecamatan AND a.id_klasifikasi_ruas = c.id_klasifikasi_ruas

            order by id_jalan ASC");

        return $result;
    }

    function ambil_data($id){

        return $this->db->query("select id_jalan, nama_jalan, panjang_ruas, lebar_ruas, a.id_klasifikasi_ruas,
a.nama_klasifikasi, c.id_kecamatan, c.kecamatan from tb_klasifikasi_ruas a JOIN tb_jalan b

            ON a.id_klasifikasi_ruas = b.id_klasifikasi_ruas

            JOIN tb_kecamatan c

            ON c.id_kecamatan = b.id_kecamatan where id_jalan='$id'");

    }

    function update_data($input) {
```

```
$query = $this->db->query("UPDATE tb_jalan SET `id_klasifikasi_ruas`='$input[klasifikasi_ruas]',  
  
`id_kecamatan`='$input[kecamatan]',  
  
`nama_jalan`='$input[nama_jalan]',  
  
`panjang_ruas`='$input[panjang_ruas]',  
  
`lebar_ruas`='$input[lebar_ruas]'  
  
WHERE `id_jalan`='$input[id_jalan]");  
  
return $query;  
  
}  
  
function simpan_data($data){  
  
    return $this->db->insert('tb_jalan',$data);  
  
}  
  
public function hapus_data($id){  
  
    $query = $this->db->query("DELETE from tb_jalan where id_jalan=$id");  
  
    return $query;  
  
}  
  
public function select_klasifikasi(){  
  
    $query = $this->db->query("select * from tb_klasifikasi_ruas");  
  
    return $query;  
  
}  
  
public function select_kecamatan(){  
  
    $query = $this->db->query("select * from tb_kecamatan");
```



```
return $query;  
  
}  
  
?>
```

