



**RANCANG BANGUN APLIKASI PENDISTRIBUSIAN PENGISIAN
ANJUNGAN TUNAI MANDIRI MENGGUNAKAN ALGORITMA
*CHEAPEST INSERTION HEURISTIC***

SKRIPSI

oleh :

Kadek Budhi Widyastiti

112410101030

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015



**RANCANG BANGUN APLIKASI PENDISTRIBUSIAN PENGISIAN
ANJUNGAN TUNAI MANDIRI MENGGUNAKAN ALGORITMA
*CHEAPEST INSERTION HEURISTIC***

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 :

Kadek Budhi Widyastiti

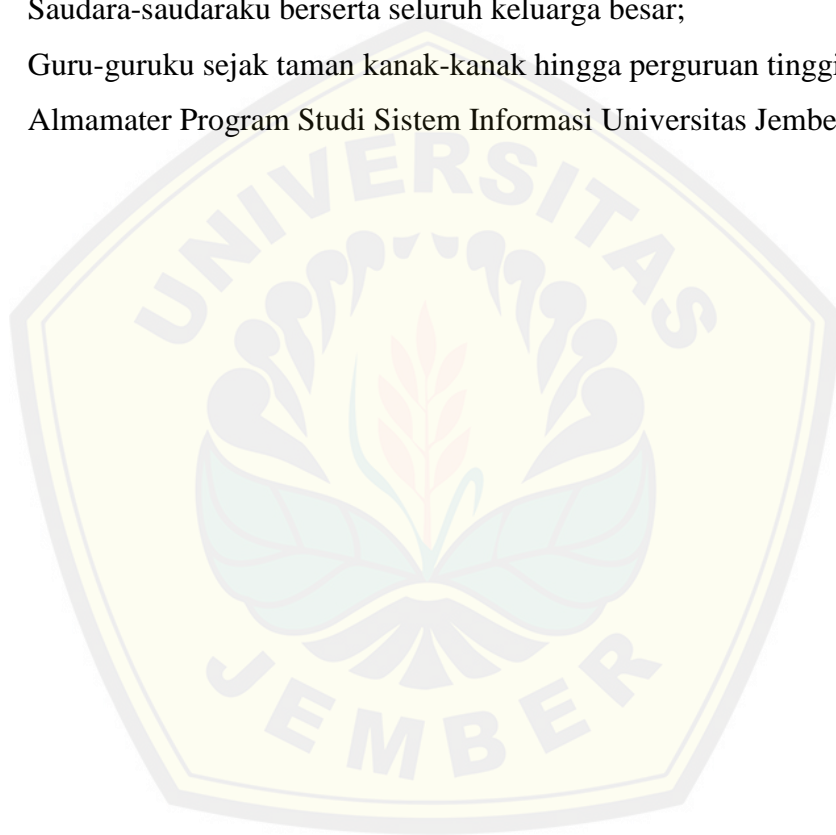
112410101030

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2015**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Hyang Widhi Wasa, yang telah memberikan kelancaran dan kemudahan dalam menyelesaikan skripsi ini
2. Ibunda tercinta Ketut Widiasih dan Ayahanda Nyoman Sudira;
3. Adik Komang Budhi Widyastana, Kakak Putu Budhi Widyarsa dan ;
4. Saudara-saudaraku beserta seluruh keluarga besar;
5. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;
6. Almamater Program Studi Sistem Informasi Universitas Jember.



MOTO

“Hasil Tidak Menghianati Usaha”



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Kadek Budhi Widyastiti

NIM : 112410101030

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Rancang Bangun Aplikasi Pendistribusian Pengisian Anjungan Tunai Mandiri menggunakan Algoritma *Cheapest Insertion Heuristic*” adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 25 Juni 2015

Yang menyatakan,

Kadek Budhi Widyastiti

NIM 112410101030

PENGESAHAN PEMBIMBING

Skripsi berjudul “Rancang Bangun Aplikasi Pendistribusian Pengisian Anjungan Tunai Mandiri menggunakan Algoritma *Cheapest Insertion Heuristic*”, telah diuji dan disahkan pada :

hari, tanggal : Senin, 25 Juni 2015

tempat : Program Studi Sistem Informasi Universitas Jember.

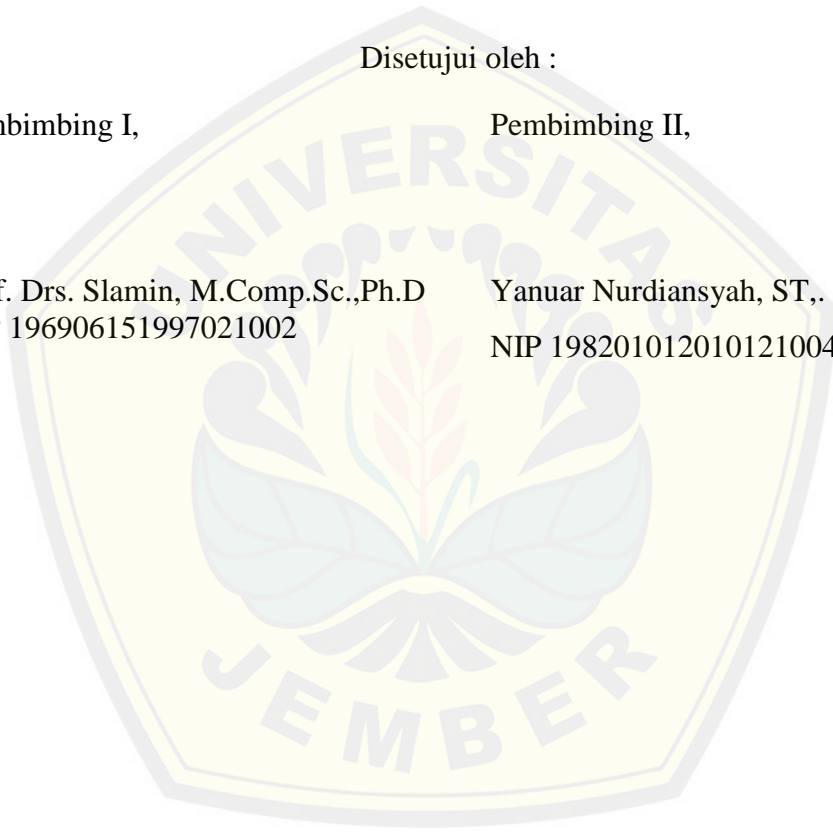
Disetujui oleh :

Pembimbing I,

Pembimbing II,

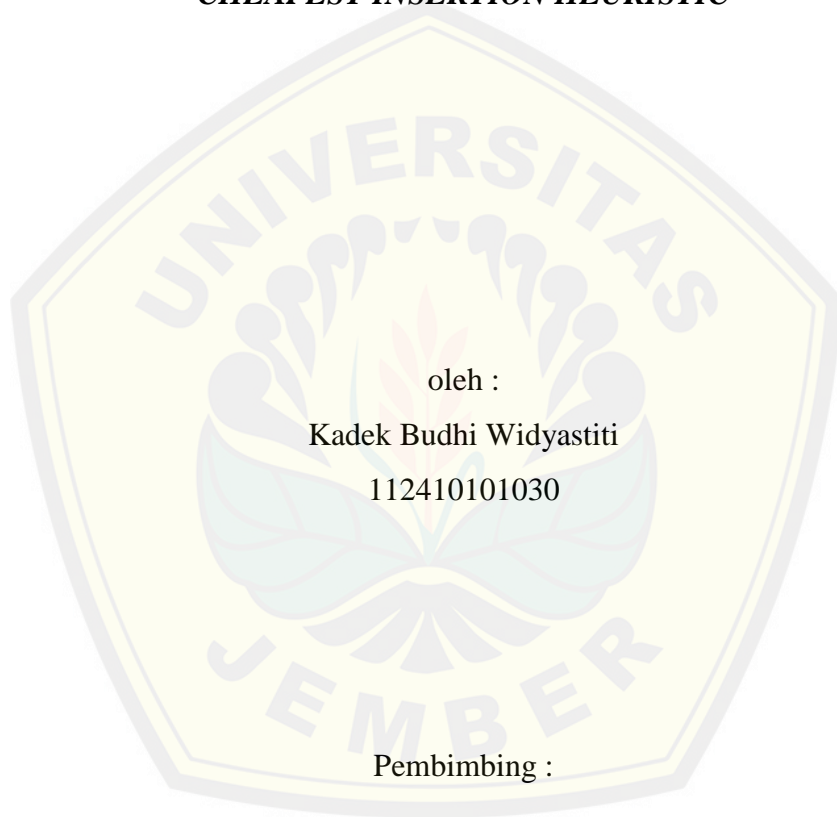
Prof. Drs. Slamini, M.Comp.Sc.,Ph.D
NIP 196906151997021002

Yanuar Nurdiansyah, ST., M.Cs.
NIP 198201012010121004



SKRIPSI

**RANCANG BANGUN APLIKASI PENDISTRIBUSIAN PENGISIAN
ANJUNGAN TUNAI MANDIRI MENGGUNAKAN ALGORITMA
*CHEAPEST INSERTION HEURISTIC***



oleh :

Kadek Budhi Widyastiti

112410101030

Pembimbing :

Dosen Pembimbing Utama : Prof. Drs. Slamir, M.Comp.Sc.,Ph.D
Dosen Pembimbing Pendamping : Yanuar Nurdiansyah, ST., M.Cs.

PENGESAHAN

Skripsi berjudul “Rancang Bangun Aplikasi Pendistribusian Pengisian Anjungan Tunai Mandiri menggunakan Algoritma *Cheapest Insertion Heuristic*”, telah diuji dan disahkan pada :

hari, tanggal : Senin, 25 Juni 2015

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji :

Penguji I,

Penguji II,

Drs. Antonius Cahya P, M.App., Sc

NIP 196909281993021001

Nelly Oktavia Adiwijaya, S.Si., MT.

NIP 198410242009122008

Mengesahkan

Ketua Program Studi,

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

NIP 196704201992011001

RINGKASAN

Rancang Bangun Aplikasi Pendistribusian Pengisian Anjungan Tunai Mandiri menggunakan Algoritma *Cheapest Insertion Heuristic*; Kadek Budhi Widyastiti; 112410101030; 150 halaman; Program Studi Sistem Informasi Universitas Jember.

Anjungan Tunai Mandiri (ATM) adalah alat kasir otomatis yang menggantikan sebagian besar fungsi dari seorang kasir (*teller*). Nasabah lebih sering melakukan transaksi dengan menggunakan ATM dibandingkan dengan menuju bank-bank yang ada. Pemakaian ATM yang sangat dibutuhkan oleh para nasabah membuat pendistribusian untuk mengisi kas ATM sangatlah penting. ATM tidak boleh kosong sehingga menyebabkan keterlambatan dalam proses transaksi.

Jumlah ATM yang terdapat pada bank BRI di Kabupaten Jember sangatlah banyak, sehingga dalam pendistribusian pengisian kas ATM dilakukan setiap hari ketika terdapat ATM dengan kas yang kosong. Pada penelitian ini dibangun Aplikasi Pendistribusian Pengisian ATM untuk mengoptimalkan rute perjalanan pendistribusian pengisian ATM yang akan ditempuh dan dapat tersistemnya pencatatan yang sebelumnya masih secara manual.

Metode yang digunakan adalah algoritma *Cheapest Insertion Heuristic* yakni metode yang melakukan pengulangan untuk mencari jarak terkecil dengan cara menyisipkan nilai yang baru hingga akhirnya membentuk sebuah *subtour* atau rute perjalanan yang optimal. Sistem ini memvisualisasikan hasil dari rute perjalanan pendistribusian pengisian ATM BRI menggunakan bantuan *Google Maps*. Sistem ini dibangun menggunakan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP) dan framework *Code Igniter* (CI). Berdasarkan pengujian yang dilakukan terhadap aplikasi pendistribusian pengisian ATM maka rute perjalanan yang ditempuh menjadi lebih optimal, lebih efisien, dan lebih memudahkan *user* untuk menggunakan sistem ini.

PRAKATA

Puji syukur kehadiran Hyang Widhi Wasa atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Aplikasi Pendistribusian Pengisian Anjungan Tunai Mandiri menggunakan Algoritma *Cheapest Insertion Heuristics*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

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

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember dan juga selaku Dosen Pembimbing Utama;
2. Yanuar Nurdiansyah, ST., M.Cs selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Anang Andrianto S.T.,MT., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ibunda tercinta Ketut Widiasih dan Ayahanda Nyoman Sudira yang telah sangat memotivasi penulis;
6. Adik Komang Budhi Widyastana dan Kakak Putu Budhi Widyarsa;
7. Saudara-saudaraku beserta seluruh keluarga besar;
8. Sahabat seperjuangan yang selalu menemani dan memberikan semangat Permadi, Nur, Sri, Adindayu, Trisna, Rawinka, Rama dan Choriawan;
9. Keluarga besar NEFOTION semua mahasiswa Program Studi Sistem Informasi angkatan 2011 yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
10. Keluarga besar Himpunan Mahasiswa Sistem Informasi (HIMASIF) periode 2011-2012
11. Keluarga besar SISFO CUP periode 2011-2014

12. Keluarga besar UKM Hindu Dharma Universitas Jember;
13. Keluarga besar UKM Bola Basket Universitas Jember;
14. Keluarga besar Paskibra Universitas Jember angkatan 2011;
15. Keluarga besar UKM Program Studi Sistem Informasi Universitas Jember;
16. Almamater, guru dan teman-teman tercinta pada Taman Kanak-kanak di TK Pertiwi Jember;
17. Almamater, guru dan teman-teman tercinta pada Sekolah Dasar di SD Negeri Kebonsari 08;
18. Almamater, guru dan teman-teman tercinta pada Sekolah Menengah Pertama di SMP Negeri 1 Jember;
19. Almamater, guru dan teman-teman tercinta pada Sekolah Menengah Atas di SMA Negeri 2 Jember;
20. PT. Bank Rakyat Indonesia (Persero) Tbk Cabang Jember;
21. Semua pihak yang tidak dapat disebutkan satu-persatu.

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

Jember, 25 Juni 2015

Penulis

DAFTAR ISI

PERSEMBAHAN	ii
MOTO	iii
PERNYATAAN	iv
SKRIPSI	vi
PENGESAHAN	vii
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 Rumusan Masalah	2
1.3 Tujuan dan Manfaat	2
1.3.1 Tujuan.....	3
1.3.2 Manfaat.....	3
1.4 Ruang Lingkup	3
1.5 Sistematika Penulisan	4
BAB 2. TINJUAN PUSTAKA	6
2.1 Penelitian Terdahulu	6
2.2 Graf	7
2.3 Travelling Salesman Problem	8
2.4 Distribusi	9
2.5 Anjungan Tunai Mandiri	9
2.6 Algoritma	10
2.7 Cheapest Insertion Heuristic	11
2.8 Google Maps	12
BAB 3. METODOLOGI PENELITIAN	13
3.1. Jenis Penelitian	13

3.2. Metode Penelitian.....	13
3.3. Tempat dan Waktu Penelitian.....	14
3.4. Tahapan Penelitian.....	14
3.4.1. Tahap Pengumpulan Data.....	15
3.4.2. Tahap Analisis.....	17
3.4.3. Tahap Pengembangan Sistem.....	27
3.5. Gambaran Umum Sistem.....	32
BAB 4. PENGEMBANGAN SISTEM.....	34
4.1 <i>Statement Of Purpose</i>	34
4.2 Analisis Kebutuhan Sistem.....	34
4.2.1 Kebutuhan Fungsional.....	35
4.2.2 Kebutuhan Non-Fungsional.....	35
4.3 Desain Sistem.....	35
4.3.1 <i>Business Process</i>	35
4.3.2 <i>Usecase Diagram</i>	36
4.3.3 <i>Usecase Skenario</i>	40
4.3.4 <i>Activity Diagram</i>	44
4.3.5 <i>Sequence Diagram</i>	45
4.3.6 <i>Class Diagram</i>	52
4.3.7 <i>Entity Relationship Diagram</i>	54
4.4 Penulisan Kode Program.....	54
4.5 Pengujian Sistem.....	59
4.5.1 <i>White Box Testing</i>	59
4.5.2 <i>Black Box Testing</i>	72
BAB 5. HASIL DAN PEMBAHASAN.....	76
5.1 Hasil Penerapan Algoritma <i>Cheapest Insertion Heuristic</i> pada Pencarian Optimasi Rute.....	76
5.2 Hasil Pembuatan Aplikasi Pendistribusian Pengisian ATM	76
5.2.1 Fitur <i>Login</i>	77
5.2.2 Halaman Melihat Peta ATM Admin.....	77
5.2.3 Fitur Manajemen Data ATM.....	78

5.2.4	Fitur Perhitungan Jarak.....	80
5.2.5	Fitur Perhitungan Rute.....	81
5.3	Hasil Penerapan Algoritma <i>CIH</i> Pada Aplikasi.....	82
5.3.1	Penginputan ATM yang akan menjadi objek.....	83
5.3.2	Pengecekan nilai awal, nilai tujuan, dan nilai sisip.....	83
5.3.3	Input nilai sisip ke dalam penyimpanan temporary proses.	84
5.3.4	Perhitungan nilai awal, nilai sisip, nilai tujuan dan nilai terkecil.....	84
5.3.5	Membalikan nilai awal ke tujuan menjadi tujuan ke awal..	85
5.3.6	Pemilihan nilai terkecil.....	85
5.3.7	Pembentukan <i>subtour</i> baru.....	86
5.3.8	Mencetak nilai penyimpanan <i>temporary</i> hasil, penyimpanan <i>temporary</i> proses.....	86
5.3.9	Perhitungan total jarak seluruh rute.....	87
5.4	Hasil Perbandingan Pendistribusian Pengisian ATM Secara Konvensional dan Menggunakan Sistem.....	88
5.5.	Pembahasan Aplikasi Pendistribusian Pengisian ATM.....	89
BAB 6.	PENUTUP.....	90
6.1	Kesimpulan.....	90
6.2	Saran.....	90
DAFTAR PUSTAKA.....		91
LAMPIRAN.....		92

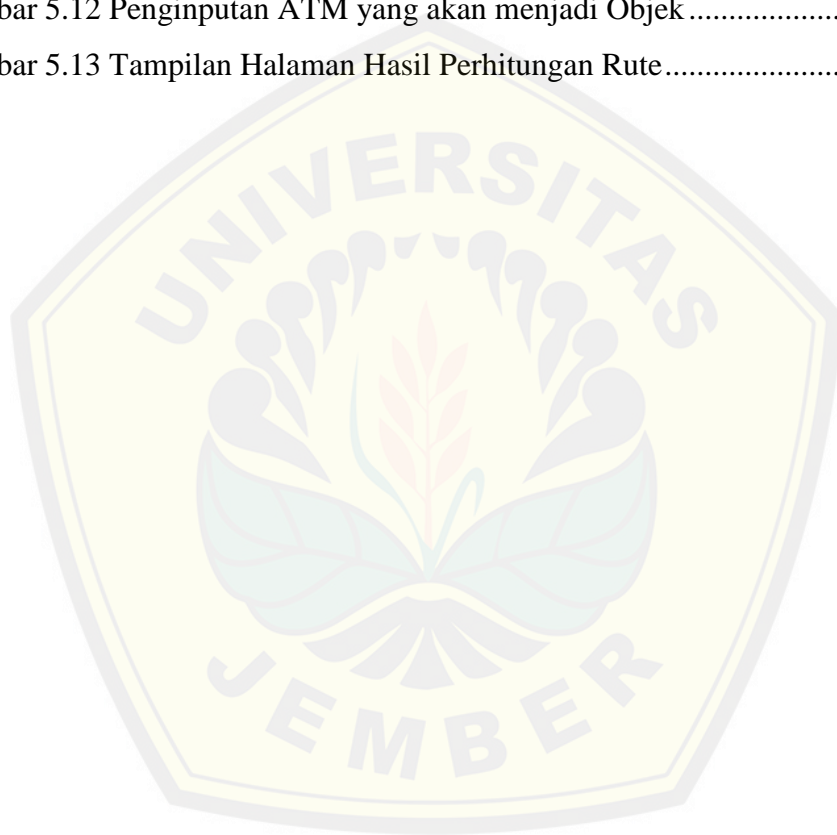
DAFTAR TABEL

Tabel 3.1 Data ATM BRI.....	17
Tabel 3.2 Perhitungan Jarak.....	23
Tabel 3.3 Hasil 1	24
Tabel 3.4 Proses 1	25
Tabel 3.5 Hasil 2	25
Tabel 3.6 Proses 2	25
Tabel 3.7 Hasil 3	26
Tabel 3.8 Proses 3	26
Tabel 4.1 Definisi Aktor	39
Tabel 4.2 Definisi <i>Usecase</i>	39
Tabel 4.3 <i>Usecase</i> Skenario Menghitung Rute	41
Tabel 4.4 Kode Program <i>Function</i> <i>haber()</i>	55
Tabel 4.5 <i>Test Case</i> Fitur Manajemen Data ATM.....	63
Tabel 4.6 <i>Test Case</i> Fitur Manajemen Data ATM.....	71
Tabel 4.7 Hasil Pengujian <i>Black Box</i> Perhitungan Rute	73
Tabel 5.1 Kode Program Pengecekan ATM	84
Tabel 5.2 Kode Program Input Nilai Sisip.....	84
Tabel 5.3 Kode Program Perhitungan Nilai Awal, Nilai Sisip, Nilai Tujuan, Nilai Terkecil	84
Tabel 5.4 Kode Program Membalik Arah Tujuan	85
Tabel 5.5 Kode Program Pemilihan Nilai Terkecil.....	85
Tabel 5.6 Kode Program Pembentukan <i>Subtour</i> Baru	86
Tabel 5.7 Kode Program Mencetak Nilai Penyimpanan <i>Temporary</i>	86
Tabel 5.8 Kode Program Perhitungan Total Jarak seluruh Rute.....	87
Tabel 1.1 <i>Usecase</i> Skenario <i>Login</i>	92

DAFTAR GAMBAR

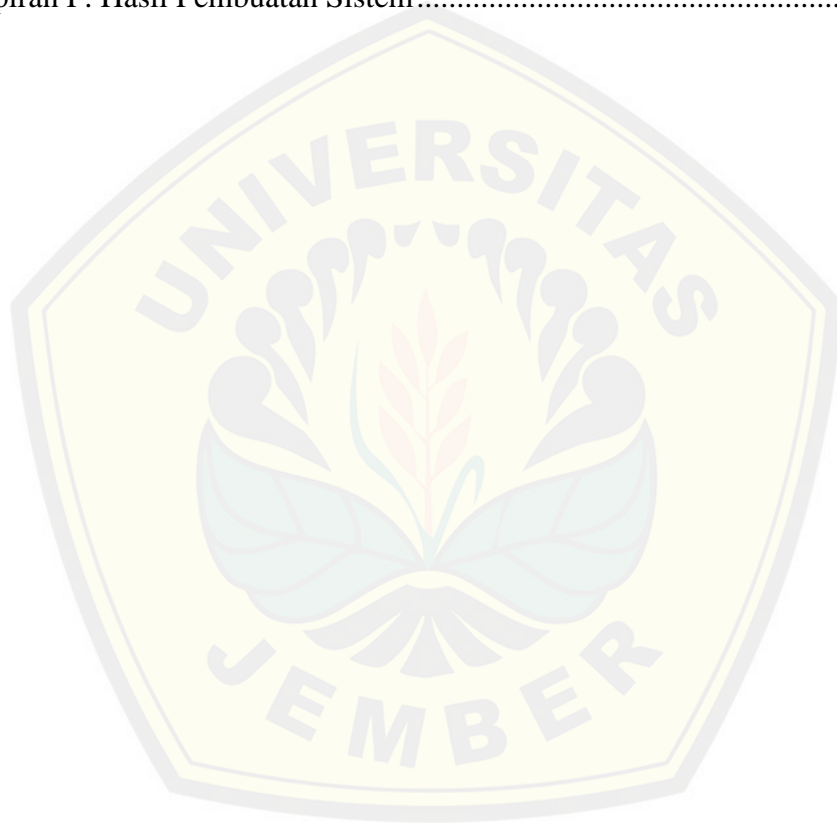
Gambar 2.1 Contoh Sebuah Graf	8
Gambar 2.2 Contoh <i>Subtour</i>	11
Gambar 3.1 Diagram Alir Tahapan Penelitian.....	14
Gambar 3.2 Diagram Blok Tahapan Analisis	18
Gambar 3.3 Diagram Alir Algoritma <i>CIH</i>	19
Gambar 3.4 Peta ATM BRI di Kabupaten Jember	21
Gambar 3.5 Sampel ATM BRI	22
Gambar 3.6 Representasi Peta ATM BRI dengan Graf	23
Gambar 3.7 Peta ATM BRI dengan metode <i>CIH</i>	27
Gambar 3.8 Tahapan Model <i>Waterfall</i>	28
Gambar 3.9 <i>Flowchart</i> sistem Pendistribusian Pengisian ATM.....	33
Gambar 4.1 <i>Bussiness Process</i> Pendistribusian Pengisian ATM.....	36
Gambar 4.2 <i>Usecase Diagram</i> Pendistribusian Pengisian ATM.....	37
Gambar 4.3 <i>Activity Diagram</i> Perhitungan Rute.....	45
Gambar 4.4 <i>Sequence Diagram</i> Menghitung Rute.....	51
Gambar 4.5 <i>Class Diagram</i> Pendistribusian Pengisian ATM.....	53
Gambar 4.6 <i>ERD</i> Pendistribusian Pengisian ATM	54
Gambar 4.7 <i>Listing</i> Program Menambah Data ATM.....	60
Gambar 4.8 <i>Listing</i> Program Mengubah Data ATM	60
Gambar 4.9 <i>Listing</i> Program Menghapus Data ATM.....	61
Gambar 4.10 Diagram Alir Manajemen Data ATM	61
Gambar 4.11 Grafik Alir Manajemen Data ATM.....	62
Gambar 4.12 <i>Listing</i> Program Perhitungan Rute	67
Gambar 4.13 Diagram Alir Perhitungan Rute.....	68
Gambar 4.14 Grafik Alir Perhitungan Rute	70
Gambar 5.1 Tampilan Halaman <i>Login</i>	77
Gambar 5.2 Tampilan Halaman Melihat Peta ATM Admin.....	78
Gambar 5.3 Tampilan Halaman Melihat Peta ATM Admin pada Keterangan.....	78
Gambar 5.4 Tampilan Halaman Mengelola Data ATM.....	79

Gambar 5.5 Tampilan Halaman Menambahkan Data ATM.....	79
Gambar 5.6 Tampilan Halaman Mengedit data ATM.....	80
Gambar 5.7 Tampilan Halaman Menghapus Data ATM.....	80
Gambar 5.8 Tampilan Halaman Perhitungan Jarak	81
Gambar 5.9 Tampilan Halaman Modal Perhitungan Jarak.....	81
Gambar 5.10 Tampilan Halaman Perhitungan Rute	82
Gambar 5.11 Tampilan Halaman Hasil Perhitungan Rute.....	82
Gambar 5.12 Penginputan ATM yang akan menjadi Objek	83
Gambar 5.13 Tampilan Halaman Hasil Perhitungan Rute.....	88



DAFTAR LAMPIRAN

Lampiran A. <i>Usecase</i> Skenario.....	93
Lampiran B. <i>Activity</i> Diagram	105
Lampiran C. <i>Sequence</i> Diagram	114
Lampiran D. Kode Program.....	127
Lampiran E. Pengujian <i>Black Box</i>	140
Lampiran F. Hasil Pembuatan Sistem.....	148



BAB 1. PENDAHULUAN

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

1.1 Latar Belakang

Penentuan pencarian rute optimum adalah aktifitas yang sangat penting yang dapat digunakan dalam berbagai contoh permasalahan, seperti pengiriman pos, pengiriman paket, rute patroli, rute pengisian bensin, rute pendistribusian produk dari gudang dengan gudang lainnya, dan juga rute pengisian Anjungan Tunai Mandiri (ATM). Dengan pemilihan rute yang optimal, maka seseorang akan dapat meminimalisir permasalahan yang ada.

Permasalahan ini biasa disebut dengan *Travelling Salesman Problem (TSP)* yakni pencarian rute optimum dalam suatu perjalanan yang mencari jarak terpendek untuk mencapai tempat tujuan. Penggambaran sederhana dari istilah *TSP* adalah permasalahan perjalanan seorang salesman yang harus mengunjungi n buah titik dengan aturan salesman harus mengunjungi setiap titik tepat satu kali dan pada akhirnya harus kembali ke titik asalnya. *Salesman* juga harus meminimalisir total jarak perjalanan. Metode *TSP* dapat juga dikatakan untuk memudahkan perjalanan maka pemetaan n titik tersebut akan digambarkan dengan sebuah graf, dimana jumlah titik (*vertex*) dan sisi (*edge*) terbatas. Sebuah sisi (*edge*) mewakili jarak antara dua titik yang dihubungkan.

TSP dapat di selesaikan dengan beberapa pendekatan dan algoritma, salah satunya adalah penyelesaian dengan menggunakan algoritma *Cheapest Insertion Heuristic (CIH)*. Algoritma ini memiliki konsep menyisipkan kota yang belum terlewati dengan tambahan jarak minimum hingga semua kota terlewati untuk mendapatkan solusi permasalahan (Rothlauf, 2011). Dengan menggunakan algoritma ini, maka permasalahan *TSP* akan dapat diselesaikan dengan baik dan akan mendapatkan hasil rute yang optimum.

Salah satu permasalahan yang termasuk kasus TSP adalah pendistribusian dalam pengisian ATM, khususnya di kabupaten Jember karena Bank cabang memiliki banyak sekali ATM yang tersebar di seluruh daerah-daerah kecil yang berada dalam satu lingkupan. Bank akan melakukan proses distribusi pengisian uang setiap harinya pada ATM-ATM yang telah kosong.

Prosedur pengisian selama ini adalah dari pusat pengendali sistem ATM yang memberikan informasi bahwa ada mesin ATM yang mempunyai saldo di bawah rata-rata untuk meminimalisir resiko saldo kas ATM habis. Setelah mendapatkan informasi, bank segera mengeluarkan kas kantor untuk melakukan pengisian ulang kas ATM melalui *teller* yang juga berfungsi untuk pencatatan pengeluaran dan penerimaan kas. Kemudian uang dibawa oleh petugas ATM untuk melakukan penggantian *cassete* dan *roll* elektronik dan juga kertas *receipt*.

Skripsi ini akan membahas penyelesaian permasalahan kasus TSP pada pendistribusian pengisian ATM sebuah Bank BRI di kabupaten Jember menggunakan algoritma *CIH*. Pada hal ini akan dilakukan penelitian yakni dengan menggunakan algoritma *CIH*, maka akan dapat ditemukan rute perjalanan yang optimum, dengan total jarak yang paling sedikit sehingga dapat meminimalisir penggunaan bahan bakar dan waktu tempuh perjalanan.

1.2 Rumusan Masalah

Berdasarkan uraian diatas maka dirumuskan permasalahan sebagai berikut :

1. Bagaimana menerapkan algoritma *CIH* untuk optimasi rute pendistribusian pengisian uang pada ATM Bank BRI di Kabupaten Jember?
2. Bagaimana merancang dan membangun aplikasi pendistribusian pengisian ATM Bank BRI di Kabupaten Jember dengan menerapkan algoritma *CIH*?

1.3 Tujuan dan Manfaat

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

1.3.1 Tujuan

Tujuan dari penelitian adalah :

1. Menerapkan algoritma CIH dalam proses pencarian optimasi rute untuk pendistribusian pengisian Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk. Cabang Jember.
2. Merancang dan membangun aplikasi pendistribusian pengisian Anjungan Tunai Mandiri untuk optimasi rute pada PT. BRI (Persero) Tbk Cabang Jember menggunakan algoritma CIH.

1.3.2 Manfaat

Manfaat yang ingin didapatkan dari penelitian ini adalah :

1. Manfaat Akademis
Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Penelitian ini juga diharapkan dapat menambah bermacam judul penelitian yang ada di Program Studi Sistem informasi Universitas Jember.
2. Manfaat bagi peneliti
 - a. Mengetahui bagaimana proses penerapan algoritma *Cheapest Insertion Heuristic* pada proses pendistribusian pengisian ATM.
 - b. Membantu instansi dalam pencarian rute terbaik untuk menyelesaikan permasalahan pendistribusian pengisian ATM.
3. Manfaat bagi objek penelitian
 - a. Memberikan inovasi baru kepada instansi tempat penelitian mengenai penggunaan *algoritma Cheapest Insertion Heuristic* untuk membangun aplikasi Pendistribusian Pengisian ATM.
 - b. Membantu instansi untuk melakukan pencarian rute terbaik dengan lebih optimal dan efektif.

1.4 Ruang Lingkup

Ruang lingkup atau disebut juga dengan batasan - batasan masalah dibuat oleh penulis agar memberikan batasan untuk objek dan tema yang akan diteliti,

sehingga tidak terjadi penyimpangan dalam proses penulisan dan pembuatan sistem informasi. Batasan masalah dalam pembuatan sistem ini dapat dilihat berikut ini.

1. Titik awal pendistribusian adalah ATM BRI CDM kantor cabang BRI Jember.
2. Obyek terdiri dari ATM-ATM Bank BRI yang tersebar diseluruh pelosok daerah-daerah di Kabupaten Jember.
3. Jalur yang dilalui kendaraan dalam kondisi normal.
4. Hanya ada satu kendaraan yang di gunakan ketika menerapkan aplikasi pendistribusian pengisian ATM.
5. Tingkat kepentingan antara tiap ATM bernilai sama.
6. Algoritma yang digunakan adalah algoritma CIH yang merupakan salah satu algoritma dalam penyelesaian TSP.

1.5 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

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

6. Penutup

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



BAB 2. TINJUAN PUSTAKA

Tinjauan pustaka adalah bagian yang akan menjelaskan teori-teori dan pustaka yang akan digunakan dalam penelitian, yakni tentang penelitian terdahulu, graf, *travelling salesman problem*, distribusi, ATM, algoritma, dan *cheapest insertion heuristic*, *google maps*.

2.1 Penelitian Terdahulu

Penelitian sebelumnya mengenai permasalahan TSP dilakukan oleh Indah Apriliani Universitas Jember dengan judul Penyelesaian *Travelling Salesman Problem* (TSP) menggunakan Algoritma Semut dan Algoritma *Cheapest Insertion Heuristic* (CIH) dengan mengambil permasalahan pada rute pengisian mesin ATM Bank BCA yang terletak di wilayah Jember dan Bondowoso. Penelitian ini dibuat untuk membandingkan antara kedua algoritma yakni algoritma semut atau algoritma CIH yang memiliki efektifitas lebih dalam menyelesaikan permasalahan pencarian rute terpendek. Pembuatan program menggunakan software *Delphi 7.0*, berdasarkan algoritma yang telah dibuat pada masing-masing algoritma.

Luaran dari penelitian ini adalah perbandingan hasil perhitungan rute optimal yang dihasilkan oleh program *Delphi 7.0*, algoritma CIH menghasilkan rute yang lebih optimal dibandingkan dengan algoritma untuk titik lokasi kurang dari 7. Berdasarkan waktu eksekusi program yang dibutuhkan oleh kedua algoritma, maka algoritma CIH membutuhkan waktu yang relatif lebih cepat dibandingkan dengan algoritma semut.

Penelitian yang lainnya yakni dengan judul Penyelesaian *Travelling Salesman Problem* dengan Algoritma *Cheapest Insertion Heuristic* dan Basis Data yang dilakukan oleh Kusri dan Jazi Eko Istiyanto. Pada penelitian ini dijelaskan bagaimana cara menyelesaikan beberapa data dengan menggunakan metode algoritma CIH dengan penyimpanan data dengan menggunakan basis data, dan juga menyelesaikan perhitungan tentang waktu proses yang dibutuhkan dalam suatu penyelesaian berdasarkan jumlah kota yang berbeda-beda.

Hasil dari penelitian ini adalah basis data dapat digunakan sebagai alat bantu dalam mengimplementasikan penyelesaian permasalahan TSP yang berperan sebagai penyimpanan data proses sehingga pengambilan informasi jarak minimal dari beberapa alternatif yang ada dapat dilakukan dengan mudah yakni dengan menggunakan query. Jumlah kota juga sangat mempengaruhi waktu proses dalam penyelesaian, karena semakin banyak kota yang diinputkan maka waktu penyelesaian program pun akan semakin membutuhkan waktu yang relatif lebih lama.

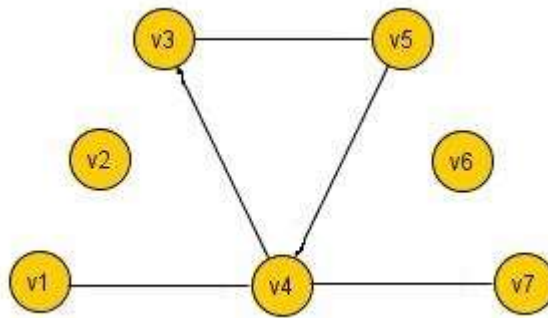
Kedua penelitian di atas menggambarkan hasil penyelesaian TSP menggunakan algoritma CIH dan belum dilanjutkan dengan pengembangan aplikasinya. Dengan demikian rancang bangun aplikasi penyelesaian TSP menggunakan algoritma CIH untuk optimasi rute pendistribusian pengisian ATM perlu dilakukan.

2.2 Graf

Sebuah graf G merupakan pasangan himpunan $(V(G), E(G))$, dimana $V(G)$ adalah himpunan berhingga tak kosong dari elemen yang disebut titik, dan $E(G)$ adalah sebuah himpunan (mungkin kosong) dari pasangan tak terurut $\{u,v\}$ dari titik-titik $u, v \in (G)$ yang disebut sisi (Slamin, 2009).

Graf G mungkin mengandung loop, yaitu sisi yang berbentuk $\{u,v\}$, dan sisi ganda yaitu sisi yang menghubungkan sepasang titik yang sama lebih dari satu. Untuk menyederhanakan notasi, sebuah sisi $\{u,v\}$ sering dinotasikan dengan uv (Slamin, 2009).

Titik u dan v dimisalkan merupakan titik-titik dari graf G . u dikatakan bertetangga dengan v jika terdapat sebuah sisi e yang menghubungkan u dan v , yaitu $e = uv$. Selanjutnya kita sebut v tetangga dari u . Himpunan semua tetangga dari u disebut ketetanggan dari u dan dinotasikan dengan $N(u)$. Kedua titik u dan v dapat juga disebut bersisian dengan sisi e (Slamin, 2009).



Gambar 2.1 Contoh Sebuah Graf

(Sumber : Slamin, 2009)

Gambar 2.1 menunjukkan sebuah contoh graf yang berordo 7 dengan himpunan titik $\{v1, v2, v3, v4, v5, v6, v7\}$, dan memiliki himpunan sisi $\{v1v4, v4v3, v3v5, v5v4, v4v7\}$ (Slamin, 2009).

Derajat dari titik v pada G adalah banyaknya titik-titik yang bertetangga dengan v , yaitu jumlah semua tetangga dari v . Dapat dilihat pada gambar 1, titik $v1$ bertetangga dengan titik $v4$, dan titik $v3$ bersisian dengan sisi $v3v4$ dan $v3v5$ (Slamin, 2009).

2.3 *Travelling Salesman Problem*

Travelling Salesman Problem adalah suatu permasalahan mencari sebuah rute tertutup untuk mengunjungi sejumlah kota, dimana setiap kota hanya dikunjungi sekali dan kembali ke kota awal setelah semua kota dikunjungi. TSP dibedakan menjadi 2 jenis yaitu *symetric TSP* dan *asymetric TSP* (Jonhson & Mc Geoch, 1995).

TSP dinyatakan sebagai permasalahan dalam mencari jarak minimal sebuah perjalanan yang berangkat dari sebuah titik awal dan kembali ke titik awal serta mengunjungi setiap titik hanya sekali. *TSP* direpresentasikan dengan menggunakan sebuah graf lengkap dan berbobot $G = (V, E)$ dengan V adalah himpunan titik yang merepresentasikan kota-kota yang akan dikunjungi dan E adalah himpunan sisi yang merepresentasikan jalan antar titik serta sisi $e = (v_i, v_j)$ memiliki bobot $c(e) = c_{ij}$ yaitu bobot sisi antara titik v_i dan v_j yang menunjukkan jarak antara titik i dan titik

j. Diberikan juga $C = (c_{ij})$ sebagai matriks bobot yang bersesuaian dengan E . Sehingga, permasalahan *TSP* adalah bagaimana menemukan siklus Hamilton dalam graf G dengan bobot minimum (Apriliani, 2011).

TSP dibagi menjadi 2 jenis, yakni :

1. TSP asimetris yakni dimisalkan, biaya dari kota 1 ke kota 2 tidak sama dengan biaya dari kota 2 ke kota 1. Dengan n kota, besarnya ruang pencarian adalah $\frac{n!}{n} = (n-1)!$ jalur yang mungkin.
2. TSP simetris yakni dimisalkan, biaya dari kota 1 ke kota 2 adalah sama dengan biaya dari kota 2 ke kota 1. Apabila dengan n kota, jumlah jalur yang mungkin adalah $\frac{n!}{2n} = \frac{(n-1)!}{2}$ jalur yang mungkin (Wiyanti, 2013).

2.4. Distribusi

Distribusi adalah sekelompok pedagang dan agen perusahaan yang mengkombinasikan antara pemindahan fisik dan nama dari suatu produk untuk menciptakan kegunaan bagi pasar tertentu (Dharmmesta, 1999). Sehingga dapat disimpulkan bahwa saluran distribusi adalah serangkaian organisasi yang saling tergantung dalam proses untuk membuat produk tersedia di pasar dan menciptakan kegunaan bagi konsumen sasaran.

Anggota saluran distribusi melaksanakan sejumlah fungsi yakni aktifitas-aktifitas yang dilaksanakan anggota saluran distribusi dalam memindahkan barang dari produsen ke konsumen dan menciptakan kegunaan produk tersebut bagi konsumen (Kotler, 2002).

2.5. Anjungan Tunai Mandiri

ATM atau dalam bahasa Indonesia juga disebut dengan Anjungan Tunai Mandiri adalah sebuah alat elektronik yang mengizinkan nasabah bank untuk mengambil uang dan mengecek rekening tabungan tanpa perlu dilayani oleh seorang *teller* manusia. Banyak ATM juga mengizinkan penyimpanan uang atau cek, transfer uang. ATM sering ditempatkan di lokasi-lokasi strategis seperti

restoran, pusat perbelanjaan, bandar udara, pasar, dan kantor-kantor bank itu sendiri (Anwar, 2015).

Jenis transaksi yang tersedia antara lain adalah :

1. Setoran Tunai
2. Penarikan Tunai
3. Transfer Dana
4. Pembelian
5. Pembayaran
6. Informasi Saldo
7. Dan Informasi Kurs

2.6. Algoritma

Algoritma adalah sebuah himpunan terhingga dari instruksi yang mempunyai karakteristik berikut ini:

1. Presisi (*precision*), langkah-langkahnya dinyatakan dengan jelas.
2. Unik (*uniqueness*), hasil lanjutan dari setiap langkah dari pelaksanaan didefinisikan secara tunggal dan semata-mata bergantung pada masukan dan hasil dari langkah sebelumnya.
3. Terhingga (*finiteness*), yaitu algoritma berhenti setelah beberapa intruksi terhingga dilaksanakan.
4. Masukan (*input*), yaitu algoritma memerlukan masukan.
5. Keluaran (*output*), yaitu algoritma menghasilkan keluaran.
6. Umum (*generality*), algoritma berlaku pada himpunan masukan.

Algoritma juga diartikan sebagai metode langkah demi langkah dari pemecahan suatu masalah. Langkah-langkah dari suatu algoritma harus dinyatakan dengan jelas sehingga dapat ditulis dalam bahasa pemrograman dan dijalankan oleh komputer (Johnsonbaugh, 1998).

Kompleksitas dari suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma tersebut untuk menyelesaikan masalah. Secara informal, algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki kompleksitas yang rendah, sementara algoritma yang

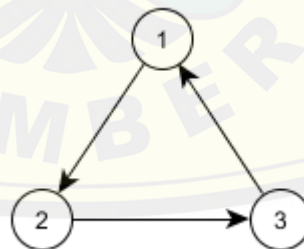
membutuhkan waktu lama untuk menyelesaikan suatu masalah membutuhkan kompleksitas yang tinggi.

2.7. *Cheapest Insertion Heuristic*

Algoritma ini memiliki konsep menyisipkan kota yang belum terlewati dengan tambahan jarak minimum hingga semua kota terlewati untuk mendapatkan solusi permasalahan (Rothlauf, 2011). Sebuah perjalanan (tour) Algoritma CIH dimulai dari simpul awal (1) menuju ke semua simpul (2,3,...,n) dan kembali ke simpul awal (1) tanpa ada simpul yang dikunjungi lebih dari satu kali dengan memperhitungkan tambahan jarak minimum ketika satu simpul disisipkan ke dalam partial tour yang ada.

Berikut ini adalah tata urutan algoritma CIH (Kusrini & Istiyanto, 2007).

1. Permasalahan dapat direpresentasikan ke suatu graf lengkap.
2. Penelusuran di mulai dari sebuah kota pertama yang dihubungkan dengan sebuah kota terakhir.
3. Hubungkan 2 kota tersebut dengan sebuah subtour. Subtour adalah perjalanan dari kota pertama dan berakhir di kota pertama sehingga membentuk sebuah siklus, misal (1,2)→(2,3) →(3,1) yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 Contoh *Subtour*

(Sumber : Hasil Analisis, 2015)

4. Substitusi salah satu arah hubungan (arc) dari dua kota dengan kombinasi dua arc, yaitu arc(i,j) dengan arc(i,k) dan arc(k,j), dengan k merupakan kota sisipan dengan tambahan jarak terkecil, yang diperoleh dari:

$$C_{ik} + C_{kj} - C_{ij} \quad [2.1]$$

Dengan :

C_{ik} adalah jarak dari kota i ke kota k ,

C_{kj} adalah jarak dari kota k ke kota j dan

C_{ij} adalah jarak dari kota i ke kota j

5. Ulangi langkah 4 sampai seluruh kota masuk dalam subtour sehingga algoritma di hentikan.

2.8 *Google Maps*

Google Maps adalah layanan gratis yang sudah disediakan oleh *Google* yang dapat digunakan untuk mengimplementasikan Sistem Informasi Geografis berbasis *website*. Salah satu keuntungan dari penggunaan *Google Maps* adalah layanan yang *free* dan bisa dikembangkan sesuai dengan keinginan karena *Google* sendiri sudah menyediakan *library* bagi para pengembang yang ingin memanfaatkan layanan *Google Maps* tersebut.

Google Maps API merupakan perkembangan dari *Google Maps* yang digunakan untuk menampilkan *Google Maps* ke dalam *website* yang dapat menampilkan peta 2 dimensi, foto satelit, relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi serta foto satelit yang di atasnya tergambar pula jalan dan nama kota sehingga dapat memudahkan pengguna untuk melihat data dalam bentuk peta secara jelas (Masykur, 2014).

BAB 3. METODOLOGI PENELITIAN

Metodologi penelitian merupakan bagian yang ditunjukkan untuk mendapat gambaran tahapan yang sistematis, yang dilakukan untuk menganalisa data dan mengembangkan sistem pada penelitian ini. Tahapan yang dilakukan adalah menjelaskan tentang jenis penelitian, metode penelitian, tempat dan waktu penelitian untuk membangun Rancang Bangun Aplikasi Pendistribusian Pengisian Anjungan Tunai Mandiri yang dilakukan pada PT. BRI (Persero) Tbk Cabang Jember.

3.1. Jenis Penelitian

Penelitian ini menggunakan dua jenis penelitian, yaitu penelitian kualitatif dan penelitian kuantitatif. Jenis penelitian kualitatif digunakan karena penelitian ini menganalisa studi kasus pada Bank BRI yang merupakan cabang di Kabupaten Jember dan jenis penelitian kuantitatif digunakan karena dalam penelitian ini menerapkan serta megkaji teori yang sudah ada sebelumnya.

3.2. Metode Penelitian

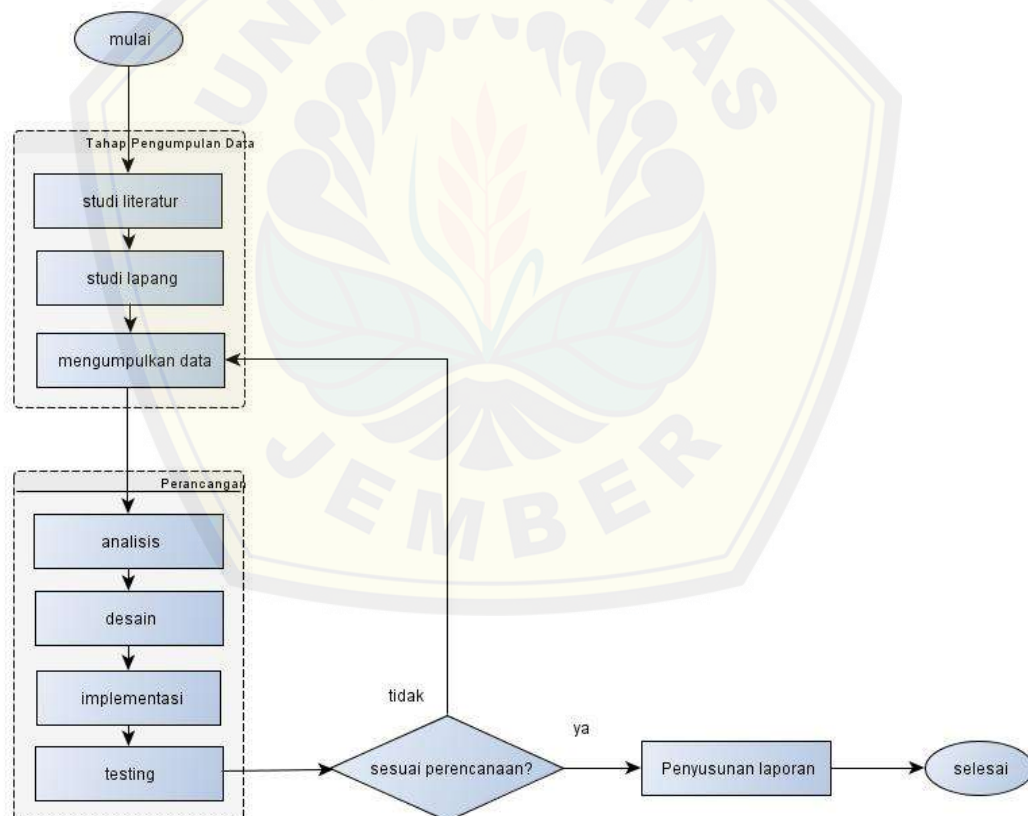
Metodologi penelitian merupakan bagian yang ditunjukkan untuk mendapat gambaran tahapan yang sistematis, yang dilakukan untuk menganalisa data dan mengembangkan sistem pada penelitian ini. Metode yang digunakan adalah penelitian studi kasus dimana melibatkan investigasi kasus, yang didefinisikan sebagai objek studi yang dibatasi untuk pnelitian dalam hal waktu, tempat, atau batas-batas fisik dengan menggunkan beberapa tahap pengumpulan data yakni wawancara, observasi lapangan, dan dokumentasi. Penelitian studi kasus digunakan untuk penelitian ini karena ditemukannya suatu proses kegiatan yang sangat sering dilakukan dan kasus tersebut akan diteliti untuk mendapatkan pemahaman yang lebih tentang kasus ini hingga dapat menyelesaikan permasalahan yang ada pada penelitian ini.

3.3. Tempat dan Waktu Penelitian

Penelitian dilakukan di PT. Bank Rakyat Indonesia (Persero) Tbk Cabang Jember yang merupakan Bank cabang di Kabupaten Jember. Waktu penelitian dilakukan selama 5 (lima) bulan, dimulai pada bulan Februari 2015 sampai dengan bulan Juni 2015.

3.4. Tahapan Penelitian

Tahapan yang digunakan dalam penelitian ini adalah tahap indentifikasi masalah, tahap pengumpulan data, tahap perancangan yakni tahap analisis dan pengembangan sistem dan tahapan penyusunan laporan. Gambaran tahapan penelitian dapat dilihat pada diagram alir pada Gambar 3.1.



Gambar 3.1 Diagram Alir Tahapan Penelitian

Langkah awal yang dilakukan dalam penelitian ini adalah tahapan pengumpulan data dimana terdapat studi literatur yakni peneliti melakukan

pencarian data dengan mencari referensi dan studi lapang yang merupakan proses dimana peneliti mendapatkan data dengan melakukan wawancara langsung pada narasumber sesuai dengan bidang yang terkait dengan permasalahan yang ada. Tahapan kedua adalah tahap perancangan dimana diawali dengan analisis data-data dan permasalahan secara keseluruhan untuk mendapatkan metode yang tepat agar dapat digunakan oleh sistem, yang dilanjutkan dengan desain, implementasi dan ketika semua telah selesai dilakukan akan dilakukan testing atau pengecekan untuk memeriksa apakah sistem yang dibuat telah sesuai dengan tahapan pada tahapan sebelumnya. Tahapan yang terakhir adalah tahap penyusunan laporan dimana ketika semua tahapan sebelumnya telah berhasil di selesaikan. Tahapan-tahapan penelitian akan dijelaskan lebih rinci sebagai berikut.

3.4.1. Tahap Pengumpulan Data

Tahapan pengumpulan data adalah tahapan dimana peneliti mencari untuk mendapatkan data-data dan informasi yang dibutuhkan untuk menyelesaikan penelitian. Pengumpulan data memiliki beberapa teknik yakni:

1. Studi Literatur

Studi Literatur dalam sebuah penelitian digunakan untuk mendapatkan referensi gambaran yang dapat membantu dalam menambah pemahaman dan wawasan yang sangat diperlukan dalam pengerjaan penelitian ini. Studi literatur bisa didapatkan melalui jurnal, buku, dan bahkan skripsi dari penelitian terdahulu, sehingga dengan demikian dapat membantu peneliti untuk mendapatkan gambaran tentang apa yang sudah dilakukan oleh peneliti sebelumnya dan dapat menghindari kesalahan yang ada dari suatu penelitian oleh peneliti sebelumnya. Bagian studi literatur yang digunakan untuk penyelesaian penelitian ini adalah berbagai jurnal dan penelitian sebelumnya tentang penggunaan travelling salesman problem dalam kehidupan sehari-hari, algoritma *CIH* yang digunakan untuk optimasi rute dengan berbagai macam studi kasus yang ada, dan berbagai macam lainnya.

2. Studi Lapang

Studi lapang merupakan teknik pengumpulan data secara langsung ke lapangan atau melihat langsung kondisi objek. Studi lapang dapat dilakukan dengan berbagai teknik, yakni diantaranya teknik observasi dengan mengadakan pengamatan terhadap objek yang diteliti yakni mengamati studi kasus yang terdapat pada PT. BRI (Persero) Tbk Cabang Jember dan teknik wawancara yang merupakan teknik dimana dilakukan dengan mengadakan aktivitas tanya jawab secara langsung kepada narasumber yakni Pelaksana IT dan *E-Channel* PT. BRI (Persero) Tbk Cabang Jember.

Data yang dibutuhkan untuk menyelesaikan penelitian ini berdasarkan hasil dari teknik penelitian berupa data primer dan data sekunder:

1. Data Primer

Data primer yang didapatkan dari observasi dan wawancara oleh narasumber PT. BRI (Persero) Tbk Cabang Jember adalah data lokasi ATM BRI di Kabupaten Jember dan penjelasan tentang langkah pendistribusian yang ada dan dipergunakan saat ini.

2. Data Sekunder

Data sekunder diperoleh dari studi literatur berupa buku, jurnal, referensi yang didapatkan dari internet, penelitian sebelumnya, dan data-data yang berhubungan dengan *Travelling Salesman Problem*, *Algoritma Cheapest Insertion Heuristic*, *Google Maps* dan studi kasus penyelesaian untuk permasalahan optimasi rute.

Data hasil dari studi literatur dan studi lapang yang telah dikumpulkan adalah data Anjungan Tunai Mandiri dan lokasi Anjungan Tunai Mandiri yang ada di Kabupaten Jember, yang akan dijelaskan di bawah ini.

3.4.1.1 Data Anjungan Tunai Mandiri

Data Anjungan Tunai Mandiri merupakan keterangan dari ATM-ATM yang ada di seluruh Kabupaten Jember, meliputi ID ATM, nama ATM, posisi ATM yakni alamat dari seluruh ATM. Data ini merupakan inti dari penggunaan sistem,

yakni untuk mencari jarak masing-masing ATM. Bank BRI memiliki 113 ATM yang terdapat di seluruh Kabupaten Jember, yang dapat dilihat pada Tabel 3.1.

3.4.1.2 Data Lokasi Anjungan Tunai Mandiri

Data lokasi Anjungan Tunai Mandiri merupakan data yang didapatkan dari proses pencarian jarak antara dua ATM. Data ini nantinya akan digunakan untuk perhitungan untuk mendapatkan jarak minimum dari seluruh ATM yang diinputkan untuk melakukan proses pendistribusian pengisian ATM. Data ATM BRI dapat dilihat contohnya pada Tabel 3.1.

Tabel 3.1 Data ATM BRI

No	Lokasi	Longitude
1.	ATM BRI 360048 CDM KANTOR CABANG BRI JEMBER	Jl. A Yani No 1 Jember
591.	ATM BANK BRI MAYANG	Jl. Raya Raung No. 1, Mayang, Jember
592.	ATM BRI KARANGANYAR	Ruko Graha Wijaya No.21, Jl. Darmawangsa, Jubung, Jember
593.	ATM BRI PATRANG	Jl Srikoyo Patrang Jember
594.	ATM BANK BRI KESILIR	Jl. Ambulu dpn Kantor Desa Kesilir Wuluhan Jember

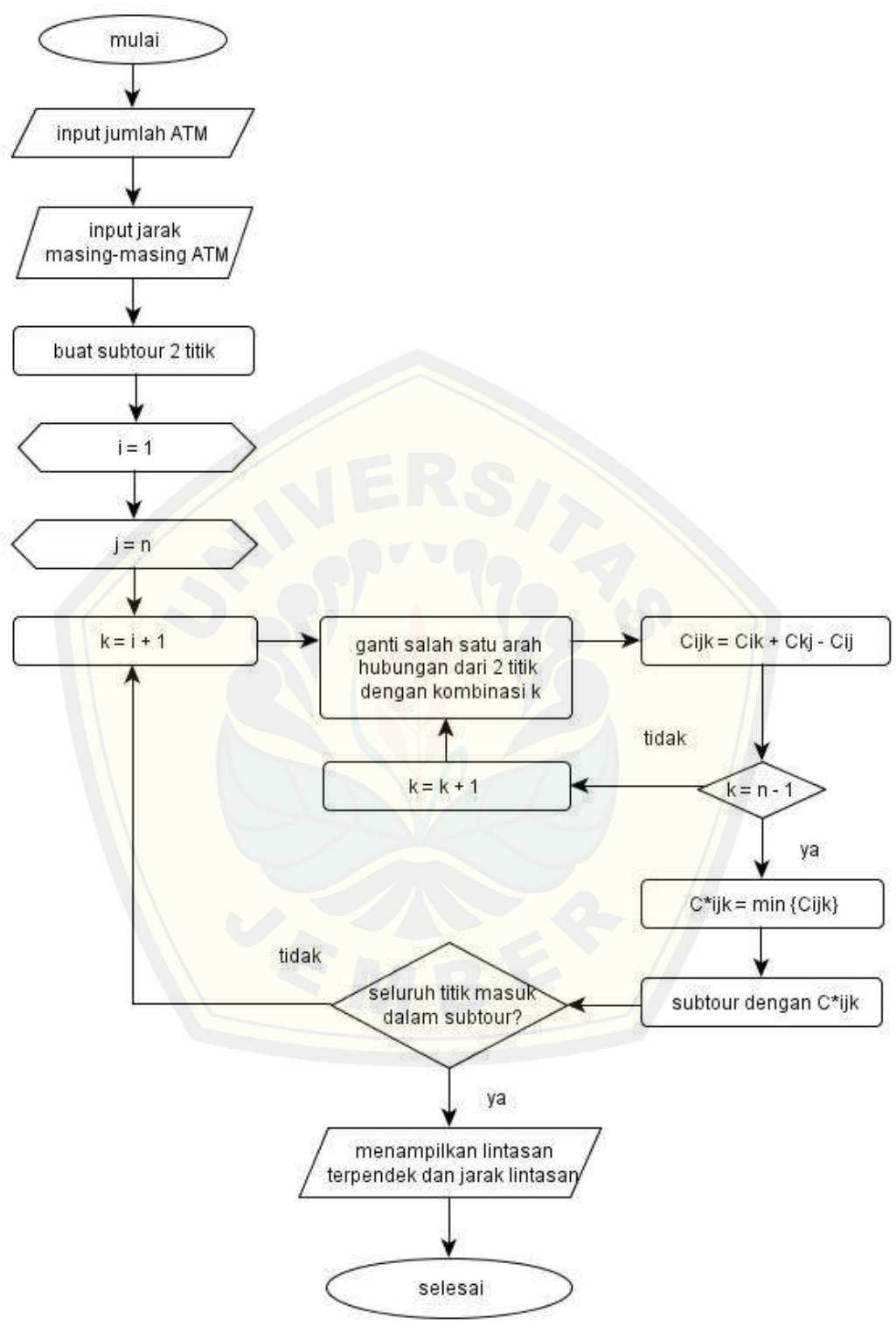
3.4.2. Tahap Analisis

Tahapan analisis merupakan tahapan dimana peneliti harus melakukan pemahaman data dengan baik sesuai dengan data yang telah diperoleh dari tahapan pengumpulan data. Tahapan selanjutnya adalah tahapan untuk menganalisa data yang telah didapatkan dengan metode *CIH*. Tahapan proses analisis data dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Blok Tahapan Analisis

Optimasi rute pendistribusian pengisian ATM memerlukan data-data diantaranya data lokasi ATM yang ada pada seluruh Kabupaten Jember, data jarak antara lokasi ATM yang satu dengan ATM yang lain yang digunakan sebagai data input untuk melakukan tahapan perhitungan proses menggunakan Algoritma Cheapest Insertion Heuristic. Output yang didapatkan dari sistem ini adalah optimasi rute yang digunakan untuk pendistribusian pengisian ATM, dengan detail proses yang dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Alir Algoritma CIH

Tahap analisis data adalah mengelompokkan data yang sudah dikumpulkan ke dalam dua jenis yaitu data *spatial* dan data atribut. Tahap ini mengelompokkan data dengan dua cara yakni dengan mendapatkan koordinat lokasi data dan memprosesnya melalui proses digitasi yang akan dijelaskan pada sub-subbab di bawah ini.

3.4.2.1 Penentuan Koordinat Lokasi

Penentuan koordinat lokasi adalah cara untuk mendapatkan titik hasil perpotongan antara garis lintang dan garis bujur yang menunjukkan suatu objek, dalam penelitian ini adalah data ATM BRI seluruh Kabupaten Jember dengan menggunakan *software Google Earth*.

Tahapan pada *Google Earth* adalah mencari koordinat lokasi data yang selanjutnya diberikan tanda titik, lalu *Google Earth* otomatis akan menampilkan koordinat lokasi titik yang merupakan hasil dari pengambilan koordinat oleh citra satelit. Hasil dengan format kml harus dikonversi menjadi format shp sehingga data dapat dimasukkan ke dalam database, dengan demikian maka digunakan *software Quantum GIS* untuk mengkonversi data tersebut.

3.4.2.2 Digitasi

Digitasi adalah proses pengonversian data spasial pada peta ke dalam format digital. Hasil data yang telah dikonversi selanjutnya dikonversi lagi menggunakan Quantum GIS menggunakan salah satu *software postgis* sehingga data dapat disimpan dalam *database*.

3.4.2.3 Data Hasil Digitasi

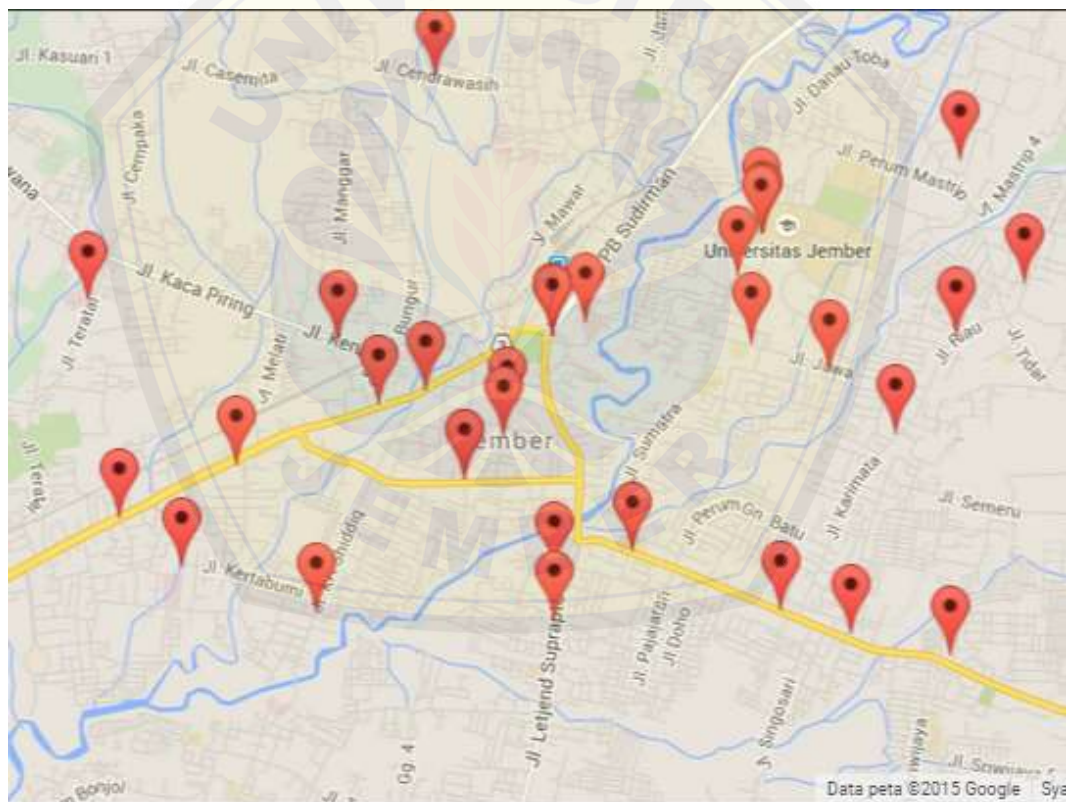
Data hasil digitasi terdiri dari dua jenis yakni data *spatial* dan data atribut. Data yang digunakan dalam penelitian pengembangan aplikasi pendistribusian pengisian Anjungan Tunai Mandiri pada PT. BRI (Persero) Tbk Cabang Jember adalah data atribut yang merupakan data yang menggambarkan aspek deskripsi dan menjelaskan tentang fenomena dalam bentuk kata-kata, angka atau tabel. Data

atribut yang diperlukan dalam penelitian ini adalah data ATM BRI di Kabupaten Jember.

3.4.2.4 Analisis Data ke Dalam Metode Sistem

Data-data yang telah didapat selanjutnya akan dianalisis ke dalam perhitungan metode *CIH* untuk mengetahui apakah metode tersebut dapat digunakan dalam proses optimasi rute pendistribusian pengisian ATM. Analisis data ke dalam metode *CIH* akan dijelaskan sebagai berikut.

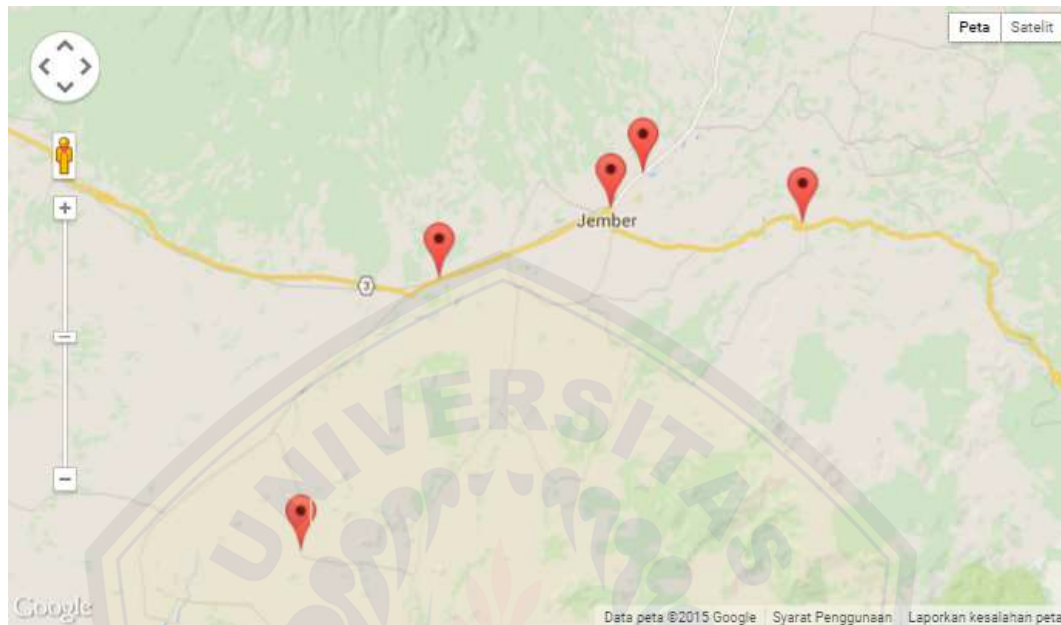
Tahap pertama adalah menggambarkan dengan jelas beberapa sampel ATM yang akan masuk ke dalam perhitungan optimasi rute untuk aplikasi pendistribusian pengisian ATM pada *google maps*, yang dapat dilihat pada gambar Gambar 3.4.



Gambar 3.4 Peta ATM BRI di Kabupaten Jember

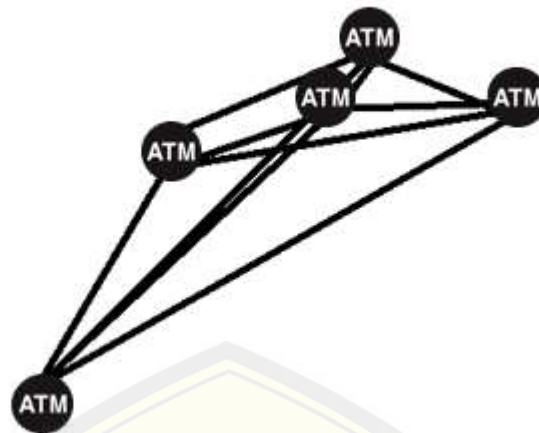
Tahapan selanjutnya adalah dengan memisalkan 5 ATM yang akan digunakan dalam melakukan perhitungan dengan menggunakan algoritma *CIH*, yang dimisalkan adalah pertama ATM BRI CDM Kantor Cabang BRI Jember,

kedua ATM BRI Mayang, ketiga ATM BRI Karanganyar, keempat ATM BRI Patrang dan kelima ATM BRI Kesilir. Penggambaran tahapan ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Sampel ATM BRI

Sampel beberapa ATM tersebut selanjutnya akan direpresentasikan dengan mengubah menjadi sebuah *graf*, dengan mengubah titik lokasi ATM menjadi *node* (titik) dan jalan yang dilewati menjadi *edge* (garis). Tahapan ini akan diaplikasikan untuk memecahkan permasalahan tentang optimasi rute dengan menggunakan metode *CIH*, yang dapat dilihat pada Gambar 3.6.



Gambar 3.6 Representasi Peta ATM BRI dengan Graf

Perhitungan pertama yang dilakukan adalah penentuan tabel jarak yakni tabel yang digunakan untuk menentukan jarak antara 2 ATM yang dituju yang ditulis dengan asal dan tujuan, dengan nomor sesuai urutan ATM yang telah diinisialisasi sebelumnya, seperti yang terlihat pada Tabel 3.2.

Tabel 3.2 Perhitungan Jarak

No.	Asal	Tujuan	Jarak
1	1	2	12,5
2	1	3	11
3	1	4	3,3
4	1	5	30,6
5	2	1	14,1
6	2	3	22,8
7	2	4	15
8	2	5	41,7
9	3	1	10,2
10	3	2	22,7
11	3	4	12,8

12	3	5	19,6
13	4	1	2,7
14	4	2	14,2
15	4	3	13,8
16	4	5	33,3
17	5	1	29,8
18	5	2	40,4
19	5	3	19,6
20	5	4	32,4

Pada tabel hasil 1 menjelaskan hasil yang ditulis untuk pertama kali, yakni asal adalah atm pertama dan tujuan adalah jumlah atm yang telah diinputkan untuk perhitungan yang akan dilakukan, seperti tabel Tabel 3.3.

Tabel 3.3 Hasil 1

No.	Asal	Tujuan	Nomor
1	1	5	1
2	5	1	2

Subtour adalah perjalanan dari titik pertama dan berakhir di titik pertama sehingga membentuk sebuah siklus, sehingga dapat dilihat pada tabel hasil pertama membuat sebuah *subtour* awal yakni (1) – (5). Tahap tabel hasil pertama kali selanjutnya diimplementasikan pada tabel proses dengan melakukan pengulangan untuk menyisipkan data ATM yang belum diinputkan, selanjutnya melakukan pengulangan untuk mengganti asal dan tujuan. Proses pencarian jarak pada tabel proses dengan menggunakan rumus di bawah ini dan perhitungan dapat dilihat pada Tabel 3.4.

$$C_{ik} + C_{kj} - C_{ij} \tag{3.1}$$

dimana

C_{ik} : jarak antara asal dan sisip

C_{kj} : jarak antara sisip dan tujuan

C_{ij} : jarak antara asal dan tujuan.

Tabel 3.4 Proses 1

No.	Asal	Sisip	Tujuan	Jarak
1	1	2	5	23,6
2	1	3	5	0
3	1	4	5	6
4	5	2	1	14,1
5	5	3	1	0
6	5	4	1	5,3

Hasil yang didapat dari proses perhitungan pada tabel proses 1 menghasilkan total jarak terkecil yakni 0 km, sehingga subtour lama terganti dengan subtour yang baru, yakni (1) – (3) – (5). Perhitungan selanjutnya adalah memasukan hasil yang ada pada tabel proses pada tabel hasil dengan menginputkan *subtour* baru dan memberikan nomer, yang dapat dilihat pada Tabel 3.5.

Tabel 3.5 Hasil 2

No.	Asal	Tujuan	Nomor
1	1	3	1
2	3	5	2
3	5	1	3

Tahapan selanjutnya adalah melakukan proses perhitungan untuk mencari jarak dengan rumus yang sama, dan memasukan titik yang belum masuk yakni titik 2 dan 4 ke dalam subtour yang baru yang dapat dilihat pada Tabel 3.6.

Tabel 3.6 Proses 2

No.	Asal	Sisip	Tujuan	Jarak
1	1	2	3	24,3
2	1	4	3	6,1
3	3	2	5	44,8
4	3	4	5	26,5
5	5	2	1	24,7

6	5	4	1	5,3
---	---	---	---	-----

Pencarian jarak minimum pada Tabel Proses 2 menghasilkan jarak 5,3 km dengan titik asal adalah 4, titik sisip adalah 4 dan titik tujuan adalah 1. Subtour selanjutnya yang terbentuk untuk menggantikan subtour sebelumnya adalah subtour (1) – (3) – (5) – (4). Subtour yang baru selanjutnya dimasukkan ke dalam tabel hasil dan diberikan nomer sesuai urutan, seperti yang terlihat pada Tabel 3.7.

Tabel 3.7 Hasil 3

No.	Asal	Tujuan	Nomor
1	1	3	1
2	3	5	2
3	5	4	3
4	4	1	4

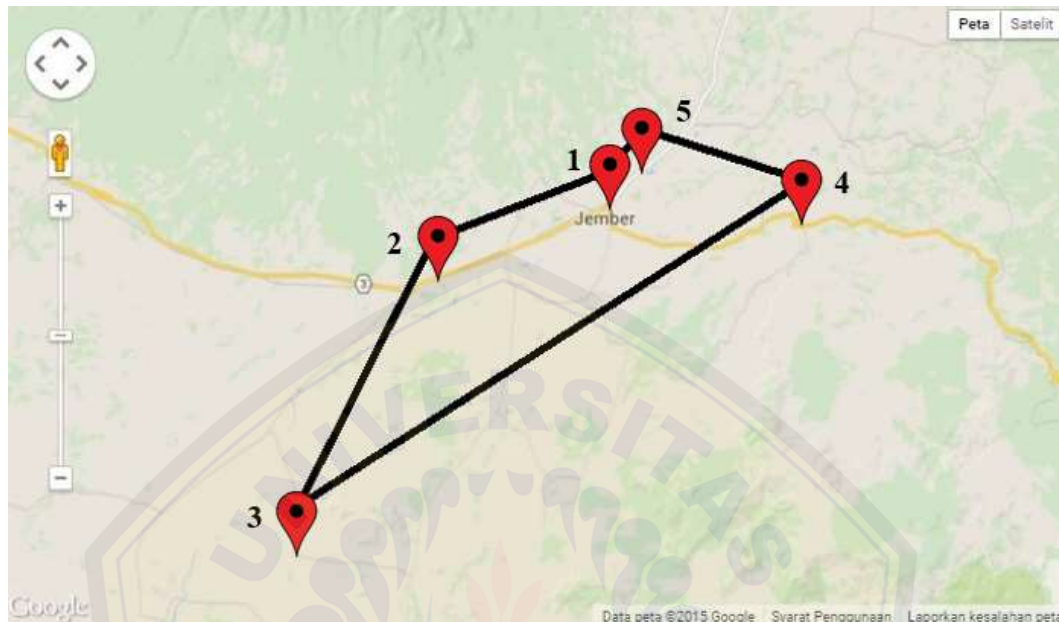
Titik yang belum masuk adalah titik 2, sehingga titik yang tersisa yakni titik 2 di masukan dalam Tabel Proses 3 ke dalam titik-titik subtour yang baru dan melakukan proses pencarian jarak terkecil dan ditemukan jarak terkecil yakni 23 km seperti yang terlihat pada Tabel 3.8.

Tabel 3.8 Proses 3

No.	Asal	Sisip	Tujuan	Jarak
1	1	2	3	24,3
2	3	2	5	44,8
3	5	2	4	23
4	4	2	1	25,6

Hasil yang didapatkan dari Tabel Proses 3 adalah hasil perhitungan dari proses optimasi rute pendistribusian pengisian ATM dengan *subtour* atau rute tujuan yakni (1) – (3) – (5) – (2) – (4) dengan total jarak tempuh perjalanan adalah total jarak yang dihitung dari jarak 2 atm, yakni jarak antara titik (1) – (3) adalah 11 km, jarak antara titik (3) – (5) adalah 19,6 km, jarak antara titik (5) – (2) adalah

40,4 km dan jarak antara titik (2) – (4) adalah 15 km, sehingga dapat dihitung bahwa dengan subtour yang telah ditemukan didapatkan total jarak keseluruhan yakni 86 km.



Gambar 3.7 Peta ATM BRI dengan metode *CIH*

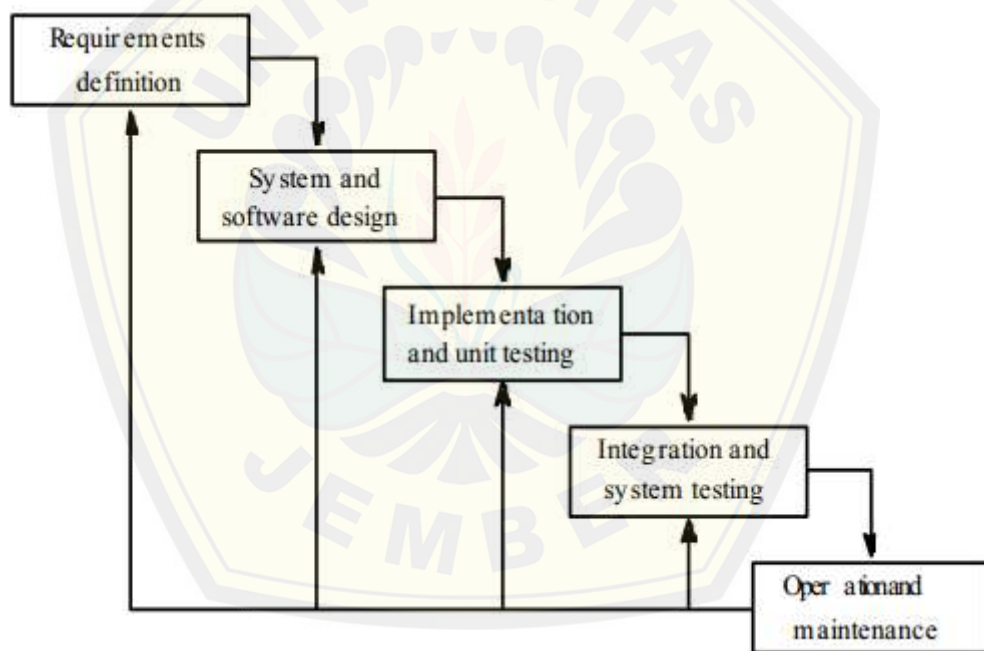
Gambar 3.7 menggambarkan optimasi rute dengan menggunakan algoritma *CIH* dengan rute yang telah dihitung menghasilkan tujuan pertama yakni ATM BRI CDM Kantor Cabang Jember menuju lokasi kedua yakni ATM BRI Karanganyar menuju ATM BRI Kesilir, ATM BRI Mayang dan lokasi terakhir yakni ATM BRI Patrang, dengan total jarak 86 km. Perhitungan di atas membuktikan bahwa proses perhitungan untuk optimasi rute pada pendistribusian pengisian ATM dapat dilakukan dengan menggunakan algoritma Cheapest Insertion Heuristic.

3.4.3. Tahap Pengembangan Sistem

Tahapan selanjutnya adalah tahapan pengembangan sistem yang digunakan untuk membangun dan merancang sistem dan dilakukan ketika tahapan sebelumnya telah selesai dilakukan. Model pengembangan sistem untuk aplikasi pendistribusian Pengisian ATM menggunakan algoritma *CIH* adalah model *waterfall*. Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam

membangun *software*. Nama model ini sebenarnya adalah *Linear Sequential Model*. Model ini sering disebut dengan *classic life cycle* atau model *waterfall*. Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya secara berurutan (Pressman, 1997).

Waterfall adalah suatu metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematis dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, *design*, kode, pengujian dan pemeliharaan (Pressman, 1997). Tata urutan pengerjaan menggunakan model *waterfall* dapat dilihat pada Gambar 3.8.



Gambar 3.8 Tahapan Model *Waterfall*

(Sumber : Roger S. Pressman, 2002)

3.4.3.1. Analisis Kebutuhan

Perancangan perangkat lunak tahap pertama dimulai dari tahap analisis kebutuhan yaitu peneliti setelah melakukan pencarian data, yakni dengan cara wawancara, melakukan studi literatur yang terkait dengan penelitian, melakukan

studi dengan sistem yang telah ada akan menemukan permasalahan yang nantinya akan di analisis kebutuhan yang diperlukan untuk mendapatkan solusi yang baik dalam penyelesaian masalah tersebut.

Tahap ini juga dibutuhkan untuk mendapatkan data yang dibutuhkan dalam pembuatan sistem, seperti kebutuhan fungsional dan kebutuhan non fungsional dari sistem terkait yang akan dibangun.

Analisis kebutuhan yang dilakukan pada penelitian ini dengan wawancara kepada Bagian *IT & E-Channel* bank BRI cabang di Kabupaten Jember untuk mendapatkan data terkait dengan kebutuhan penelitian seperti data lokasi ATM-ATM yang terdapat pada lingkup kabupaten Jember, data jarak antar masing-masing ATM, data kendaraan, data petugas lapang, dan juga data profil dari bank.

3.4.3.2.Desain Sistem

Pembuatan desain sistem pada aplikasi penelitian ini menggunakan *Unified Modeling Language (UML)* yang dirancang dengan konsep *Object-Oriented Programming (OOP)*. Pemodelan *UML* yang digunakan adalah sebagai berikut:

1. *Business Process* menggambarkan sebuah proses yang didalamnya juga terdapat *trigger* yang mendorong sistem tersebut berjalan, *resource* dan *information* yang dibutuhkan dalam pembuatan sistem, *output* dan juga *goal* yang akan dihasilkan oleh sistem tersebut.
2. *Use Case Diagram* menggambarkan sebuah skenario yang menampilkan interaksi antara pengguna dan sistem. Oleh karena itu, *use case diagram* menunjukkan serangkaian tindakan serta aktor yang dapat melakukan tindakan tersebut.
3. Skenario menggambarkan fitur dan isi yang ada pada *usecase diagram* dengan lebih terperinci dengan menjelaskan alur sistem dan keadaan yang akan terjadi pada suatu *event* tertentu.
4. *Activity Diagram* menggambarkan aktifitas yang terjadi dalam sebuah sistem yang sesuai dengan scenario yang diimplementasikan dalam bentuk diagram alir.

5. *Sequence Diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem. *Sequence Diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu.
6. *Class Diagram* merupakan bangunan utama dalam pemodelan berorientasi objek. Diagram ini menggambarkan sebuah pandangan dari satu aspek tertentu dari model atau keseluruhan beserta hubungan mereka. *Class Diagram* terutama digunakan untuk membangun sebuah arsitektur sistem dengan menangkap dan mendefinisikan *class-class* dan *interface* dan hubungannya.
7. *Entity Relationship Diagram* merupakan suatu model yang digunakan untuk menjelaskan hubungan yang terdapat antara basis data berdasarkan objek-objek yang mempunyai hubungan atau relasi.

3.4.3.3. Implementasi

Implementasi adalah pembuatan kode program yang dibuat berdasarkan desain yang telah dibuat sebelumnya, sehingga pada tahap implementasi perlu dilakukan tahapan yakni penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)*, *Cascading Style Sheet (CSS)*, dan *Hyper Text Markup Language (HTML)*, dan manajemen basis data menggunakan *DBMS PostgreSQL*.

3.4.3.4. Pengujian

Tahap pengujian merupakan proses dimana untuk mengidentifikasi adanya sesuatu yang tidak sesuai pada aplikasi yang diharapkan. Pengujian sangat dibutuhkan untuk memastikan dan menjaga kualitas dari aplikasi yang kita buat, untuk melakukan evaluasi dari perbedaan antara fitur yang diinginkan pada tahap analisis dengan sistem yang telah terbentuk, dan untuk mengetahui apakah aplikasi yang kita buat telah memenuhi persyaratan yang ada. Pada aplikasi ini dilakukan dua tahap pengujian, yakni *white box testing* dan *black box testing*.

1. *Black box testing* adalah pengujian berdasarkan spesifikasi kebutuhan yang ada dan tidak memerlukan pemeriksaan kode yang ada. Pengujian ini dilakukan berdasarkan persyaratan-persyaratan yang diberikan, sehingga kesalahan-kesalahan yang ada dapat diidentifikasi dan dapat segera ditangani dengan mudah (Nidhra & Dondeti, 2012).
2. *White box testing* adalah pengujian pada fungsi internal dari sudut pandang pengembang yang berfokus pada logika internal dan struktur kode. Teknik ini menguji setiap cabang dan keputusan dalam program untuk menemukan kesalahan ketik acak, *debugging* kode dan mengungkapkan pemrograman yang salah asumsi (Nidhra & Dondeti, 2012). Tahapan teknik pengujian white box meliputi *listing* program, grafik alir, kompleksitas siklomatik, jalur program independen dan pengujian basis set.
 - a) *Listing Program*

Listing Program adalah langkah untuk memberikan nomor kepada kode-kode program baik berupa penggunaan kondisi atau menjalankan *statement* biasa yang akan digunakan untuk tahap pengujian selanjutnya.
 - b) *Grafik Alir*

Grafik alir merupakan penggambaran aliran kontrol yang didapatkan dari hasil penomoran dari *listing* program. Grafik alir digunakan menggambarkan aliran kontrol dengan *node-node* (simpul) yang dihubungkan dengan *edge* (garis) sesuai dengan jalannya program.
 - c) *Kompleksitas Siklomatik*

Kompleksitas siklomatik adalah pengukuran kompleksitas kode sumber yang berhubungan dengan sejumlah kesalahan yang terdapat pada kode program. Teknik pengujian ini dihitung dengan mengukur jumlah jalur *linear* melalui modul program.
 - d) *Jalur Program Independen*

Jalur program independen merupakan sebuah atau kumpulan pernyataan dari setiap jalur program yang memiliki kondisi baru, yang ketika dinyatakan dalam grafik alir maka jalur program independen

harus bergerak setidaknya sepanjang satu edge yang belum dilintasi sebelum jalur tersebut didefinisikan,

e) Pengujian Basis Set

Pengujian Basis Set adalah pengujian dengan memasukan data ke dalam grafik alir yang selanjutnya akan dieksekusi setidaknya satu kali untuk mengetahui bahwa sistem tersebut telah memenuhi syarat kelayakan perangkat lunak atau tidak.

3.4.3.5. Pemeliharaan

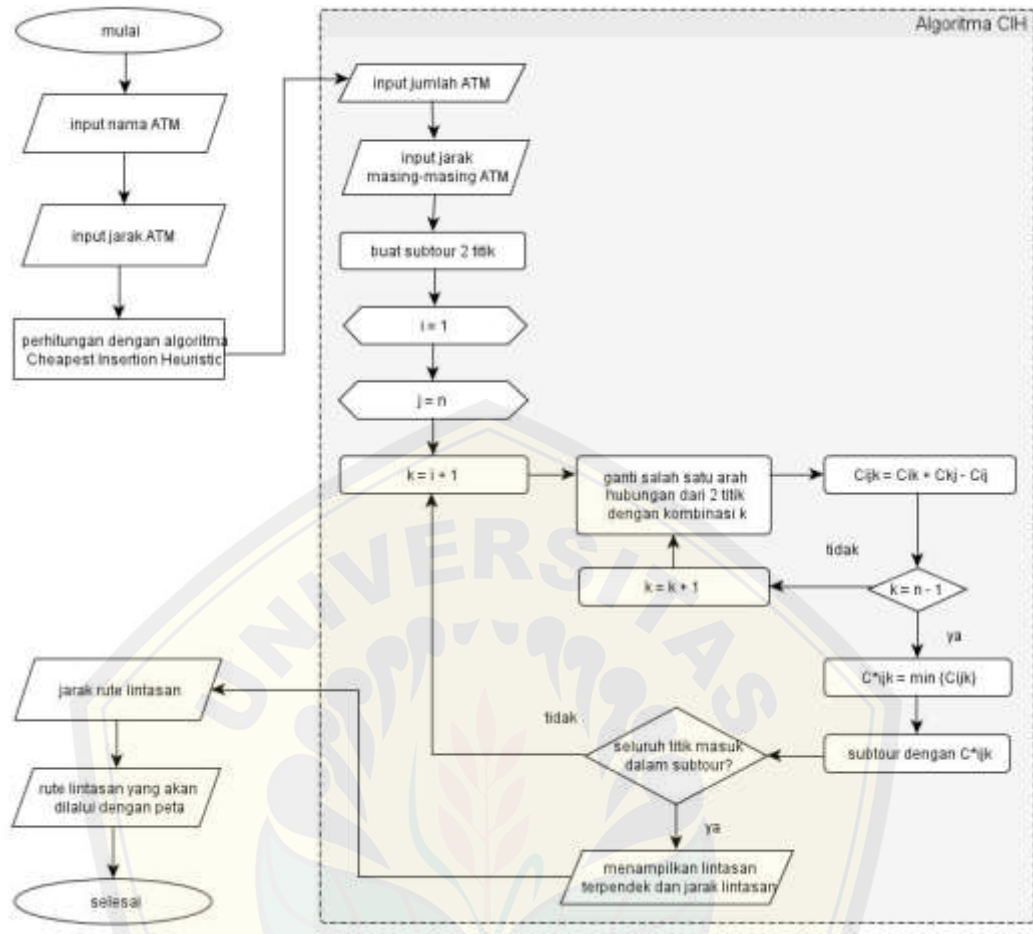
Tahapan pemeliharaan adalah tahapan yang dilakukan ketika sistem yang telah selesai diuji oleh PT. BRI (Persero) Tbk Cabang Jember tidak sesuai, seperti terdapat kesalahan atau *error* dan apabila sistem mengalami perubahan yang disebabkan oleh kebutuhan pengguna yang dapat dikarenakan penyesuaian dengan kebutuhan yang sebenarnya.

3.5. Gambaran Umum Sistem

Aplikasi yang akan dibangun adalah aplikasi pendistribusian pengisian ATM dengan menggunakan algoritma CIH yang akan digunakan untuk mengelola data-data yang diberikan oleh salah satu bank yang merupakan cabang di Kabupaten Jember yakni data lokasi ATM-ATM yang terdapat pada lingkup kabupaten Jember, data jarak antar masing-masing ATM, data kendaraan, data petugas lapang, dan juga data profil dari bank.

Algoritma CIH dalam hal ini digunakan untuk mengolah data masing-masing ATM dengan jarak, sehingga ketika petugas lapang mengantarkan pendistribusian uang untuk ATM memiliki rute yang telah ditentukan, berdasarkan jumlah ATM yang diinputkan dan akan menghasilkan jarak tempuh yang lebih optimal.

Gambar 3.9 yang merupakan skema dari alur penerapan aplikasi pendistribusian pengisian ATM. Parameter awal yang digunakan adalah menginputkan ATM-ATM yang terdapat pada lingkup kawasan Bank BRI di Kabupaten Jember, menginputkan jarak antar masing-masing ATM yang dilanjutkan dengan menerapkan pengerjaan menggunakan algoritma CIH.



Gambar 3.9 Flowchart sistem Pendistribusian Pengisian ATM

BAB 4. PENGEMBANGAN SISTEM

Analisis data dan pengembangan sistem merupakan bagian yang menggambarkan untuk menganalisis dan mengembangkan sistem pendistribusian pengisian anjungan tunai mandiri secara keseluruhan dengan menggunakan metode *CIH*. Pengembangan sistem dilakukan dengan menggunakan model *waterfall*, dengan tahapan yakni analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, pengujian sistem dan pemeliharaan.

4.1 *Statement Of Purpose*

Aplikasi pendistribusian pengisian ATM dengan menggunakan algoritma *Cheapest Insertion Heuristic (CIH)* yang akan digunakan untuk mengelola data-data yang diberikan oleh PT.BRI (Persero) Tbk Cabang Jember yakni data lokasi ATM-ATM yang terdapat pada lingkup Kabupaten Jember, data jarak antar masing-masing ATM, data kendaraan, data petugas lapang, dan juga data profil dari bank.

Algoritma *CIH* dalam hal ini digunakan untuk mengolah data masing-masing ATM dengan jarak, sehingga ketika petugas lapang mengantarkan pendistribusian uang untuk ATM memiliki rute yang telah ditentukan, berdasarkan jumlah gudang yang diinputkan dan akan menghasilkan jarak tempuh yang lebih optimal.

4.2 Analisis Kebutuhan Sistem

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

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional dari Aplikasi Pendistribusian Pengisian ATM antara lain:

1. Sistem dapat menyimpan dan memanejemen (*create,update,delete*) seluruh data ATM Bank BRI Kabupaten Jember.
2. Sistem dapat menghitung jarak antara seluruh ATM yang terdapat pada seluruh Kabupaten Jember.
3. Sistem dapat melihat seluruh ATM yang terdapat pada seluruh Kabupaten Jember.
4. Sistem dapat melakukan perhitungan untuk mengoptimalkan jarak tempuh untuk melakukan pengisian ATM yang akan di tuju.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non fungsional dari Aplikasi Pendistribusian Pengisian ATM antara lain:

1. Tampilan sistem menarik dan mudah di pahami oleh seluruh pengguna atau user.
2. Sangat mudah untuk di operasikan.

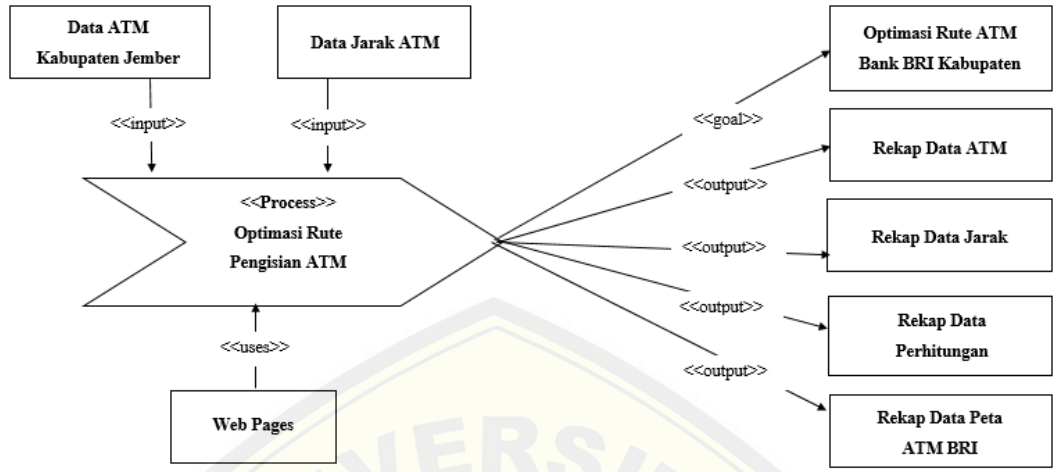
4.3 Desain Sistem

Desain sistem merupakan tahapan yang menggambarkan aplikasi pendistribusian pengisian ATM dengan tahapan yakni *bussines process, Usecase Diagram, Usecase Skenario, Activity Diagram, Sequence Diagram, Class Diagram* dan *Entity Relationship Diagram (ERD)*.

4.3.1 *Business Process*

Business Process merupakan sekumpulan proses yang dilakukan untuk mencapai hasil yang diinginkan. Terdapat beberapa komponen untuk mencapai tujuan tersebut, yaitu data yang menjadi masukan (*input*) dan media yang digunakan sehingga akan dihasilkan data keluraan (*output*) untuk melakukan tujuan

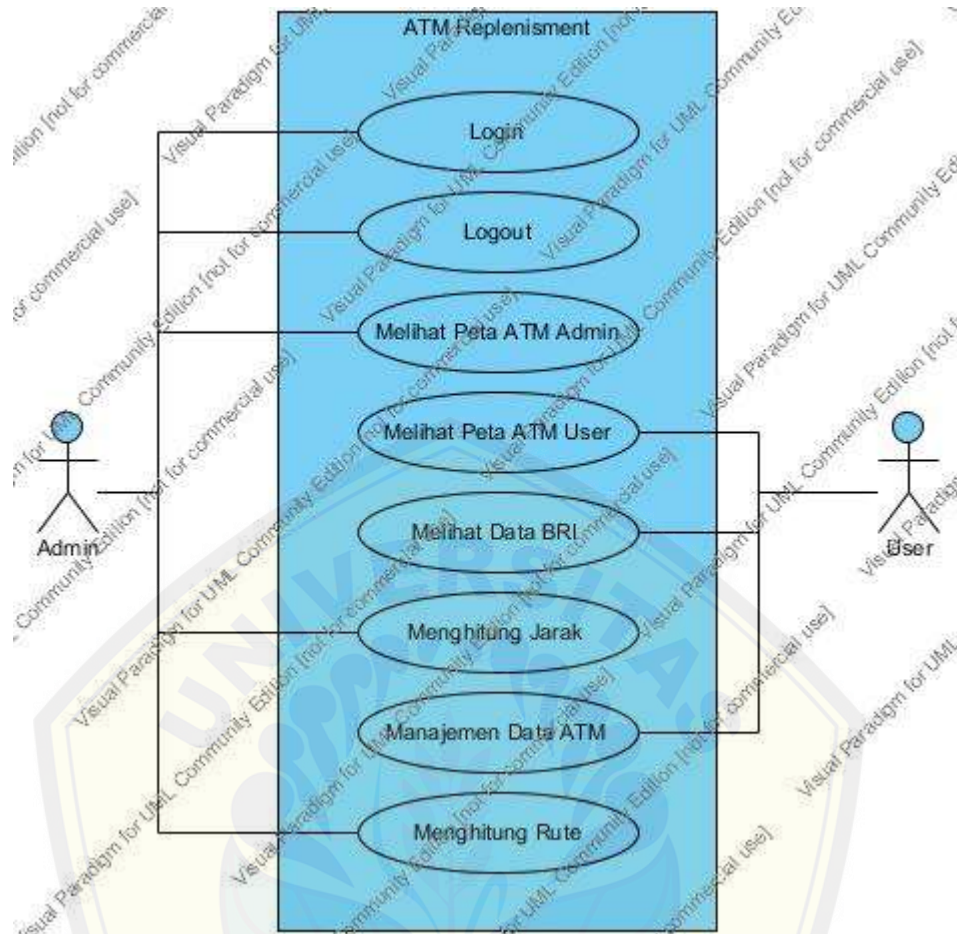
(goal) yang ingin dicapai. *Business Process* Pendistribusian Pengisian ATM dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Bussiness Process* Pendistribusian Pengisian ATM

4.3.2 Usecase Diagram

Usecase Diagram adalah dokumentasi untuk menggambarkan fitur dan aktor yang terdapat pada sistem yang akan dibuat. *Usecase* diagram Pendistribusian Pengisian Anjungan Tunai Mandiri dapat dilihat seperti yang dijelaskan pada Gambar 4.2.



Gambar 4.2 Usecase Diagram Pendistribusian Pengisian ATM

Penjelasan tentang definisi aktor dan definisi *usecase* dalam *usecase* diagram Pendistribusian Pengisian Anjungan Tunai Mandiri akan dijelaskan di bawah ini.

1. Definisi Aktor

Definisi aktor merupakan penjelasan tentang aktor – aktor sebagai pengguna dari Pendistribusian Pengisian Anjungan Tunai Mandiri yang akan dibagun. Terdapat 2 (dua) aktor dari hasil analisis seperti yang dijelaskan pada

Tabel 4.1.



Tabel 4.1 Definisi Aktor

No	Aktor	Deskripsi
1.	<i>Admin</i>	Aktor <i>admin</i> merupakan aktor yang memiliki hak akses untuk mengelola sistem pendistribusian pengisian ATM. Admin dapat untuk melakukan proses <i>login</i> , proses <i>logout</i> , proses melihat data ATM admin, mengelola jarak ATM, memajemen data ATM, dan proses perhitungan rute ATM. Aktor admin ini merupakan bagian <i>IT dan E-Channel</i> bank BRI Cabang Jember.
2.	<i>User</i>	Aktor <i>user</i> merupakan aktor yang hanya bisa melihat data ATM user, melihat data BRI, dan proses perhitungan rute. Aktor <i>user</i> ini merupakan petugas bagian pengisian dan dapat juga karyawan BRI lainnya.

2. Definisi *Usecase*

Definisi *Usecase* merupakan penjelasan dari masing – masing *usecase* atau fitur – fitur pada aplikasi pendistribusian pengisian Anjungan Tunai Mandiri. Penjelasan *usecase* dapat di lihat pada Tabel 4.2.

Tabel 4.2 Definisi *Usecase*

No	<i>Usecase</i>	Penjelasan
1	<i>Login</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>login</i> atau autentifikasi untuk masuk sistem.
2	<i>Logout</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>logout</i> atau autentifikasi untuk keluar dari sistem.

Berlanjut

Lanjutan

3	Melihat Peta ATM Admin	Merupakan <i>usecase</i> yang menggambarkan proses untuk melihat data ATM yang berada di Kabupaten Jember dengan bantuan peta untuk admin.
4	Melihat Peta ATM User	Merupakan <i>usecase</i> yang menggambarkan proses untuk melihat data ATM yang berada di Kabupaten Jember dengan bantuan peta untuk User.
5	Melihat data BRI	Merupakan <i>usecase</i> yang menggambarkan proses untuk dapat melihat sejarah dan visi & misi bank BRI.
6	Menghitung Jarak	Merupakan <i>usecase</i> yang menggambarkan proses untuk menginputkan jarak yang di dapat dari inputan ATM bank BRI.
7	Manajemen Data ATM	Merupakan <i>usecase</i> yang menggambarkan proses untuk menambah, mengedit dan menghapus data Bank BRI yang terdapat di Kabupaten Jember.
8	Menghitung Rute	Merupakan <i>usecase</i> yang menggambarkan proses untuk melakukan perhitungan hingga di temukan solusi pencarian untuk optimasi rute.

4.3.3 *Usecase* Skenario

Usecase skenario adalah dokumentasi terhadap kebutuhan fungsional sistem.

Usecase skenario Pendistribusian Pengisian ATM adalah sebagai berikut.

1. *Usecase* Skenario Menghitung Rute

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario Menghitung Rute, dapat dilihat pada tabel Tabel 4.3.

Tabel 4.3 *Usecase* Skenario Menghitung Rute

Nama	Menghitung Rute
<i>Trigger</i>	<i>User</i>
<i>Pre Condition</i>	<i>User</i> menekan menu “Perhitungan Rute”
<i>Post Condition</i>	<i>User</i> mengolah data pada ATM untuk mencari rute terbaik dan total jarak keseluruhan rute
SKENARIO NORMAL PERHITUNGAN RUTE	
Aktor	Sistem
1. Memilih menu “Perhitungan Rute”	
	2. Menampilkan halaman Perhitungan Rute
SKENARIO NORMAL INPUT JUMLAH ATM	
Aktor	Sistem
3. Menginputkan jumlah ATM pada inputan yang akan di routing	
4. Klik <i>button</i> “tambah”	
	5. Menampilkan <i>dropdown</i> ATM sebanyak jumlah yang telah di inputkan
Berlanjut	

Lanjutan

SKENARIO NORMAL SUBMIT JUMLAH ATM

Aktor	Sistem
6. Memilih ATM pada <i>dropdown</i> hingga semua ATM yang akan di <i>routing</i> telah di pilih	
7. Klik <i>button</i> "submit"	
	8. Sistem melakukan perhitungan dengan metode <i>Cheapest Insertion Heuristic</i>
	9. Mengambil titik awal dan titik akhir ATM yang telah dipilih
	10. Menyisipkan ATM lain. Dan melakukan perhitungan penjumlahan jarak dengan menggunakan rumus $[C_{ik}+C_{kj}-C_{ij}]$
	11. Memilih jarak terkecil untuk dijadikan sebagai rute yang baru
	12. Menyisipkan ATM lain yang tersisa pada rute yang baru dan menghitung total jarak dengan rumus $[C_{ik}+C_{kj}-C_{ij}]$
	13. Dilakukan berulang kali hingga seluruh ATM telah masuk pada rute
	14. Memilih total jarak terkecil dari opsi penyisipan seluruh rute

Berlanjut

Lanjutan

SKENARIO NORMAL *SUBMIT JUMLAH ATM*

- | | |
|--|---|
| | 15. Sistem menampilkan halaman Hasil Routing dengan Peta dan keterangan langkah perjalanan yang akan ditempuh dengan alamat dan total jarak keseluruhan |
|--|---|

SKENARIO NORMAL *RESET JUMLAH ATM*

Aktor	Sistem
6. Memilih ATM pada <i>dropdown</i> hingga semua ATM yang akan di <i>routing</i> telah di pilih	
7. Klik <i>button</i> “Reset”	8. Menampilkan halaman Perhitungan Rute dengan <i>dropdown</i> kembali tidak tersisi

SKENARIO ALTERNATIF INPUT JUMLAH ATM

Jika aktor tidak menginputkan jumlah ATM

Aktor	Sistem
4. Klik <i>button</i> “add”	5. Menampilkan halaman Routing ATM dan <i>alert</i> “Maaf! Isi data dengan lengkap”

SKENARIO ALTERNATIF SUBMIT JUMLAH ATM

Jika aktor tidak memilih *dropdown* ATM

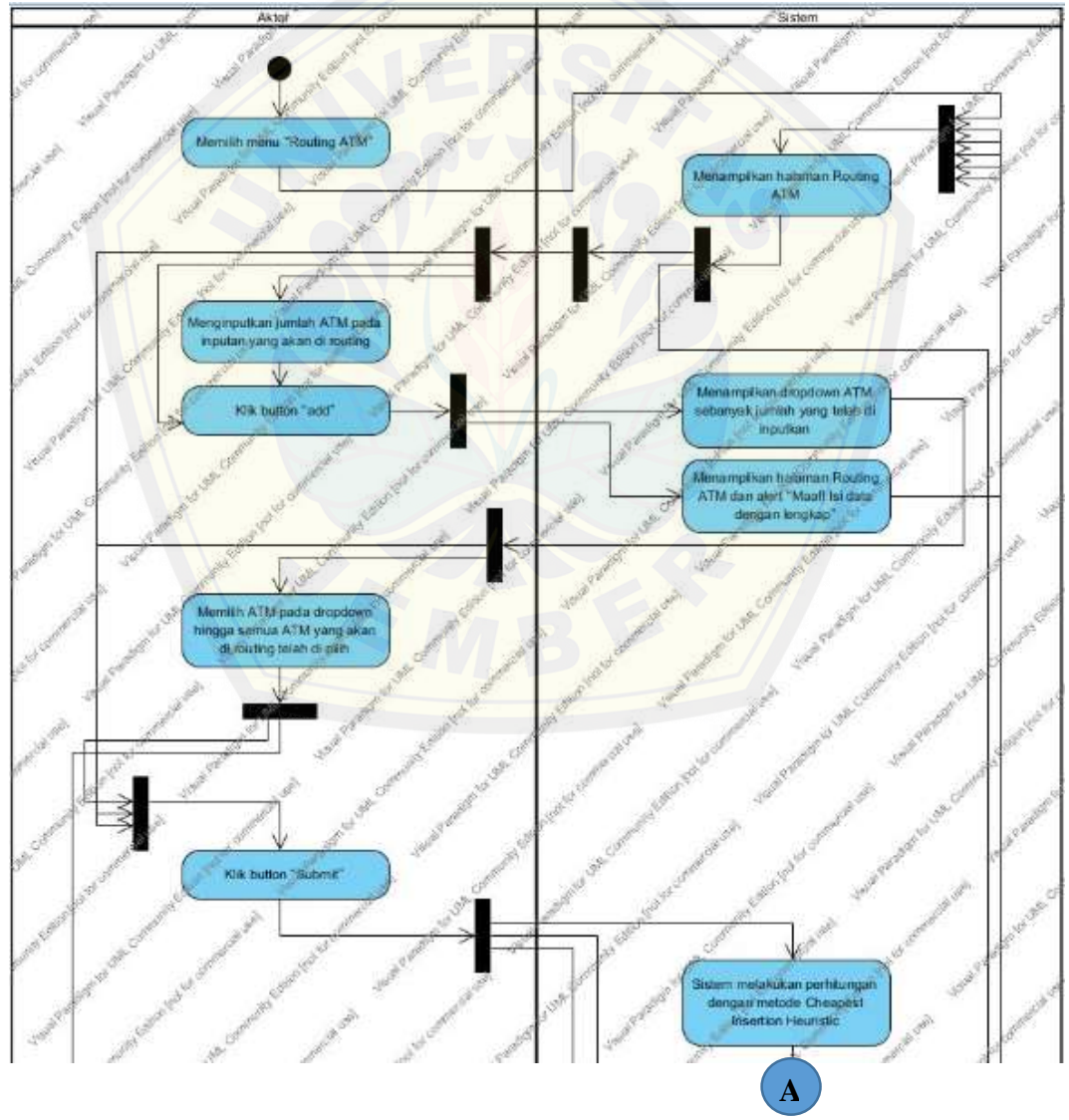
Aktor	Sistem
7. Klik <i>button</i> “Submit”	8. Sistem menampilkan <i>alert</i> “Please select an item in the list”

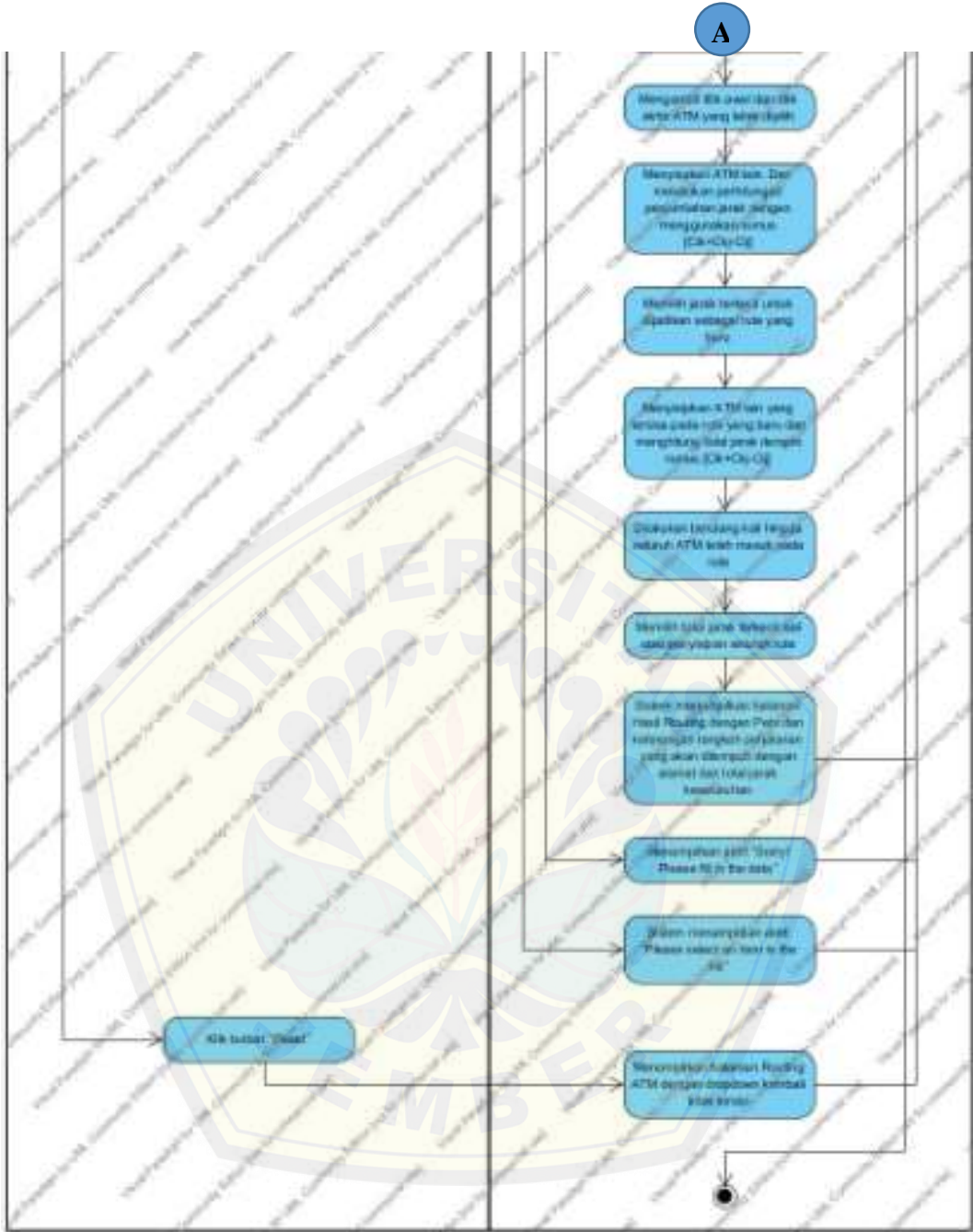
4.3.4 Activity Diagram

Activity Diagram merupakan tahapan untuk menggambarkan aliran aktivitas dalam system pendistribusian pengisian ATM, yang dapat dilihat sebagai berikut:

1. Activity Diagram Menghitung Rute

Activity diagram Menghitung Rute dapat dilakukan oleh *user*. Activity diagram perhitungan rute digunakan untuk melakukan perhitungan untuk optimasi rute menggunakan metode *CIH* untuk jumlah ATM yang telah diinputkan, yang dapat dilihat pada Gambar 4.3.





Gambar 4.3 Activity Diagram Perhitungan Rute

4.3.5 Sequence Diagram

Sequence Diagram adalah tahapan dokumentasi suatu aliran diagram terurut yang menampilkan interaksi - interaksi antar objek di dalam system, yang juga menggambarkan skenario dan memodelkan aliran logika dalam sistem dengan cara

visual. Sequence diagram dari sistem Pendistribusian Pengisian ATM adalah sebagai berikut:

1. *Sequence Diagram Login*

Sequence diagram *login* menggambarkan tentang penjelasan dari fungsi atau *method* yang akan digunakan pada *class login*. *Class view* atau tampilan yang terdapat pada *class login* adalah *vu_index* yang berperan ketika kita berada di halaman *user* dan ingin *login*, *vu_login* adalah halaman dimana kita akan melakukan *login* pada tampilan *login*, *va_index* adalah tampilan yang digunakan ketika kita berhasil melakukan *login*. *Class controller* yakni *cu_index*, *cu_login* dan *ca_index* yang berfungsi untuk memanggil *class* tampilan yang akan ditampilkan, dengan fungsi-fungsi yang akan digunakan, seperti pada *cu_login* terdapat fungsi *login()* untuk melakukan proses *login*. *Class Model* yang digunakan adalah *m_login* yang digunakan untuk melakukan pengambilan data ketika proses *login*, sehingga kita data *login* sesuai maka kita dapat pindah pada halaman selanjutnya, tetapi ketika data tidak sesuai maka kita tidak akan bisa melakukan proses *login*. Penggambaran *sequence login* dapat dilihat pada lampiran.

2. *Sequence Diagram Logout*

Sequence diagram logout menggambarkan tentang proses *logout* dari halaman admin menuju halaman *user*. *Class* yang digunakan hanya *class va_Admin* yakni halaman tampilan admin sebelum kita melakukan *logout*, dan halaman *vu_user* adalah tampilan yang digunakan setelah kita berhasil melakukan proses *logout*. *Class controler* yang digunakan hanyalah *ca_index* yang di dalamnya terdapat fungsi *logout()* yang digunakan sebagai proses untuk melakukan *logout*. Penggambaran *sequence logout* dapat dilihat pada lampiran.

3. *Sequence Diagram Melihat Peta ATM Admin*

Sequence diagram melihat peta ATM admin adalah penggambaran sistem dengan menggunakan *class view* yakni *va_index* yang merupakan tampilan yang digunakan untuk melihat tampilan peta untuk admin, selanjutnya *class controller* yang digunakan adalah *ca_index* yang digunakan untuk melakukan

proses pemanggilan data pada *database* melalui *model* untuk dapat menampilkan peta pada halaman *va_index* dan yang terakhir adalah *class model* yakni *m_view* yang digunakan untuk melakukan pemanggilan data pada *database* sesuai pada *controller*. Penggambaran *sequence* melihat peta ATM admin dapat dilihat pada lampiran.

4. *Sequence* Diagram Melihat Peta ATM User

Sequence diagram melihat peta ATM *user* adalah penggambaran sistem dengan menggunakan *class view* yakni *vu_index* yang merupakan tampilan yang digunakan untuk melihat tampilan peta untuk *user*, selanjutnya *class controller* yang digunakan adalah *cu_index* yang digunakan untuk melakukan proses pemanggilan data pada *database* melalui *model* untuk dapat menampilkan peta pada halaman *va_index*, dan yang terakhir adalah *class model* yakni *m_view* yang digunakan untuk melakukan pemanggilan data pada *database* sesuai pada *controller*. Penggambaran *sequence* melihat peta ATM *user* dapat dilihat pada lampiran.

5. *Sequence* Diagram Melihat Data BRI

Sequence diagram melihat data BRI adalah penggambaran sistem untuk melihat profil tentang BRI yang digambarkan dengan menggunakan *class view* yakni *vu_user* yang merupakan tampilan utama yang digunakan sehingga selanjutnya kita dapat melakukan menu yang terdapat pada *va_sejarah* dan *va_visimisi* yang merupakan tampilan untuk menampilkan halaman sejarah dari bank BRI dan halaman visi dan misi. *Class controller* yang digunakan adalah *cu_bri* untuk melakukan proses pemindahan halaman dengan menggunakan fungsi *sejarah()* dan fungsi *visimisi()*. Penggambaran *sequence* melihat data BRI dapat dilihat pada lampiran.

6. *Sequence* Diagram Menghitung Jarak

Sequence diagram menghitung jarak digunakan untuk menggambarkan proses admin menginputkan data jarak yang didapatkan dari perhitungan antara 2 ATM. *Class* yang digunakan yakni *va_admin* dan *va_atm* pada *class* tampilan, yang digunakan untuk menampilkan halaman untuk melakukan perhitungan jarak yang terdapat juga fungsi *calculateDistance()* pada

halaman *va_atm* yang dibuat menggunakan kode *javascript* untuk melakukan semua proses perhitungan, selanjutnya *ca_atm* pada *class controller* yang digunakan untuk melakukan pemindahan tampilan dan melakukan penyimpanan data proses perhitungan pada database menggunakan fungsi *tambah_jarak()* dan yang terakhir adalah *m_save* yakni *class model* yang digunakan untuk menghubungkan proses penyimpanan dari *controller* menuju *database* dengan menggunakan *query*. Penggambaran *sequence* perhitungan jarak dapat dilihat pada lampiran.

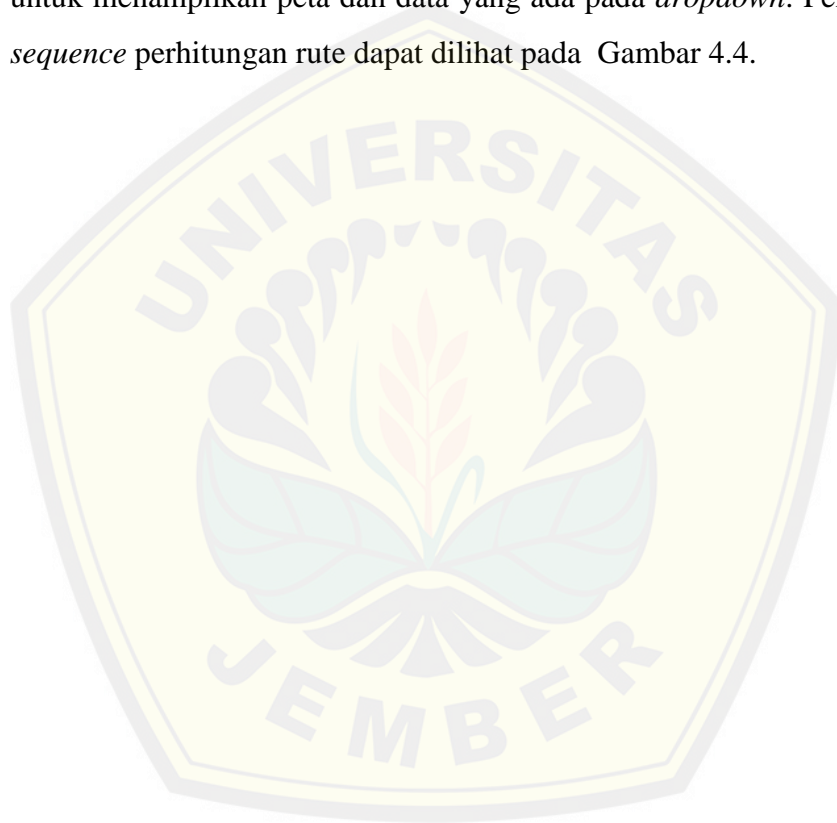
7. *Sequence* Diagram Manajemen Data ATM

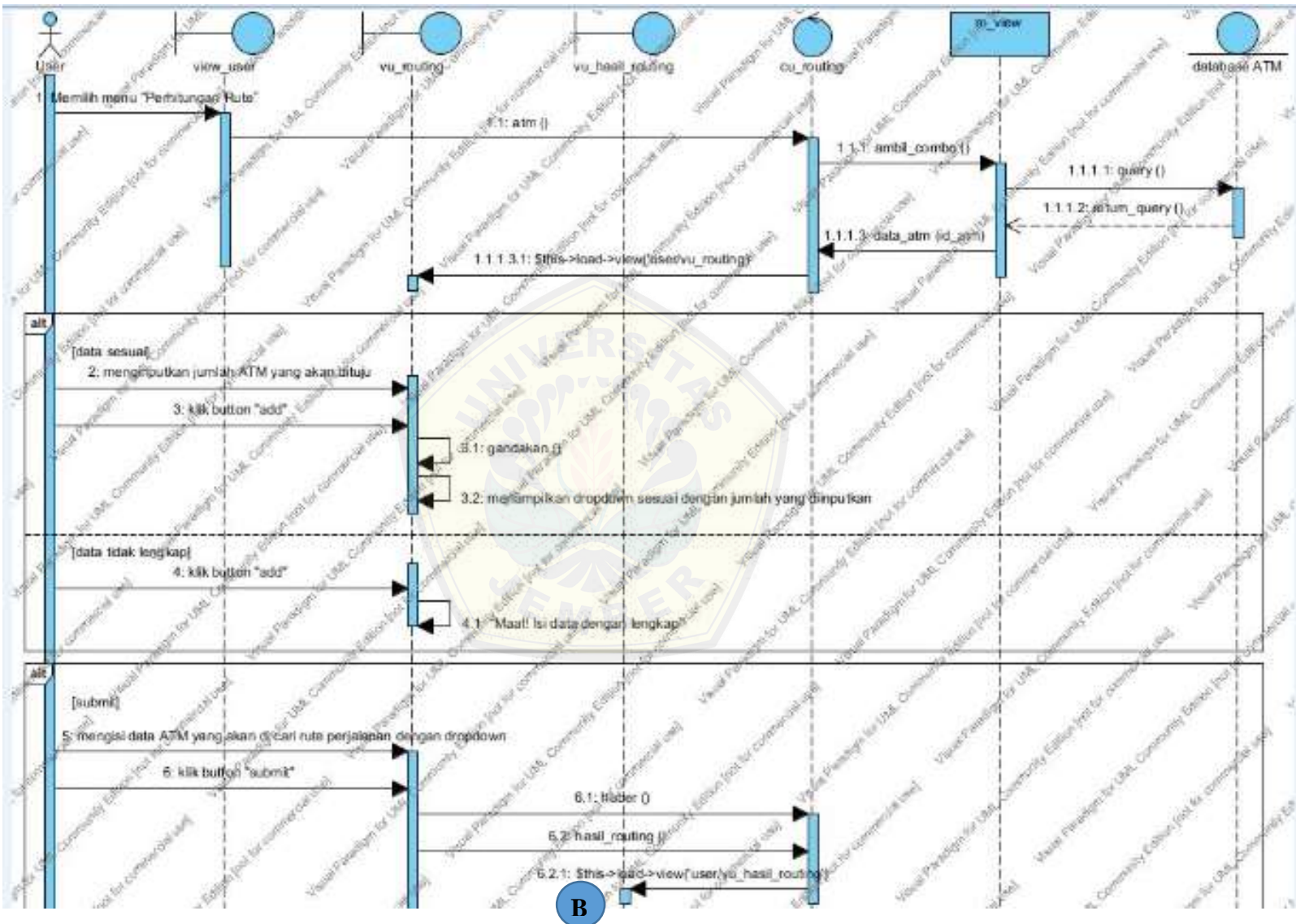
Sequence diagram manajemen data ATM adalah *class* yang digunakan untuk menggambarkan proses penambahan, pengeditan, dan penghapusan data ATM. *Class* yang digunakan pada *class view* adalah *va_admin*, *va_data_atm* yakni tampilan untuk menampilkan seluruh keterangan tentang data ATM yang terdapat *button* untuk melakukan penambahan, pengeditan dan penghapusan data, *va_tambah_data_atm* adalah tampilan yang digunakan untuk melakukan proses penambahan ATM, *va_edit_data_atm* tampilan yang digunakan untuk melakukan proses pengeditan ATM yang sudah ada. *Class controller* yang digunakan adalah *ca_data_atm* yang digunakan untuk melakukan proses penambahan dengan menggunakan fungsi *tambah_atm()*, proses pengeditan dengan menggunakan fungsi *edit_atm()*, dan proses penghapusan dengan menggunakan fungsi *hapus_data_atm()*. *Class model* yang digunakan adalah *m_view* untuk menampilkan data ATM, *m_save* untuk melakukan proses penyimpanan penambahan dan pengeditan data ATM pada *database*, dan *m_delete* untuk melakukan proses penghapusan data ATM pada *database*. Penggambaran *sequence* manajemen data ATM dapat dilihat pada lampiran.

8. *Sequence* Diagram Menghitung Rute

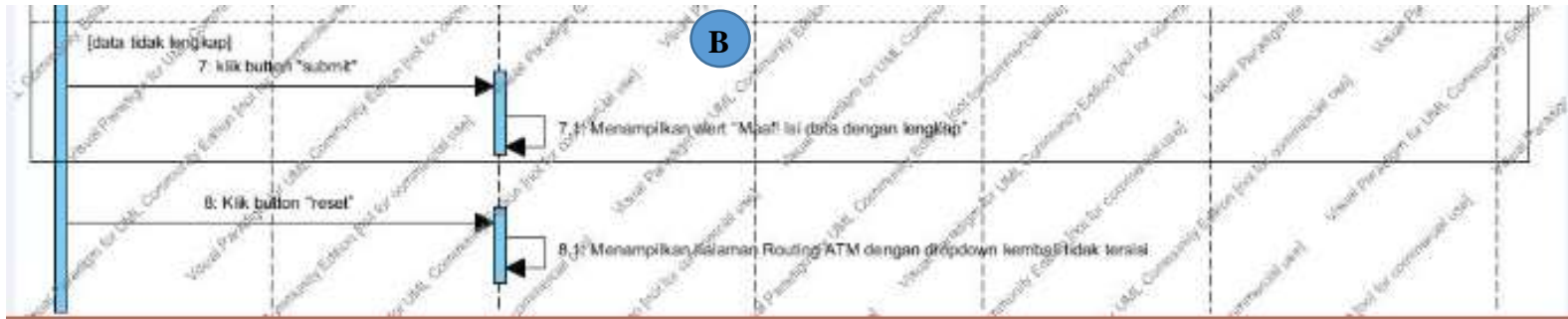
Sequence diagram menghitung rute adalah tahapan yang digunakan untuk melakukan proses perhitungan rute dengan menggunakan metode. *Class view* yang digunakan adalah *vu_user*, *vu_routing* yang merupakan tampilan dimana proses penginputan jumlah ATM yang diinputkan, dan proses

pemilihan ATM yang akan digunakan untuk pencarian optimasi rute, yang nanti hasilnya akan ditampilkan pada tampilan *va_hasil_routing*. *Class controller* yang digunakan adalah *class cu_routing* yang merupakan *class* yang memiliki fungsi *haber()* untuk melakukan proses perhitungan optimasi rute dengan menggunakan metode dan juga fungsi *hasil_routing* untuk menampilkan halaman yang digunakan untuk menampilkan hasil perhitungan. *Class model* yang digunakan adalah *m_view* yang digunakan untuk menampilkan peta dan data yang ada pada *dropdown*. Penggambaran *sequence* perhitungan rute dapat dilihat pada Gambar 4.4.

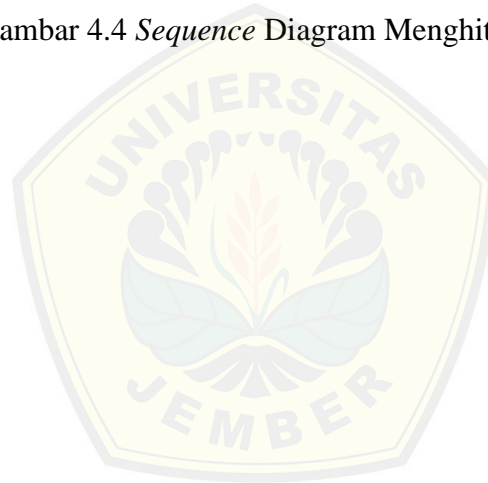




B



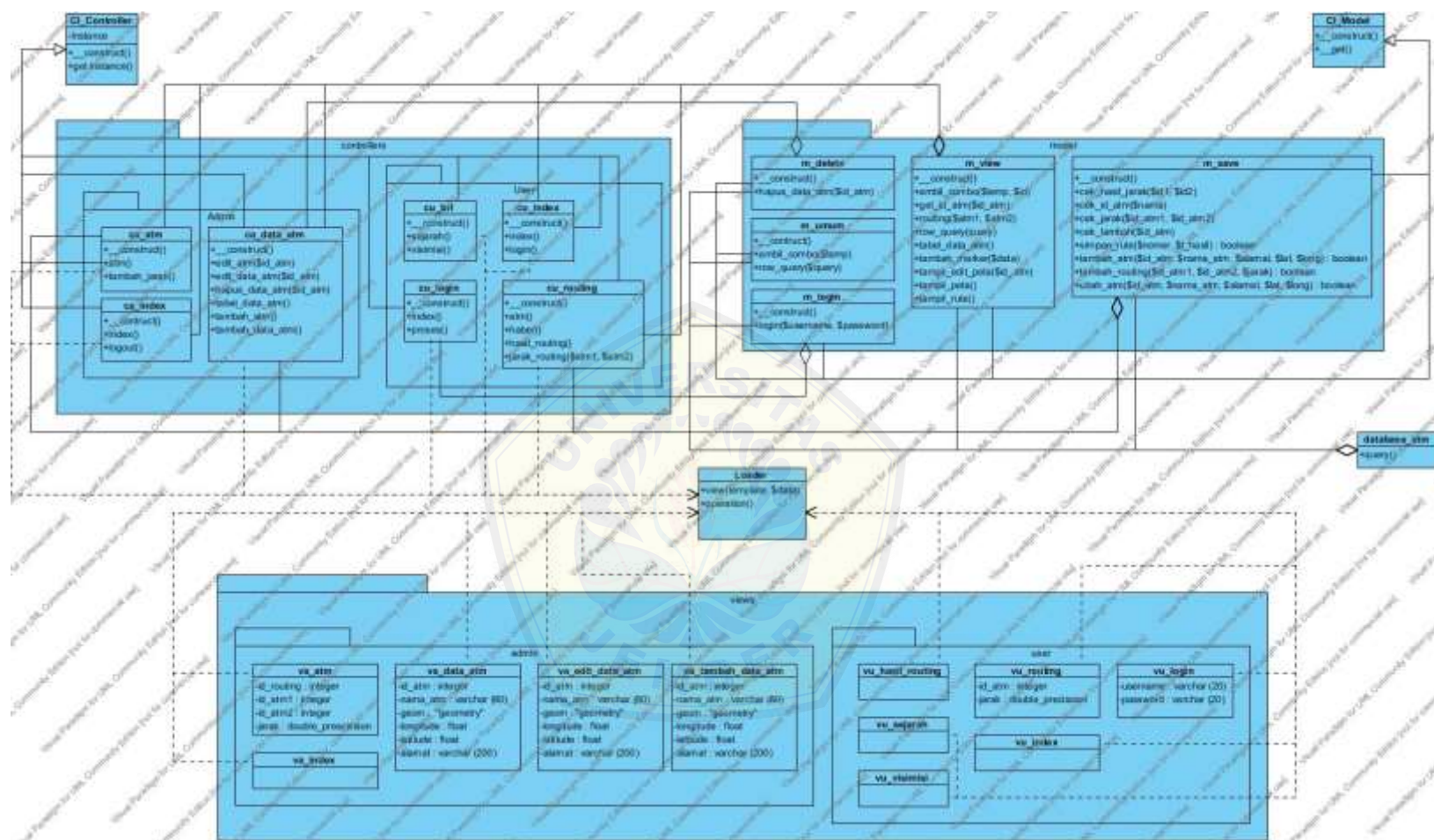
Gambar 4.4 *Sequence* Diagram Menghitung Rute



4.3.6 Class Diagram

Class diagram adalah tahapan yang menggambarkan hubungan antar kelas-kelas yang digunakan dalam membentuk sistem. *Model, View, Controller* yang biasa disebut dengan *MVC* pada *OOP (Object Oriented Program)* adalah contoh-contoh kelas yang digunakan. *Class Diagram* yang dibangun berdasarkan *sequence* diagram pada sistem pendistribusian pengisian ATM terdiri dari 5 *class model* dan 7 *class controller* yang dapat dilihat pada Gambar 4.5.

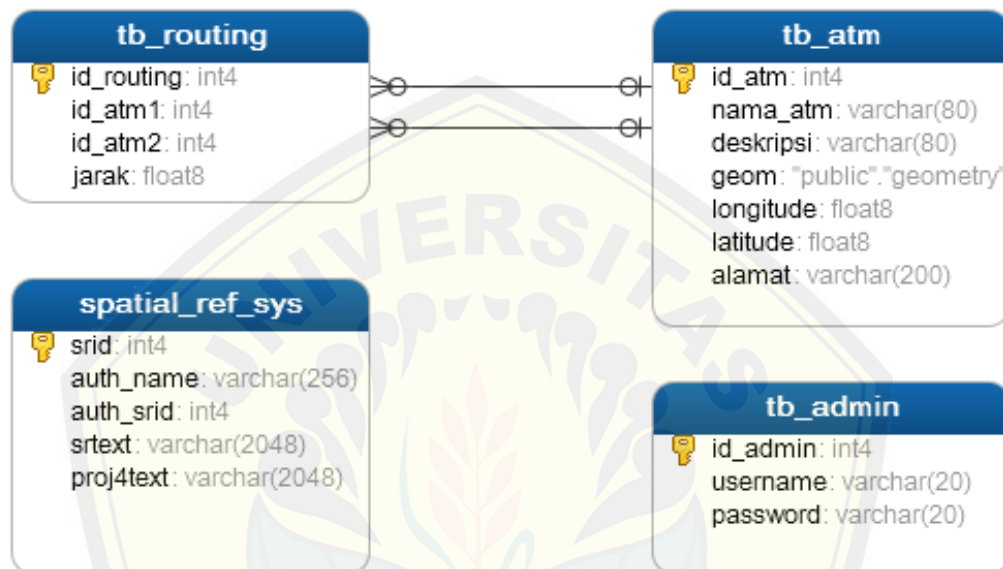




Gambar 4.5 Class Diagram Pendistribusian Pengisian ATM

4.3.7 Entity Relationship Diagram

Entity Relationship Diagram merupakan tahapan untuk menggambarkan komponen dan struktur dari *database* yang digunakan dalam membuat suatu sistem. *ERD* yang dihasilkan dari *database* pendistribusian pengisian ATM terdiri dari 4 entitas yang selanjutnya dapat dilihat pada Gambar 4.6.



Gambar 4.6 *ERD* Pendistribusian Pengisian ATM

4.4 Penulisan Kode Program

Tahapan penulisan kode program adalah tahapan dimana pengimplementasian yang dilakukan dari proses yang telah kita buat menjadi sebuah kode program.

1. Bahasa yang digunakan dalam penulisan kode program yakni bahasa pemrograman *Page Hyper Text Pre-Processor* (PHP) dan juga menggunakan bantuan *framework Code Igniter* (CI).
2. Manajemen basis data menggunakan *DBMS PostgreSQL*.

Kode program yang digunakan untuk proses perhitungan menggunakan algoritma *CIH* terletak pada *class* *cu_routing* yang terdapat pada *package* *user* dalam *package* *controller*, yang dapat dilihat pada Tabel 4.4.

Tabel 4.4 Kode Program *Function* *haber()*

42.	<code>public function haber() {</code>
43.	<code> \$hasil = array(\$atm[0]['value'], \$atm[(count(\$atm) - 1)]['value']);</code>
44.	<code> \$id_in_hasil = array(0,(count(\$atm) - 1));</code>
45.	
46.	<code> while (count(\$hasil) < count(\$atm)) {</code>
47.	<code> // pembentuk nilai yang disisipkan & tabel hasil</code>
48.	<code> \$terkecil = "";</code>
49.	<code> \$id_in_proses = array();</code>
50.	<code> for (\$i = 0, \$ln = count(\$atm); \$i < \$ln; \$i++) {</code>
51.	<code> \$sada = FALSE;</code>
52.	<code> for (\$j = 0, \$ln2 = count(\$hasil); \$j < \$ln2; \$j++) {</code>
53.	<code> if (\$hasil[\$j] == \$atm[\$i]['value']) {</code>
54.	<code> \$sada = TRUE;</code>
55.	<code> }</code>
56.	<code> }</code>
57.	<code> if (!\$sada) {</code>
58.	<code> array_push(\$id_in_proses, \$i);</code>
59.	<code> }</code>
60.	<code> }</code>
61.	
62.	<code> // pembentuk tabel proses dan terkecil</code>
63.	<code> for (\$j = 0, \$ln = count(\$id_in_hasil); \$j < (\$ln - 1); \$j++) {</code>
64.	<code> for (\$k = 0, \$ln2 = count(\$id_in_proses); \$k < \$ln2; \$k++) {</code>
65.	<code> \$a_to_s = \$this->jarak_routing(\$atm[\$id_in_hasil[\$j]]['value'], \$atm[\$id_in_proses[\$k]]['value']);</code>
66.	<code> \$s_to_t = \$this->jarak_routing(\$atm[\$id_in_proses[\$k]]['value'], \$atm[\$id_in_hasil[(\$j +</code>
67.	<code>1)]]['value']);</code>
68.	<code> \$a_to_t = \$this->jarak_routing(\$atm[\$id_in_hasil[\$j]]['value'], \$atm[\$id_in_hasil[(\$j +</code>
69.	<code>1)]]['value']);</code>
70.	<code> echo \$id_in_hasil[\$j] . ' ' . \$id_in_proses[\$k] . ' ' . \$id_in_hasil[(\$j + 1) . ' ' .</code>
71.	<code> (\$a_to_s + \$s_to_t - \$a_to_t);</code>
72.	
73.	<code> // meletakkan nilai terkecil</code>
74.	<code> if (\$terkecil != "") {</code>
75.	<code> \$tmp = explode(";", \$terkecil);</code>
76.	<code> if (\$tmp[1] > (\$a_to_s + \$s_to_t - \$a_to_t)) {</code>

```

77.         $sterkecil = $id_in_hasil[$j] . ';ID;' . $id_in_proses[$k] . ';ID;' . $id_in_hasil[( $j + 1)] . ';H;'
78.         . ($a_to_s + $s_to_t - $a_to_t);
79.     }
80. } else {
81.     $sterkecil = $id_in_hasil[$j] . ';ID;' . $id_in_proses[$k] . ';ID;' . $id_in_hasil[( $j + 1)] . ';H;'
82.     . ($a_to_s + $s_to_t - $a_to_t);
83. }
84. echo '<br />';
85. }
86. if ( $j == ($ln - 2) ) {
87.     echo '<br />';
88.     for ( $k = 0, $ln2 = count($id_in_proses); $k < $ln2; $k++ ) {
89.         $a_to_s = $this->jarak_routing($atm[$id_in_hasil[( $ln - 1)]]['value'],
90. $atm[$id_in_proses[$k]]['value']);
91.         $s_to_t = $this->jarak_routing($atm[$id_in_proses[$k]]['value'],
92. $atm[$id_in_hasil[0]]['value']);
93.         $a_to_t = $this->jarak_routing($atm[$id_in_hasil[( $ln - 1)]]['value'],
94. $atm[$id_in_hasil[0]]['value']);
95.         echo $id_in_hasil[( $ln - 1)] . '|' . $id_in_proses[$k] . '|' . $id_in_hasil[0] . '|'
96.         . ($a_to_s + $s_to_t - $a_to_t);
97.
98.         // meletakkan nilai terkecil
99.         $tmp = explode("H;", $sterkecil);
100.        if ($tmp[1] > ($a_to_s + $s_to_t - $a_to_t) ) {
101.            $sterkecil = $id_in_hasil[( $ln - 1)] . ';ID;' . $id_in_proses[$k] . ';ID;' . $id_in_hasil[0] . ';H;'
102.            . ($a_to_s + $s_to_t - $a_to_t);
103.        }
104.        echo '<br />';
105.    }
106. }
107. echo '<br />';
108. }
109. echo 'Terkecil : ' . $sterkecil;
110. echo '<br /><br />';
111.
112. $tmp1 = explode("H;", $sterkecil);
113. $tmp2 = explode("ID;", $tmp1[0]);
114. $tmp_proses = array();
115. for ( $l = 0, $ln3 = count($id_in_hasil); $l < $ln3; $l++ ) {
116.     array_push($tmp_proses, $id_in_hasil[$l]);
117.     for ( $m = 0, $ln4 = count($tmp2); $m < ($ln4 - 1); $m++ ) {

```

```
118.         if ($id_in_hasil[$l] == $tmp2[$m]) {
119.             array_push($tmp_proses, $tmp2[(($m + 1)]));
120.         }
121.     }
122. }
123. $id_in_hasil = $tmp_proses;
124.
125. $tmp_hasil = array();
126. for ($n = 0, $ln5 = count($id_in_hasil); $n < $ln5; $n++) {
127.     array_push($tmp_hasil, $atm[$id_in_hasil[$n]]['value']);
128. }
129. $t_hasil = $tmp_hasil;
130. }
131. echo 't_hasil : '; print_r($t_hasil);
132. echo '<br /><br />';
133. echo 'id_hasil : '; print_r($id_in_hasil);
134. echo '<br /><br />';
135. echo 'id_progres : '; print_r($id_in_proses);
136. echo '<br /><br />';
137. echo 'Rute tercepat : ' . $t_hasil[0];
138.
139. //menyimpan pada tabel hasil, dengan id=1 diinput pertama
140. $nomer = 1;
141. $jumlah=0;
142. $o = 0;
143. for ($i = 1; $i < count($t_hasil); $i++) {
144.     echo ' --> ' . $t_hasil[$i];
145.     $nomer++;
146.     $jumlah = $jumlah + $this->jarak_routing($t_hasil[0], $t_hasil[1]);
147.     echo ' --> ' . $jumlah;
148.     $o++;
149. }
150. echo '<br /><br /><br />';
151.
152. $rute_perjalanan = array();
153. for ($i = 0; $i < count($t_hasil); $i++) {
154.     array_push($rute_perjalanan, $this->m_view->ambil_combo("tb_atm", $t_hasil[$i]));
155. }
156. $this->session->set_userdata("rute_perjalanan", $rute_perjalanan);
157. $this->session->set_userdata("total_jarak", $jumlah);
```

Penggambaran kode program di atas merupakan kode program *fuction* `haber()` yang digunakan untuk melakukan proses perhitungan rute dengan menggunakan algoritma *CIH*. Tahapan pertama yakni memasukan nilai *array* `$t_hasil` dan `$id_in_hasil` yang dapat dilihat pada baris program nomer 43 dan 44, selanjutnya adalah melakukan proses pengulangan hingga syarat terpenuhi, yakni nilai dari `$t_hasil` yang diurutkan kurang dari jumlah `$atm` yang diinputkan yang dapat dilihat pada kode program nomer 46.

Tahapan selanjutnya adalah pembentukan nilai ATM yang disisipkan, dengan melakukan proses pengulangan dengan syarat bahwa nilai yang disisipkan bukan merupakan nilai awal dan nilai akhir yang selanjutnya akan dimasukan ke dalam variabel `$id_in_proses` yang terdapat pada kode program nomer 49 hingga 60, dan selanjutnya juga melakukan pengisian variabel terkecil dengan nilai yang masih kosong yang terdapat pada kode program bernomor 48.

Pengerjaan selanjutnya adalah untuk membentuk tabel proses dan melakukan perhitungan dengan rumus yang didapatkan dari penggunaan algoritma *CIH* untuk mendapatkan nilai terkecil guna mendapatkan *subtour* baru. Awal melakukan pengulangan untuk menyisipkan nilai yang ada pada variabel `$id_in_proses` yang menjadi nilai sisipan, selanjutnya diulang juga dengan mencari jarak untuk mendapatkan nilai terkecil, yang prosesnya dapat dilihat pada nomer 74 hingga 85. Apabila nilai sisipan semua telah berhasil masuk, maka akan dilanjutkan dengan membalik nilai awal menjadi nilai tujuan dan nilai tujuan menjadi nilai awal, dan melakukan pengulangan kembali hingga semua nilai sisipan berhasil masuk yang dilihat pada kode program nomer 86 hingga 108. Tahapan selanjutnya adalah ketika nilai terkecil yang telah didapatkan akan ditampilkan yang dapat dilihat pada kode program nomer 109.

Pembentukan tabel hasil dengan cara data yang didapatkan dari perhitungan sebelumnya, yakni pada tabel proses dengan jarak terkecil akan menjadi *subtour* baru, yang akan di masukan dalam tabel hasil dengan cara melakukan pengulangan hingga nilai pada seluruh *subtour* telah masuk, yang dapat dilihat pada kode program bernomor 112 hingga 123. Tahapan berikutnya adalah nilai *subtour* baru

yang ada pada variabel \$tmp_hasil akan dimasukkan pada variabel \$t_hasil dan memperbarui data sebelumnya, yang dapat dilihat pada kode program bernomor 129, selanjutnya adalah melakukan pengulangan kembali hingga seluruh nilai ATM yang dimasukkan telah berhasil masuk.

Pengecekan dilakukan dengan cara mencetak hasil yang telah dihitung, yang dapat dilihat pada kode program 131 hingga 137. Perhitungan terakhir adalah menghitung total jarak dari hasil seluruh *subtour* yakni dengan pengulangan menggunakan *array*, hingga semuanya masuk maka akan mendapatkan total jarak dari perjalanan yang dapat dilihat pada nomer 152 hingga 155 dan pemanggilan rute perjalanan, total jarak dapat dilihat pada nomer 156 hingga 157.

4.5 Pengujian Sistem

Pada penelitian ini peneliti menggunakan dua metode pengujian sistem yaitu *White Box Testing* dan *Black Box Testing* yang dapat dijelaskan di bawah ini:

4.5.1 *White Box Testing*

White box testing adalah pengujian yang dilakukan setelah sistem dibuat dengan melihat kode program, macam *white box testing* adalah *listing* program, grafik alir, kompleksitas siklomatis, pengujian jalur program dan *test case*. Pengujian yang akan dilakukan pada aplikasi pendistribusian pengisian ATM adalah pengujian pada fitur manajemen data ATM dan fitur perhitungan rute, yang dijelaskan seperti berikut ini:

1. *Pengujian White Box* Menggunakan Manajemen Data ATM

Pengujian *white box* manajemen data ATM mencakup fitur menambah, mengubah dan menghapus data ATM, yang dapat dilihat di bawah ini:

(a) *Listing* Program Fitur Manajemen Data ATM

1) *Listing* Program Menambah Data ATM

```

1 public function tambah_atm() { } 1
2 $this->form_validation->set_rules('id_atm', 'id_atm', 'required');
3 $this->form_validation->set_rules('nama_atm', 'nama_atm', 'required');
4 $this->form_validation->set_rules('alamat', 'alamat', 'required');
5 $this->form_validation->set_rules('lat', 'lat', 'required');
6 $this->form_validation->set_rules('long', 'long', 'required');
7 if ($this->form_validation->run()) { } 3
8     $id_atm = $this->input->post('id_atm', true);
9     $nama_atm = $this->input->post('nama_atm', true);
10    $alamat = $this->input->post('alamat', true);
11    $lat = $this->input->post('lat', true);
12    $long = $this->input->post('long', true);
13    $cek = $this->m_save->cek_tambah($id_atm);
14    if ($cek->id_atm == null) { } 6
15        $query = $this->m_save->tambah_atm($id_atm, $nama_atm, $alamat, $lat, $long); } 7
16    }
17    if ($query) { } 8
18        $this->session->set_userdata('operation', "sukses");
19        $this->session->set_userdata('message', "Berhasil Input ATM");
20        redirect("admin/ca_data_atm/tabel_data_atm");
21    } else {
22        $this->session->set_userdata('operation', "gagal");
23        $this->session->set_userdata('message', "Gagal Input ATM");
24        redirect("admin/ca_data_atm/tambah_data_atm");
25    }
26 } else {
27     $this->session->set_userdata('operation', "gagal");
28     $this->session->set_userdata('message', "Isi data dengan lengkap");
29     redirect("admin/ca_data_atm/tambah_data_atm");
30 }
31 } } 11
    
```

Gambar 4.7 Listing Program Menambah Data ATM

2) Listing Program Mengubah Data ATM

```

114 public function edit_atm($id_atm) { } 1
115 $this->form_validation->set_rules('id_atm', 'id_atm', 'required');
116 $this->form_validation->set_rules('nama_atm', 'nama_atm', 'required');
117 $this->form_validation->set_rules('alamat', 'alamat', 'required');
118 $this->form_validation->set_rules('lat', 'lat', 'required');
119 $this->form_validation->set_rules('long', 'long', 'required');
120 if ($this->form_validation->run()) { } 3
121     $id_atm = $this->input->post('id_atm', true);
122     $nama_atm = $this->input->post('nama_atm', true);
123     $alamat = $this->input->post('alamat', true);
124     $lat = $this->input->post('lat', true);
125     $long = $this->input->post('long', true);
126     $query = $this->m_save->ubah_atm($id_atm, $nama_atm, $alamat, $lat, $long);
127     if ($query) { } 6
128         $this->session->set_userdata('operation', "sukses");
129         $this->session->set_userdata('message', "Berhasil mengubah data ATM");
130         redirect("admin/ca_data_atm/tabel_data_atm");
131     } else {
132         $this->session->set_userdata('operation', "gagal");
133         $this->session->set_userdata('message', "Gagal mengubah data ATM");
134         redirect("admin/ca_data_atm/edit_data_atm/$id_atm");
135     }
136 } else {
137     $this->session->set_userdata('operation', "gagal");
138     $this->session->set_userdata('message', "Isi data dengan lengkap");
139     redirect("admin/ca_data_atm/edit_data_atm/$id_atm");
140 } } } 9
    
```

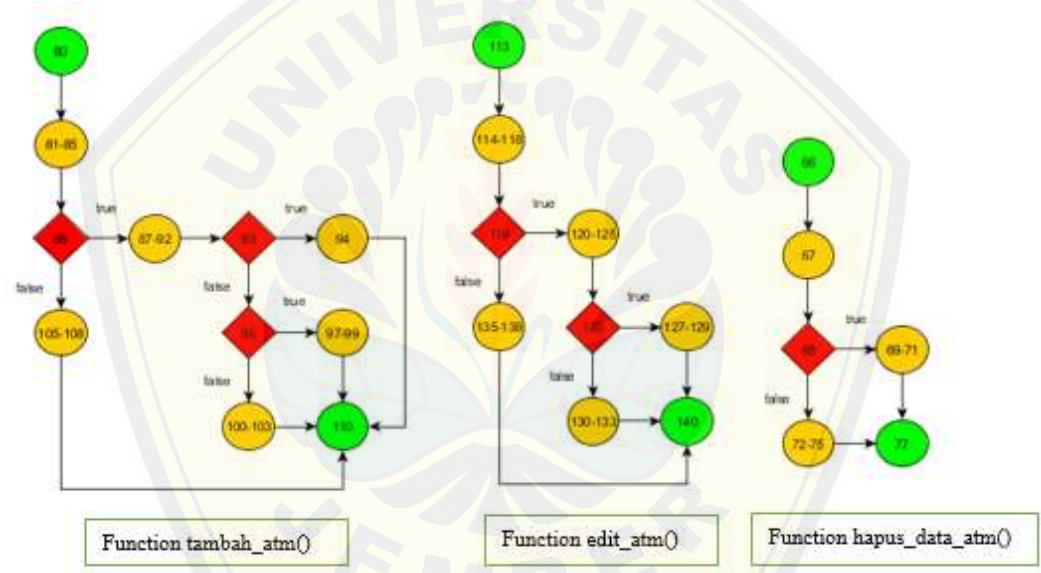
Gambar 4.8 Listing Program Mengubah Data ATM

3) Listing Program Menghapus Data ATM

```
66 public function hapus_data_atm($id_atm) { } 1
67     $query = $this->m_delete->hapus_data_atm($id_atm); } 2
68     if ($query) { } 3
69         $this->session->set_userdata('operation', "sukses");
70         $this->session->set_userdata('message', "Data ATM berhasil dihapus"); } 4
71         redirect("admin/ca_data_atm/tabel_data_atm");
72     } else {
73         $this->session->set_userdata('operation', "gagal");
74         $this->session->set_userdata('message', "Data ATM tidak berhasil dihapus"); } 5
75         redirect("admin/ca_data_atm/tabel_data_atm");
76     }
77 } } 6
```

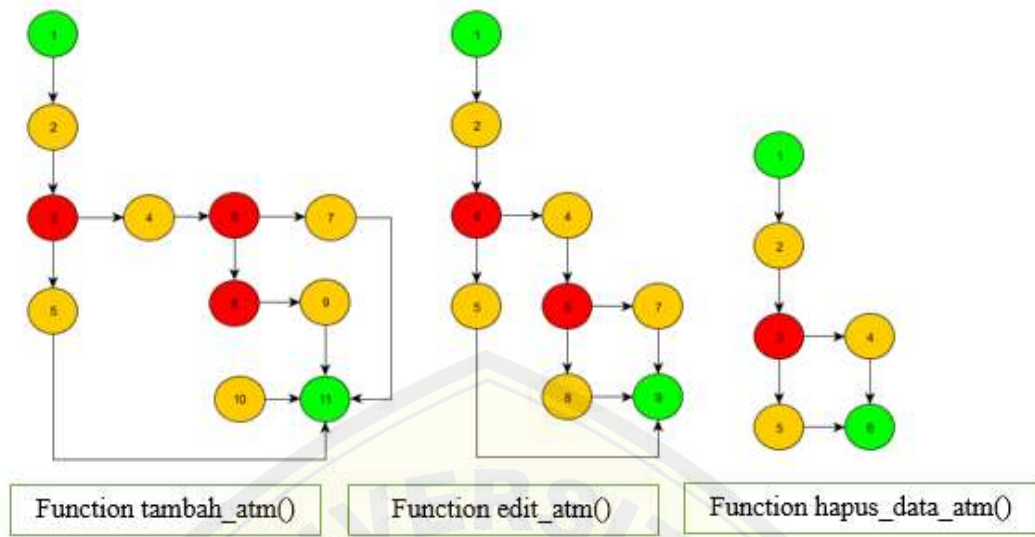
Gambar 4.9 Listing Program Menghapus Data ATM

(b). Diagram Alir Fitur Manajemen Data ATM



Gambar 4.10 Diagram Alir Manajemen Data ATM

(c). Grafik Alir Manajemen Data ATM



Gambar 4.11 Grafik Alir Manajemen Data ATM

(d). Kompleksitaas Siklometik

Tahapan pengujian ini menggunakan rumus $V(G) = E - N + 2$, dimana E adalah jumlah *edge* dan N adalah jumlah *node* grafik alir.

Function tambah_atm() $V(G) = E - N + 2 = 12 - 11 + 2 = 3$

Function edit_atm() $V(G) = E - N + 2 = 10 - 9 + 2 = 3$

Function hapus_data_atm() $V(G) = E - N + 2 = 6 - 6 + 2 = 2$

(e). Pengujian Jalur Program Fitur Manajemen Data ATM

Pengujian jalur program fitur manajemen data ATM berdasarkan gambar di atas adalah sebagai berikut:

1) *Function* tambah_atm() :

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

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

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

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

2) *Function* edit_atm () :

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

Jalur 2 : 1 – 2 – 3 – 4 – 6 – 8 – 9

Jalur 3 : 1 – 2 – 3 – 5 – 9

3) *Function* hapus_data_atm () :

Jalur 1 : 1 – 2 – 3 – 4 – 6

Jalur 2 : 1 – 2 – 3 – 5 – 6

(f). *Test Case* Fitur Manajemen Data ATM

Tabel 4.5 *Test Case* Fitur Manajemen Data ATM

Test Case function tambah_atm()	
Jalur 1	
<i>Test Case</i>	Jika berhasil menyimpan data ATM setelah melakukan pengecekan id_atm
Target yang diharapkan	Menyimpan data ATM ke database dan menampilkan alert “Selamat! Berhasil Input ATM”
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6 – 7 – 11
Jalur 2	
<i>Test Case</i>	Jika berhasil menyimpan data ATM
Target yang diharapkan	Menyimpan data ATM ke database dan menampilkan alert “Selamat! Berhasil Input ATM”
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6 – 8 – 9 – 11
Jalur 3	
<i>Test Case</i>	Jika tidak berhasil menyimpan data ATM setelah melakukan pengecekan id_atm

Berlanjut

Lanjutan

Jalur 3

Target yang diharapkan	Kembali ke halaman tambah data ATM dan menampilkan alert “Maaf! Gagal Input ATM”
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6 – 8 – 10 – 11
Jalur 4	
<i>Test Case</i>	Jika tidak berhasil menyimpan data ATM karena ada data yang tidak diisi atau kosong
Target yang diharapkan	Kembali ke halaman tambah data ATM dan menampilkan alert “Maaf! Isi data dengan lengkap”
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 5 – 11
Test Case function edit_atm()	
Jalur 1	
<i>Test Case</i>	Jika berhasil menyimpan data ATM
Target yang diharapkan	Menyimpan data ATM ke database dan menampilkan alert “Selamat! Berhasil Mengubah Data ATM”
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6 – 7 – 9
Jalur 2	
<i>Test Case</i>	Jika tidak berhasil menyimpan data ATM karena data tidak sesuai
Target yang diharapkan	Kembali ke halaman edit data ATM dan menampilkan alert “Maaf! Gagal Input ATM”
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6 – 8 – 9

Berlanjut

Lanjutan

Jalur 3	
<i>Test Case</i>	Jika tidak berhasil menyimpan data ATM karena ada data yang tidak diisi atau kosong
Target yang diharapkan	Kembali ke halaman edit data ATM dan menampilkan alert “Maaf! Isi data dengan lengkap”
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 5 – 9
<i>Test Case function hapus_data_atm()</i>	
Jalur 1	
<i>Test Case</i>	Jika berhasil menghapus data ATM
Target yang diharapkan	Menghapus data ATM ke database dan menampilkan alert “Selamat! Data ATM berhasil dihapus”
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 6
Jalur 2	
<i>Test Case</i>	Jika tidak berhasil menghapus data ATM
Target yang diharapkan	Kembali ke halaman data ATM dan menampilkan alert “Maaf! Data ATM tidak berhasil dihapus”
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 5 – 6

2. *Pengujian White Box* Fitur Perhitungan Rute

White box testing adalah pengujian yang dilakukan setelah sistem dibuat dengan melihat kode program, macam *white box testing* adalah *listing* program, grafik alir, kompleksitas siklomatis, pengujian jalur program dan *test case*. Pengujian yang akan dilakukan pada aplikasi pendistribusian pengisian ATM adalah pengujian pada fitur manajemen data ATM dan fitur perhitungan rute, yang dijelaskan seperti berikut ini:

(a). Listing Program Fitur Perhitungan Rute

```

1 public function haber() { } 1
2   $arr = $REQUEST['arr'];
3   $c_hasil = array($arr[0]['value'], $arr[(count($arr) - 1)]['value']);
4   $id_in_hasil = array(0, (count($arr) - 1));
5
6   while (count($c_hasil) < count($arr)) { } 3
7     $sterkecil = "";
8     $id_in_proses = array(); } 4
9     for ($i = 0, $in = count($arr); $i < $in; $i++) { } 5
10      $seda = FALSE; } 6
11      for ($j = 0, $in2 = count($c_hasil); $j < $in2; $j++) { } 7
12        if ($c_hasil[$j] == $arr[$i]['value']) { } 8
13          $seda = TRUE; } 9
14        }
15      if (!$seda) { } 10
16        array_push($id_in_proses, $i); } 11
17    }
18
19    for ($j = 0, $in = count($id_in_hasil); $j < ($in - 1); $j++) { } 12
20      for ($k = 0, $in2 = count($id_in_proses); $k < $in2; $k++) { } 13
21        $a_to_s = $this->jarak_routing($arr[$id_in_hasil[$j]]['value'], $arr[$id_in_proses[$k]]['value']);
22        $a_to_t = $this->jarak_routing($arr[$id_in_proses[$k]]['value'], $arr[$id_in_hasil[$j + 1]]['value']);
23        $a_to_u = $this->jarak_routing($arr[$id_in_hasil[$j]]['value'], $arr[$id_in_hasil[$j + 1]]['value']);
24        echo $id_in_hasil[$j] . " | " . $id_in_proses[$k] . " | " . $id_in_hasil[$j + 1] . " | " .
25          ($a_to_s + $a_to_t - $a_to_u);
26
27      if ($sterkecil == "") { } 15
28        $tmp = explode(":", $sterkecil); } 16
29        if ($tmp[1] > ($a_to_s + $a_to_t - $a_to_u)) { } 17
30          $sterkecil = $id_in_hasil[$j] . ":" . $id_in_proses[$k] . ":" . $id_in_hasil[$j + 1] . ":" . $tmp[1]; } 18
31        }
32      } else {
33        $sterkecil = $id_in_hasil[$j] . ":" . $id_in_proses[$k] . ":" . $id_in_hasil[$j + 1] . ":" . $tmp[1]; } 19
34      }
35      echo "\n";
36    }
37
38    if ($i == ($in - 1)) { } 20
39      echo "\n";
40      for ($k = 0, $in2 = count($id_in_proses); $k < $in2; $k++) { } 21
41        $a_to_s = $this->jarak_routing($arr[$id_in_hasil[$in - 1]]['value'], $arr[$id_in_proses[$k]]['value']);
42        $a_to_t = $this->jarak_routing($arr[$id_in_proses[$k]]['value'], $arr[$id_in_hasil[0]]['value']);
43        $a_to_u = $this->jarak_routing($arr[$id_in_hasil[$in - 1]]['value'], $arr[$id_in_hasil[0]]['value']);
44        echo $id_in_hasil[$in - 1] . " | " . $id_in_proses[$k] . " | " . $id_in_hasil[0] . " | " .
45          ($a_to_s + $a_to_t - $a_to_u);
46        $tmp = explode(":", $sterkecil);
47        if ($tmp[1] > ($a_to_s + $a_to_t - $a_to_u)) { } 23
48          $sterkecil = $id_in_hasil[$in - 1] . ":" . $id_in_proses[$k] . ":" . $id_in_hasil[0] . ":" . $tmp[1]; } 24
49        }
50      echo "\n";
51
52  }
53  echo "\n";
54
55  $tmp1 = explode(":", $sterkecil);
56  $tmp2 = explode(":", $tmp1[0]);
57  $tmp_proses = array();
58  for ($i = 0, $in3 = count($id_in_hasil); $i < $in3; $i++) { } 26
59    array_push($tmp_proses, $id_in_hasil[$i]); } 27
60    for ($n = 0, $in4 = count($tmp2); $n < ($in4 - 1); $n++) { } 28
61      if ($id_in_hasil[$i] == $tmp2[$n]) { } 29 } 30
62      array_push($tmp_proses, $tmp2[$n + 1]); }
63    }
64  }
65  $id_in_hasil = $tmp_proses; } 31
66
67  $tmp_hasil = array(); } 32
68  for ($n = 0, $in5 = count($id_in_hasil); $n < $in5; $n++) { } 33
69    array_push($tmp_hasil, $arr[$id_in_hasil[$n]]['value']); } 34
70  }
71  $c_hasil = $tmp_hasil; } 35
72
73  echo "c_hasil : ";
74  print_r($c_hasil);
75  echo "\n";
76  echo "id_hasil : ";
77  print_r($id_in_hasil);

```

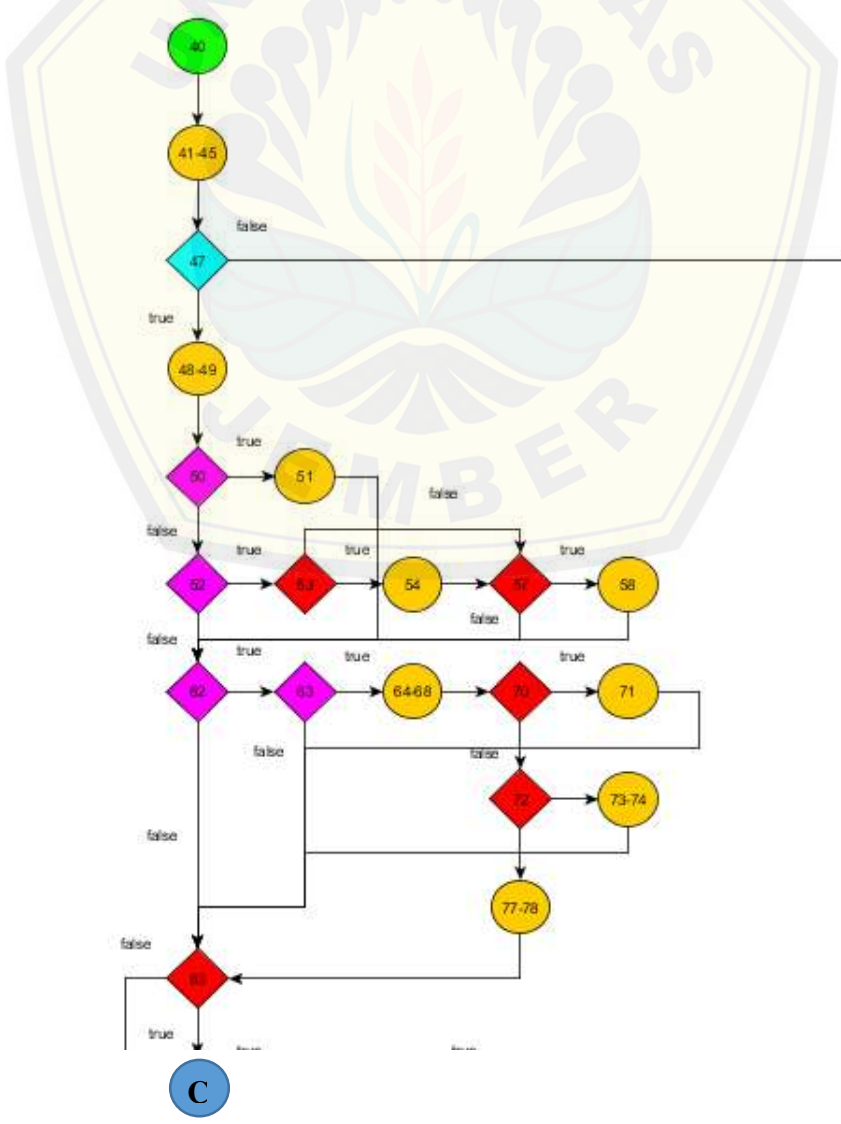
```

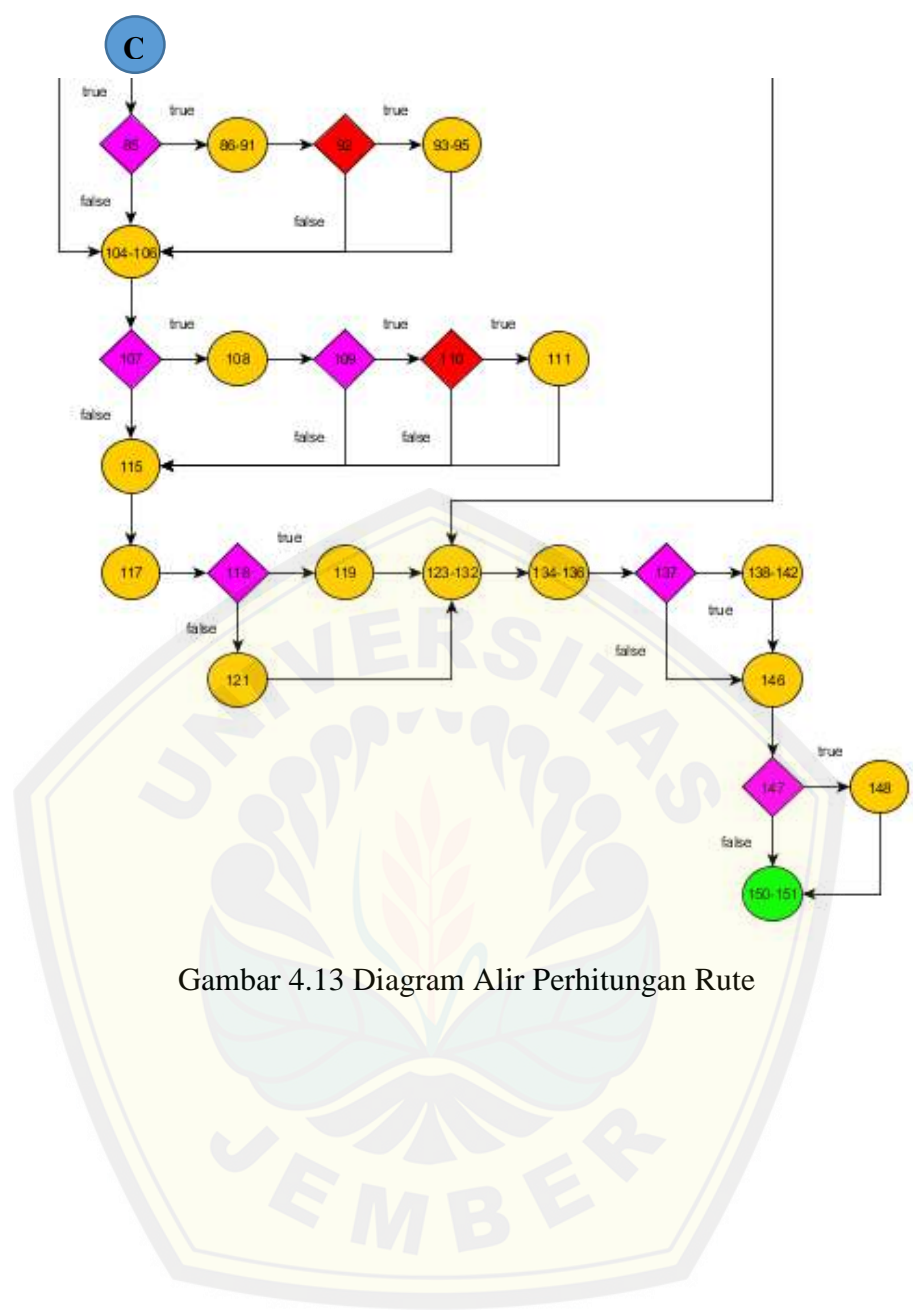
128 echo `chr /chr />` } 36
129 echo `id progres : `
130 print_r($id_in_progres)
131 echo `chr /chr />`
132 echo `Rute tercepat : ` , $r_hasil[0]
133
134 $ruiter = 1; } 37
135 $jumlah = 0;
136 $o = 0;
137 for ($i = 1; $i < count($r_hasil); $i++) { } 38
138 echo `--> ` , $r_hasil[$i];
139 $power++;
140 $jumlah = $jumlah + $this->jarak_routing($r_hasil[0], $r_hasil[$i]); } 39
141 echo `--> ` , $jumlah;
142 $o++;
143 }
144 echo `chr /chr />`
145
146 $rute_perjalanan = array(); } 40
147 for ($i = 0; $i < count($r_hasil); $i++) { } 41
148 array_push($rute_perjalanan, $this->view_rhasil_combo("$b_ain", $r_hasil[$i])); } 42
149 }
150 $this->session->set_userdata("rute_perjalanan", $rute_perjalanan); } 43
151 $this->session->set_userdata("total_jarak", $jumlah);
152 }

```

Gambar 4.12 Listing Program Perhitungan Rute

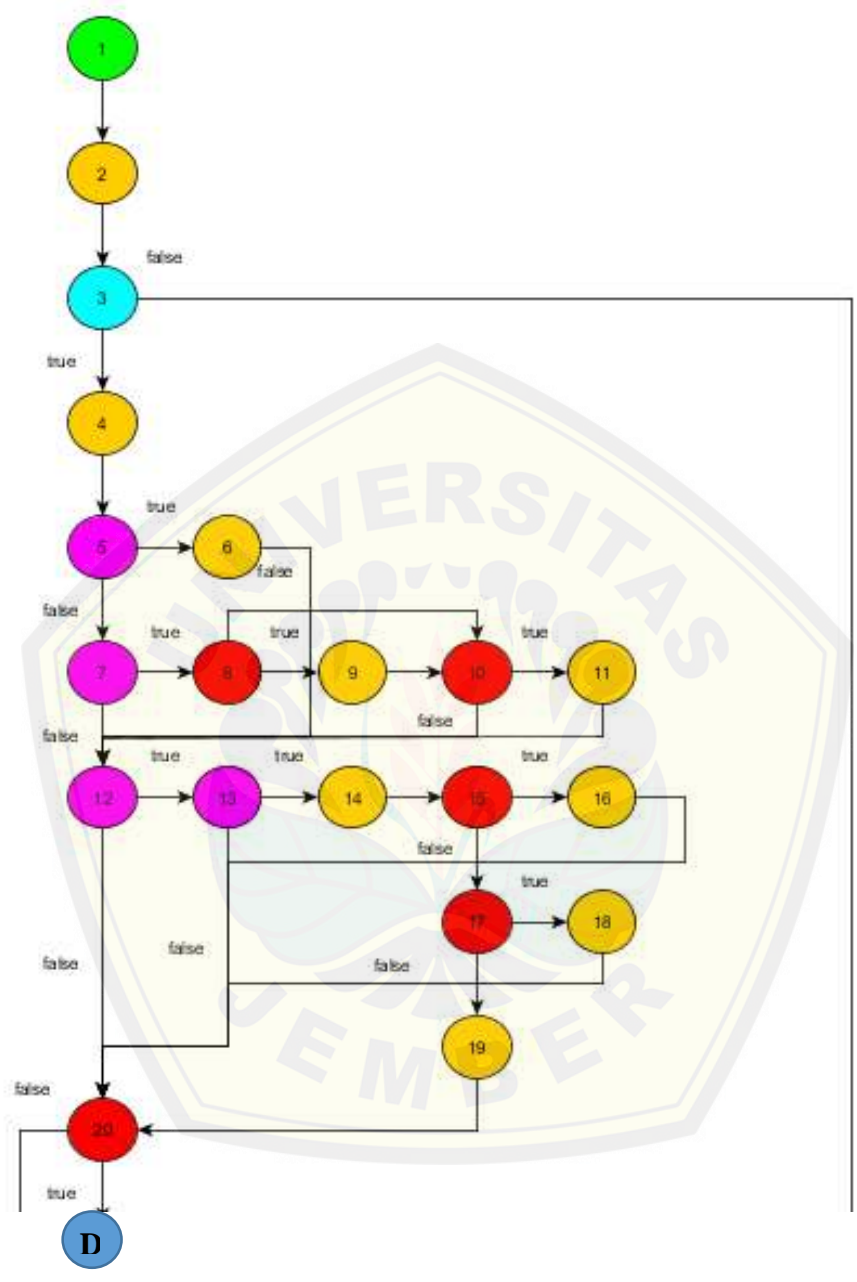
(b). Diagram Alir Fitur Perhitungan Rute

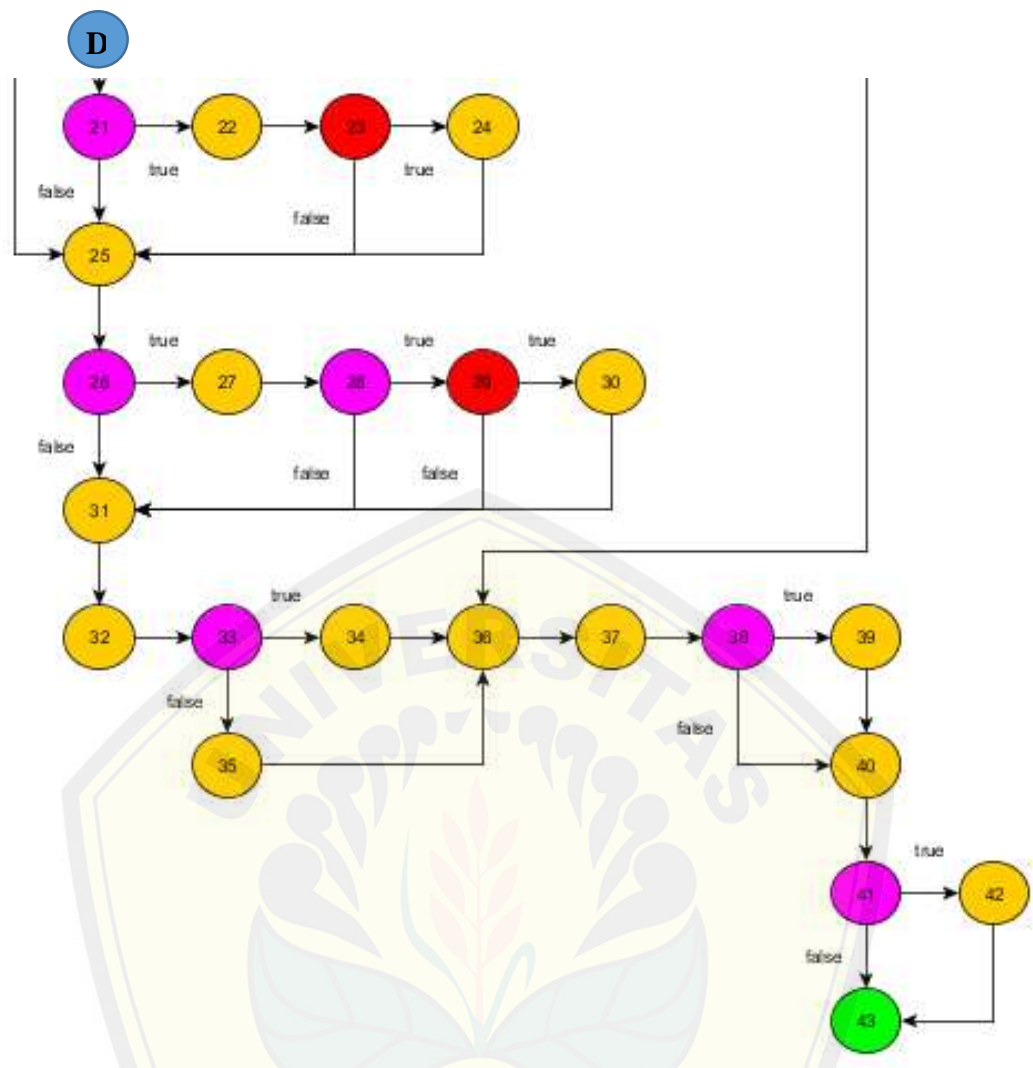




Gambar 4.13 Diagram Alir Perhitungan Rute

(c). Grafik Alir Fitur Perhitungan Rute





Gambar 4.14 Grafik Alir Perhitungan Rute

(d). Kompleksitas Siklometik

Tahapan pengujian ini menggunakan rumus $V(G) = E - N + 2$, dimana E adalah jumlah *edge* dan N adalah jumlah *node* grafik alir.

Function haber() $V(G) = E - N + 2 = 61 - 43 + 2 = 20$

(e). Pengujian Jalur Program Fitur Perhitungan Rute

Pengujian jalur program fitur Perhitungan Rute berdasarkan gambar di atas adalah sebagai berikut:

4) *Function* haber() :

Jalur 1 : 1 – 2 – 3 – 4 – 5 – 6 – 12 – 13 – 14 – 15 – 16 – 20 – 21 –
22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 36 –
37 – 38 – 39 – 40 – 41 – 42 – 43

Jalur 2 : 1 – 2 – 3 – 36 – 37 – 38 – 39 – 40 – 41 – 43

(f). *Test Case* Fitur Manajemen Data ATM

Tabel 4.6 *Test Case* Fitur Manajemen Data ATM

<i>Test Case function haber ()</i>	
Jalur 1	
<i>Test Case</i>	Jika melakukan pengecekan pada variable hasil dengan jumlah inputan ATM memenuhi syarat untuk melakukan pencarian nilai yang disisipkan
Target yang diharapkan	Melakukan perhitungan untuk mencari nilai yang akan disisipkan untuk masuk pada variabel proses yang selanjutnya akan diinputkan nilai awal, nilai sisip, nilai tujuan pada penyimpanan temporary dan selanjutnya akan dilakukan perhitungan untuk pencarian jarak terkecil dengan menggunakan rumus yang ada pada algoritma <i>cheapest insertion heuristic</i> , setelah ditemukan nilai terkecil maka tahapan selanjutnya adalah membalik subtour dengan nilai awal dan tujuan menjadi nilai tujuan menuju nilai awal. Tahapan selanjutnya adalah membandingkan nilai terkecil, dan setelah menemukan maka nilai tersebut dimasukan ke dalam penyimpanan temporary hasil.

Berlanjut

Lanjutan

Jalur 1

	Tahap akhir adalah mencetak nilai hasil dan nilai proses, dan menghitung total jarak dari perhitungan hasil akhir subtour yang telah didapatkan pada penyimpanan temporary hasil akhir.
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 4 – 5 – 6 – 12 – 13 – 14 – 15 – 16 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43
Jalur 2	
<i>Test Case</i>	Jika melakukan pengecekan pada variable hasil dengan jumlah inputan ATM tidak memenuhi syarat untuk melakukan pencarian nilai yang disisipkan
Target yang diharapkan	Mencetak nilai hasil dan nilai proses, dan menghitung total jarak dari perhitungan hasil akhir subtour yang telah didapatkan pada penyimpanan temporary hasil akhir.
Hasil Pengujian	Benar
Path/Jalur	1 – 2 – 3 – 36 – 37 – 38 – 39 – 40 – 41 – 43

4.5.2 Black Box Testing

Tahapan pengujian *black box* dilakukan untuk menguji apakah kebutuhan yang dibutuhkan oleh *user* atau kebutuhan yang tertera pada kebutuhan fungsional sudah sesuai atau tidak, sehingga pengujian ini dilakukan pada *form* untuk setiap *usecase* yang dilakukan oleh bagian *IT & E-Channel* bank BRI cabang Jember yang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Hasil Pengujian *Black Box* Perhitungan Rute

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Perhitungan Rute	Memilih menu “Perhitungan Rute”	a) Menampilkan halaman <i>Perhitungan Rute</i>	[√] Berhasil [] Gagal
2.	Input Jumlah ATM	a) Menginputkan jumlah ATM pada inputan yang akan di routing b) Klik <i>button</i> “ <i>add</i> ”	a) Menampilkan dropdown ATM sebanyak jumlah yang telah di inputkan	[√] Berhasil [] Gagal
		a) Klik <i>button</i> “ <i>add</i> ” ketika input jumlah ATM tidak terisi	a) Menampilkan halaman Routing ATM dan <i>alert</i> “Maaf! Isi data dengan lengkap”	[√] Berhasil [] Gagal
3.	Submit Jumlah ATM	a) Memilih ATM pada <i>dropdown</i> hingga semua ATM yang akan di routing telah di pilih	a) Sistem melakukan perhitungan dengan metode <i>Cheapest Insertion Heuristic</i>	[√] Berhasil [] Gagal

Berlanjut

Lanjutan

		b) Klik <i>button</i> "submit"	b) Mengambil titik awal dan titik akhir ATM yang telah dipilih c) Menyisipkan ATM lain. Dan melakukan perhitungan penjumlahan jarak dengan menggunakan rumus $[C_{ik}+C_{kj}-C_{ij}]$ d) Memilih jarak terkecil untuk dijadikan sebagai rute yang baru e) Menyisipkan ATM lain yang tersisa pada rute yang baru dan menghitung total jarak dengan rumus $[C_{ik}+C_{kj}-C_{ij}]$ f) Dilakukan berulang kali hingga seluruh ATM telah masuk pada rute	
--	--	--------------------------------	--	--

Berlanjut

Lanjutan

			<p>g) Memilih total jarak terkecil dari opsi penyisipan seluruh rute</p> <p>h) Sistem menampilkan halaman Hasil Routing dengan Peta dan keterangan langkah perjalanan yang akan ditempuh dengan alamat dan total jarak keseluruhan</p>	
		Klik <i>button</i> “ <i>Submit</i> ” ketika dropdown tidak terisi	<p>a) Sistem menampilkan <i>alert</i> “<i>Please select an item in the list</i>”</p>	<p><input checked="" type="checkbox"/> Berhasil</p> <p><input type="checkbox"/> Gagal</p>
4.	Reset Jumlah ATM	<p>a) Memilih ATM pada <i>dropdown</i> hingga semua ATM yang akan di routing telah di pilih</p> <p>b) Klik <i>button</i> “<i>Reset</i>”</p>	<p>a) Menampilkan halaman <i>Routing</i> ATM dengan <i>dropdown</i> kembali tidak tersisi</p>	<p><input checked="" type="checkbox"/> Berhasil</p> <p><input type="checkbox"/> Gagal</p>

Tabel 4.5 adalah hasil pengujian yang dilakukan pada *usecase* perhitungan rute. Hasil pengujian yang dilakukan oleh *user* membuktikan bahwa hasil program perhitungan rute telah sesuai dengan kebutuhan yang diinginkan oleh *user*. Pengujian selanjutnya dapat dilihat pada lampiran.

BAB 5. HASIL DAN PEMBAHASAN

Hasil dan Pembahasan adalah tahapan dimana akan menggambarkan hasil dari pembangunan sistem Pendistribusian Pengisian ATM dengan menggunakan Algoritma *CIH* dan juga pembahasannya. Penjabaran hasil dan pembahasan ini nantinya akan bertujuan untuk menjelaskan bagaimana peneliti menyelesaikan perumusan masalah serta tujuan dan manfaat dari penelitian tentang pencarian optimasi jalur terpendek.

5.1 Hasil Penerapan Algoritma *Cheapest Insertion Heuristic* pada Pencarian Optimasi Rute

Hasil Penerapan Algoritma *CIH* pada pencarian optimasi rute bertujuan untuk memberikan kemudahan kepada PT. BRI (Persero) Tbk Cabang Jember, khususnya bagian *IT & E-Channel* dan bagian pengisian ATM sebagai *user* untuk melakukan pencarian jalur terpendek atau optimasi rute untuk melakukan proses pendistribusian pengisian ATM.

Penerapan dengan menggunakan Algoritma *CIH* pada penelitian ini dilakukan untuk menganalisa dan menyelesaikan permasalahan tentang optimasi rute untuk pendistribusian pengisian ATM. Peneliti melakukan percobaan dengan manual apakah perhitungan dengan algoritma ini layak atau tidak sebelum peneliti menggunakan dalam pembuatan sistem, yang perhitungannya dapat dilihat pada Metodologi Penelitian.

Hasil dari percobaan manual dengan menggunakan algoritma *CIH* yang dapat dilihat pada pengerjaan sebelumnya menunjukkan bahwa algoritma ini layak atau cocok digunakan dalam pembangunan aplikasi pendistribusian pengisian ATM.

Kesimpulan yang dihasilkan dengan menggunakan algoritma *CIH* pada proses pendistribusian pengisian ATM bahwa metode ini merupakan metode yang baik dan dapat melakukan pencarian rute perjalanan dengan optimal dengan menggunakan waktu pengerjaan yang relatif cepat. Berdasarkan kesimpulan tersebut maka, peneliti menyatakan bahwa algoritma ini layak untuk menjadi algoritma dalam pembangunan sistem pendistribusian pengisian ATM untuk pencarian jalur terpendek.

5.2 Hasil Pembuatan Aplikasi Pendistribusian Pengisian ATM

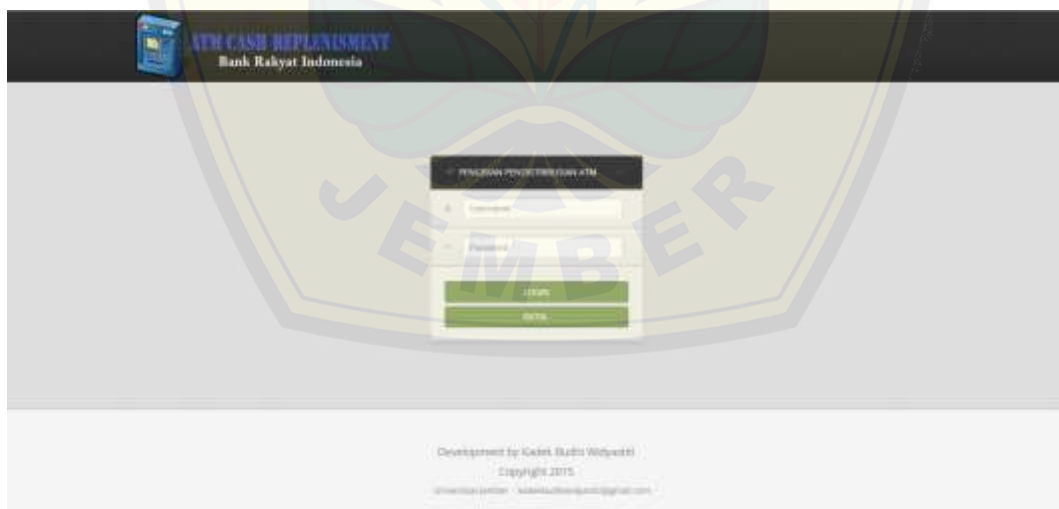
Pembuatan aplikasi ini dibuat dengan menggunakan 2 hak akses, yakni sebagai *user* yakni bagian yang bertugas melakukan pendistribusian pengisian ATM dan admin yakni bagian *IT & E Channel* dari Bank BRI Cabang Jember. Admin disini dapat mengakses fitur

untuk mengelola data ATM, yakni menambahkan ATM yang akan baru dibuat, mengedit ATM yang sudah ada dan Menghapus ATM yang tidak di pakai kembali, fitur untuk melakukan pencarian jarak yang dibutuhkan untuk proses pencarian jalur dengan cara menghubungkan 2 ATM yang kita inginkan untuk mencari jaraknya. Admin juga dapat melakukan perhitungan pencarian rute yang terletak pada halaman *user*, dan juga fitur tambahan yakni fitur *login*, fitur *logout*, fitur melihat index sistem yakni melihat peta ATM BRI yang ada di seluruh Kabupaten Jember.

User disini memiliki fitur utama yakni melakukan perhitungan untuk mencari optimasi rute untuk melakukan pendistribusian pengisian ATM. Fitur tambahan untuk *user* antara lain fitur melihat data Bank BRI yakni melihat visi & misi dan sejarah dari Bank BRI, fitur untuk melihat index user atau melihat peta ATM Bank BRI yang ada di Kabupaten Jember. Fitur untuk admin dan *user* lebih jelasnya akan digambarkan pada penjelasan di bawah ini.

5.2.1 Fitur *Login*

Aplikasi pengisian pendistribusian ATM dibuat untuk admin agar dapat masuk menuju halaman admin dengan *login* terlebih dahulu. Fitur ini dibuat agar tidak semua *user* kecuali admin yang dapat mengakses halaman ini. Fitur keamanan dengan halaman login untuk masuk memiliki *username* admin dan *password* admin, yang dapat dilihat pada Gambar 5.1.

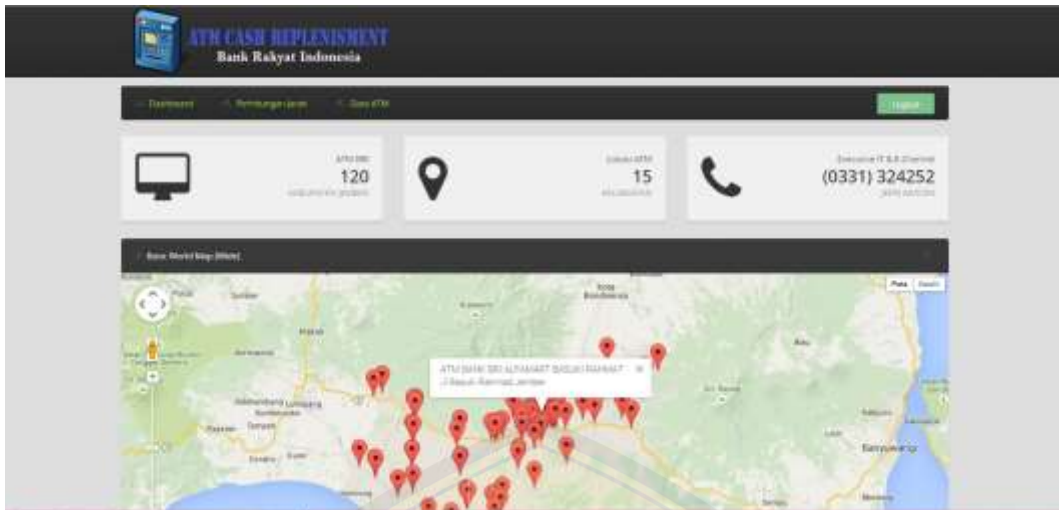


Gambar 5.1 Tampilan Halaman *Login*

5.2.2 Halaman Melihat Peta ATM Admin

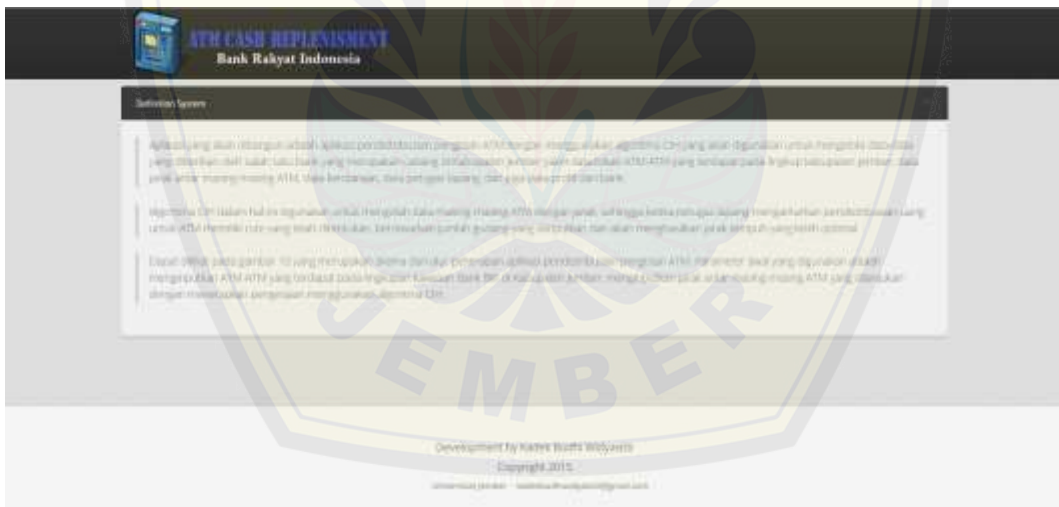
Admin yang berhasil masuk menuju halaman login, tampilan pertama yang keluar pada halaman admin adalah halaman melihat peta admin. Halaman ini berisi tentang peta ATM BRI

yang tersedia di seluruh Kabupaten Jember dan ketika *icon* ATM dipilih maka akan terdapat keterangan yang menjelaskan ATM tersebut, yang dapat dilihat pada Gambar 5.2.



Gambar 5.2 Tampilan Halaman Melihat Peta ATM Admin

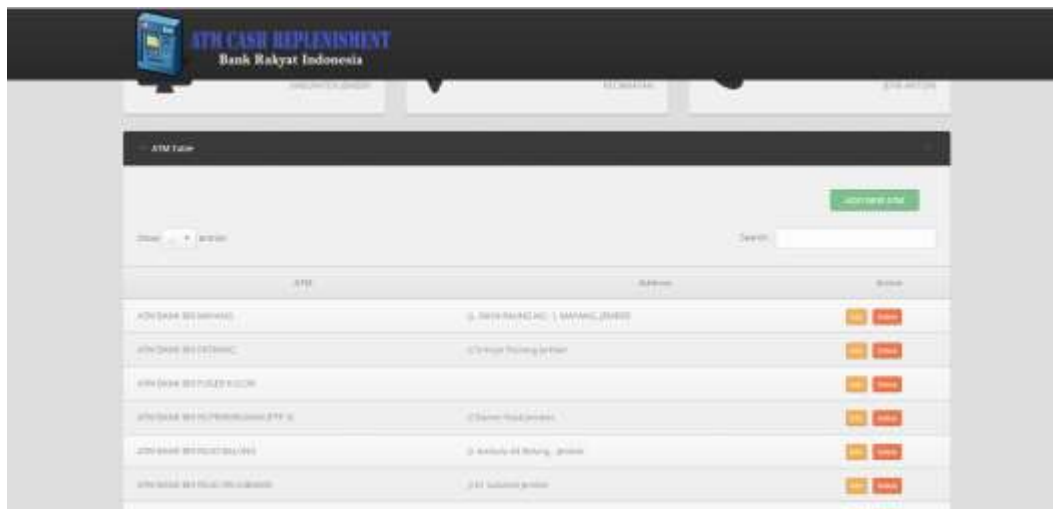
Admin juga dapat melihat tampilan keterangan tentang aplikasi yang telah dibangun ini, yang dapat dilihat pada Gambar 5.3.



Gambar 5.3 Tampilan Halaman Melihat Peta ATM Admin pada Keterangan

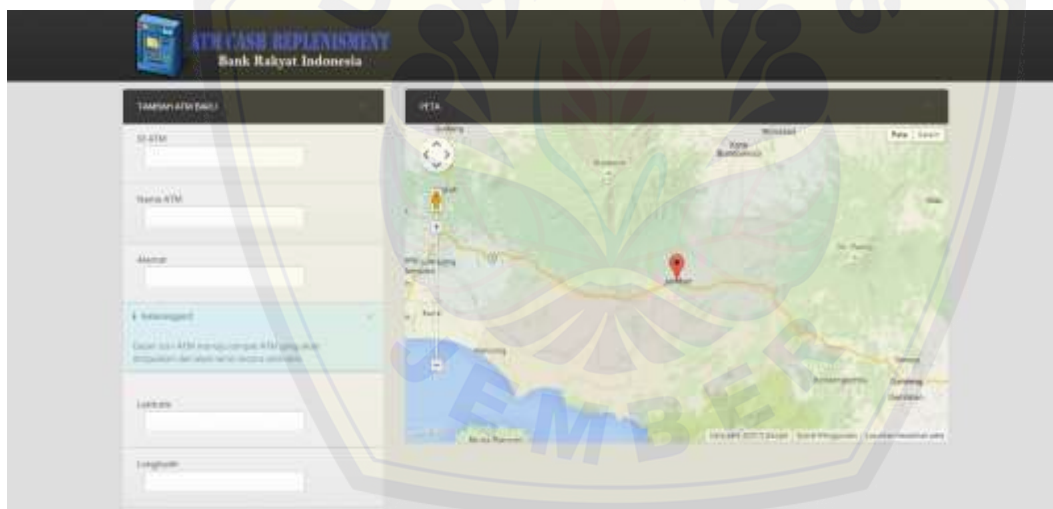
5.2.3 Fitur Manajemen Data ATM

Admin juga dapat mengelola data ATM yakni menambahkan data ATM baru, mengubah data ATM yang telah ada, dan menghapus data ATM. Juga terdapat fitur untuk melihat jumlah ATM yang kita inginkan dengan memilih pada dropdown show, dan fitur untuk memudahkan pencarian ATM dengan memasukan pada textfield search. Fitur manajemen data ATM dapat dilihat pada Gambar 5.4.



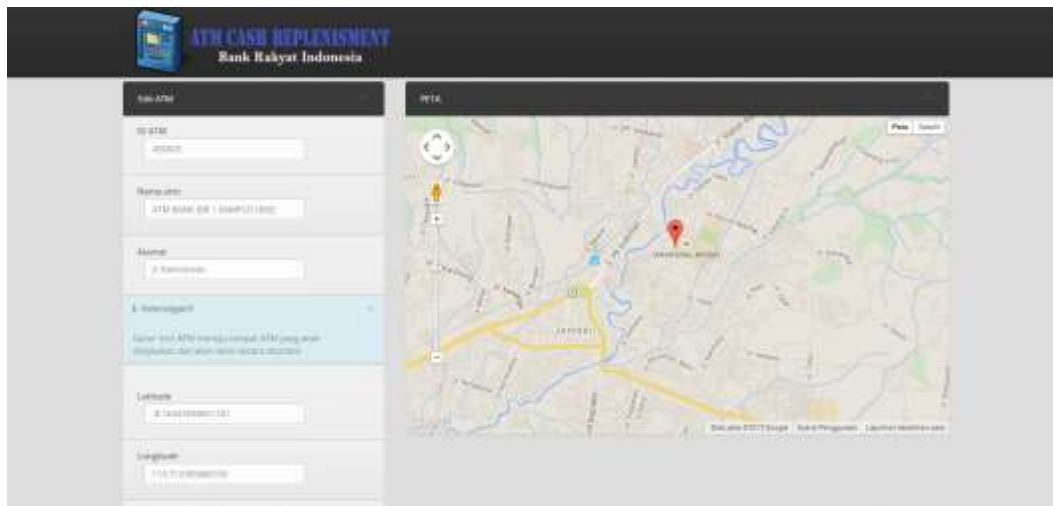
Gambar 5.4 Tampilan Halaman Mengelola Data ATM

Admin menambahkan data ATM dengan menginputkan data pada textfield yang masih kosong dan drag pada icon ATM pada peta. Fitur menambahkan data ATM dapat dilihat pada Gambar 5.5.



Gambar 5.5 Tampilan Halaman Menambahkan Data ATM

Proses mengedit data hanya membutuhkan untuk admin menggantikan nama ATM dan alamat ATM yang akan diperbarui. Fitur mengedit data ATM dapat dilihat pada Gambar 5.6.



Gambar 5.6 Tampilan Halaman Mengedit data ATM

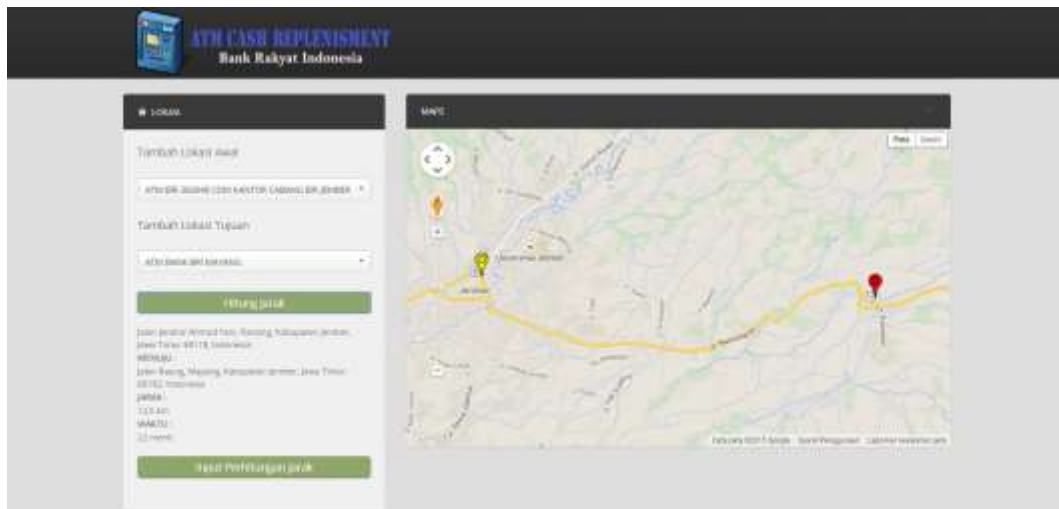
Proses penghapusan data, admin hanya perlu untuk menekan button delete dan data akan terhapus. Fitur manajemen data ATM dapat dilihat pada Gambar 5.7.



Gambar 5.7 Tampilan Halaman Menghapus Data ATM

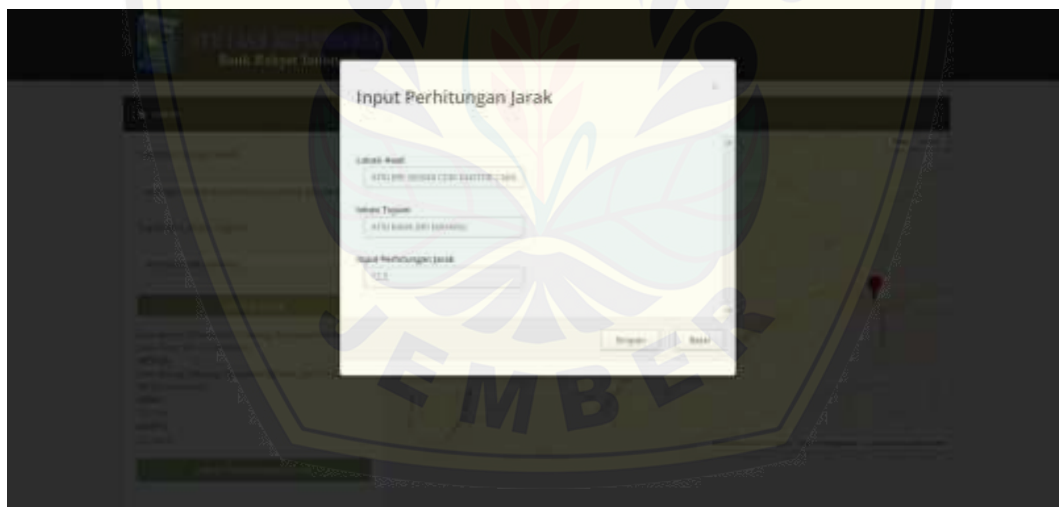
5.2.4 Fitur Perhitungan Jarak

Admin dapat mencari jarak antar 2 ATM dengan menggunakan fitur perhitungan jarak. Proses menginputkan 2 ATM yang ingin dicari jaraknya, maka dengan otomatis jarak akan tampil. Penggambaran fitur perhitungan jarak dapat dilihat pada Gambar 5.8.



Gambar 5.8 Tampilan Halaman Perhitungan Jarak

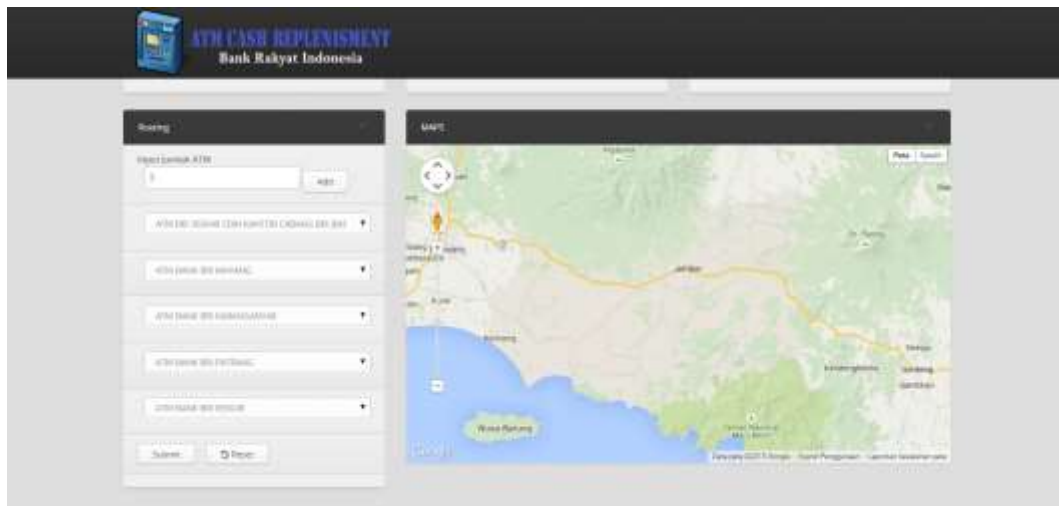
Proses penambahan jarak antar 2 ATM selanjutnya dengan ditampilkan modal yang berisi data yang akan diinputkan, selanjutnya data akan disimpan ketika sesuai atau belum pernah diinputkan, dan akan berhasil ketika belum pernah diinputkan, yang dapat dilihat pada Gambar 5.9.



Gambar 5.9 Tampilan Halaman Modal Perhitungan Jarak

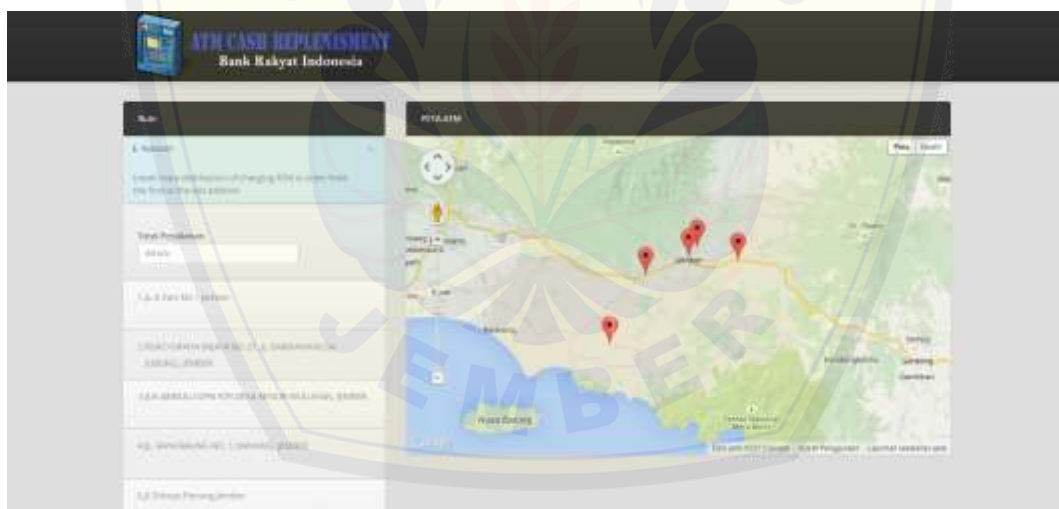
5.2.5 Fitur Perhitungan Rute

Fitur perhitungan rute merupakan fitur yang terdapat pada halaman user, namun admin juga dapat mengakses. Fitur ini adalah fitur yang dibuat untuk melakukan perhitungan untuk mendapatkan hasil dari rute yang nantinya akan digunakan untuk melakukan pendistribusian pengisian ATM, yang dapat dilihat pada Gambar 5.10.



Gambar 5.10 Tampilan Halaman Perhitungan Route

Tahapan setelah mengisikan ATM mana saja yang akan dituju, selanjutnya setelah melakukan proses perhitungan maka akan tampil halaman hasil yang berisi total jarak keseluruhan pada rute, alamat mana saja yang akan dikunjungi, dan tampilan ATM pada peta. Tampilan dari halaman fitur hasil perhitungan rute dapat dilihat pada Gambar 5.11.



Gambar 5.11 Tampilan Halaman Hasil Perhitungan Route

Fitur tambahan selain di atas seperti fitur logout, fitur melihat peta ATM user, fitur melihat data BRI telah diuraikan dan dapat dilihat pada lampiran F.

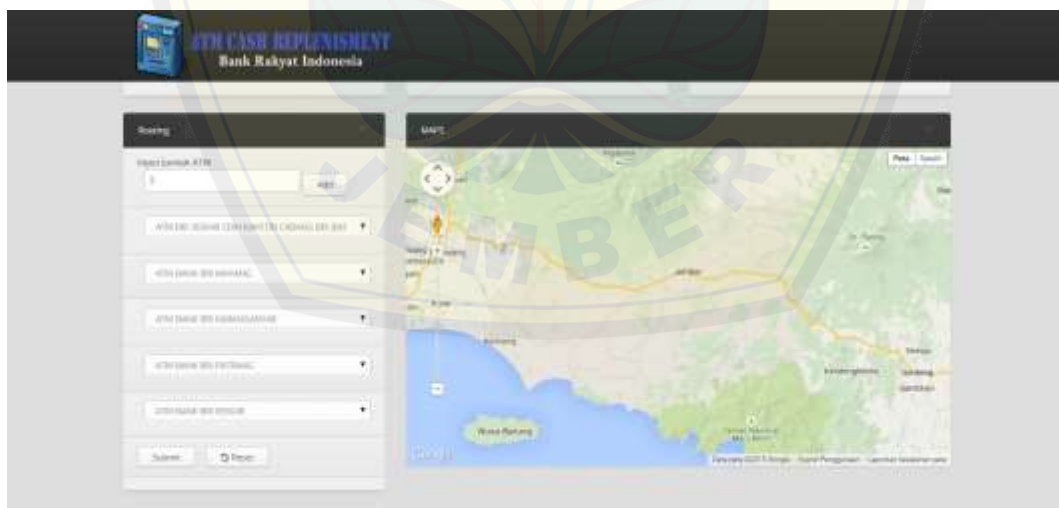
5.3 Hasil Penerapan Algoritma *CIH* Pada Aplikasi

Penerapan algoritma *CIH* pada aplikasi pendistribusian pengisian ATM terletak pada fitur perhitungan rute yang terdapat pada halaman *user*. Tahap pertama penerapan metode ini

adalah yang pertama melakukan penginputan jumlah ATM yang akan menjadi objek untuk pendistribusian pengisian ATM, selanjutnya adalah pengecekan jumlah ATM untuk dijadikan nilai awal, nilai akhir dan perbandingan untuk nilai sisip, lalu memasukan nilai sisip dalam penyimpanan proses *temporary*, selanjutnya adalah perhitungan tabel proses dengan pengulangan dan perhitungan nilai terkecil, membalikkan nilai awal ke tujuan menjadi nilai tujuan ke awal dan melakukan proses perhitungan kembali untuk mendapatkan nilai terkecil, lalu memilih nilai terkecil dan memasukan ke dalam penyimpanan *temporary* hasil menjadi sebuah *subtour* baru, selanjutnya mengulangi tahap pertama hingga semua ATM telah berada pada *subtour*, dan jika semua ATM telah masuk pada *subtour* maka akan melakukan mencetak nilai yang ada pada penyimpanan *temporary* hasil, penyimpanan *temporary* proses, dan tahap akhir adalah menghitung total jarak dari hasil *routing* yang didapatkan.

5.3.1 Penginputan ATM yang akan menjadi objek

Tahap awal adalah memasukan berapa jumlah ATM yang akan dijadikan objek untuk melakukan perhitungan rute, selanjutnya pemilihan ATM mana saja yang telah disediakan pada *dropdown*. Pemilihan selesai, semua ATM masuk maka menekan *button submit* untuk melakukan perhitungan. Pemilihan ATM sebagai objek perhitungan dapat dilihat pada Gambar 5.12.



Gambar 5.12 Penginputan ATM yang akan menjadi Objek

5.3.2 Pengecekan nilai awal, nilai tujuan, dan nilai sisip

Tahap ini adalah tahap dimana melakukan proses pengecekan, apakah jumlah ATM yang diinputkan telah masuk semua, jika tidak maka akan melakukan proses atau tahapan selanjutnya. Proses pengecekan dapat dilihat pada Tabel 5.1.

Tabel 5.1 Kode Program Pengecekan ATM

```
public function haber() {
    $t_hasil = array($atm[0]['value'], $atm[(count($atm) - 1)]['value']);
    $id_in_hasil = array(0,(count($atm) - 1));

    while (count($t_hasil) < count($atm)) {
```

5.3.3 Input nilai sisip ke dalam penyimpanan temporary proses

Tahapan selanjutnya ketika masih ada nilai ATM, maka melakukan proses pengulangan untuk mengetahui ATM mana saja yang belum masuk ke dalam perhitungan yang selanjutnya dimasukan ke dalam penyimpanan temporary proses. Tahapan ini dapat dilihat pada Tabel 5.2.

Tabel 5.2 Kode Program Input Nilai Sisip

```
$sterkecil = "";
$id_in_proses = array();
for ($i = 0, $ln = count($atm); $i < $ln; $i++) {
    $sada = FALSE;
    for ($j = 0, $ln2 = count($t_hasil); $j < $ln2; $j++) {
        if ($t_hasil[$j] == $atm[$i]['value']) {
            $sada = TRUE;
        }
    }
    if (!$sada) {
        array_push($id_in_proses, $i);
    }
}
```

5.3.4 Perhitungan nilai awal, nilai sisip, nilai tujuan dan nilai terkecil

Nilai sisip yang dimasukan ke dalam penyimpanan *temporary* proses selanjutnya akan dihitung dengan nilai awal, nilai tujuan, dan juga pencarian nilai terkecil dengan menggunakan rumus yang telah ada, dan dilakukan proses pengulangan hingga semua nilai sisip telah masuk. Perhitungan ini dapat dilihat pada Tabel 5.3.

Tabel 5.3 Kode Program Perhitungan Nilai Awal, Nilai Sisip, Nilai Tujuan, Nilai Terkecil

```
for ($j = 0, $ln = count($id_in_hasil); $j < ($ln - 1); $j++) {
    for ($k = 0, $ln2 = count($id_in_proses); $k < $ln2; $k++) {
        $a_to_s = $this->jarak_routing($atm[$id_in_hasil[$j]]['value'], $atm[$id_in_proses[$k]]['value']);
        $s_to_t = $this->jarak_routing($atm[$id_in_proses[$k]]['value'], $atm[$id_in_hasil[( $j + 1)]]['value']);
```

```

$a_to_t = $this->jarak_routing($atm[$id_in_hasil[$j]]['value'], $atm[$id_in_hasil[( $j + 1)]]['value']);
echo $id_in_hasil[$j] . '|' . $id_in_proses[$k] . '|' . $id_in_hasil[( $j + 1) ] . '|'
. ($a_to_s + $s_to_t - $a_to_t);

// meletakkan nilai terkecil
if ($sterkecil != "") {
    $tmp = explode(";", $sterkecil);
    if ($tmp[1] > ($a_to_s + $s_to_t - $a_to_t)) {
        $sterkecil = $id_in_hasil[$j] . ';ID;' . $id_in_proses[$k] . ';ID;' . $id_in_hasil[( $j + 1) ] . ';H;'
. ($a_to_s + $s_to_t - $a_to_t);
    }
} else {
    $sterkecil = $id_in_hasil[$j] . ';ID;' . $id_in_proses[$k] . ';ID;' . $id_in_hasil[( $j + 1) ] . ';H;'
. ($a_to_s + $s_to_t - $a_to_t);
}

```

5.3.5 Membalikan nilai awal ke tujuan menjadi tujuan ke awal

Perhitungan dengan tujuan dari nilai awal ke nilai tujuan, dilanjutkan dengan membalikan tujuan dari nilai tujuan ke nilai awal. Perhitungan pun dilanjutkan dengan memasukan nilai sisip dan pencarian nilai terkecil, hingga seluruh nilai sisip masuk ke dalam proses perhitungan. Tahapan ini dapat dilihat pada Tabel 5.4.

Tabel 5.4 Kode Program Membalik Arah Tujuan

```

if ($j == ($ln - 2)) {
    echo '<br />';
    for ($k = 0, $ln2 = count($id_in_proses); $k < $ln2; $k++) {
        $a_to_s = $this->jarak_routing($atm[$id_in_hasil[( $ln - 1)]]['value'], $atm[$id_in_proses[$k]]['value']);
        $s_to_t = $this->jarak_routing($atm[$id_in_proses[$k]]['value'], $atm[$id_in_hasil[0]]['value']);
        $a_to_t = $this->jarak_routing($atm[$id_in_hasil[( $ln - 1)]]['value'], $atm[$id_in_hasil[0]]['value']);
        echo $id_in_hasil[( $ln - 1) ] . '|' . $id_in_proses[$k] . '|' . $id_in_hasil[0] . '|'
. ($a_to_s + $s_to_t - $a_to_t);
    }
}

```

5.3.6 Pemilihan nilai terkecil

Tahapan selanjutnya adalah ketika kita berhasil mendapatkan nilai terkecil, maka kita akan menyimpan dalam penyimpanan *temporary* hasil untuk memperbarui data sebelumnya. Kode program dapat dilihat pada Tabel 5.5.

Tabel 5.5 Kode Program Pemilihan Nilai Terkecil

```

$tmp1 = explode(";", $sterkecil);

```

```

$tmp2 = explode(";ID;", $tmp1[0]);
$tmp_proses = array();
for ($l = 0, $ln3 = count($id_in_hasil); $l < $ln3; $l++) {
    array_push($tmp_proses, $id_in_hasil[$l]);
    for ($m = 0, $ln4 = count($tmp2); $m < ($ln4 - 1); $m++) {
        if ($id_in_hasil[$l] == $tmp2[$m]) {
            array_push($tmp_proses, $tmp2[($m + 1)]);
        }
    }
}
}

```

5.3.7 Pembentukan *subtour* baru

Langkah berikutnya setelah memasukan nilai yang baru kepada penyimpanan *temporary* hasil, selanjutnya nilai yang berada di dalam penyimpanan tersebut akan menjadi sebuah *subtour* baru dengan pemisahan awal dan tujuan dengan menggunakan *array*. Kode program perhitungan ini dapat dilihat dalam Tabel 5.6.

Tabel 5.6 Kode Program Pembentukan *Subtour* Baru

```

$id_in_hasil = $tmp_proses;

$tmp_hasil = array();
for ($n = 0, $ln5 = count($id_in_hasil); $n < $ln5; $n++) {
    array_push($tmp_hasil, $atm[$id_in_hasil[$n]]['value']);
}
$_hasil = $tmp_hasil;
}

```

5.3.8 Mencetak nilai penyimpanan *temporary* hasil, penyimpanan *temporary* proses

Setelah semua perhitungan telah selesai maka proses akan kembali kepada tahapan pengecekan. Pengecekan terhadap nilai sisip, jika nilai sisip masih ada yang belum masuk ke dalam penyimpanan *temporary* hasil, maka proses akan berulang kembali, tetapi jika semua nilai sisip telah masuk ke dalam penyimpanan *temporary* hasil, maka nilai yang berada pada penyimpanan *temporary* hasil dan penyimpanan *temporary* proses akan dicetak. Kode programnya dapat dilihat pada Tabel 5.7.

Tabel 5.7 Kode Program Mencetak Nilai Penyimpanan *Temporary*

```

echo 't_hasil : '; print_r($_hasil);
echo '<br /><br />';

```



```

echo 'id_hasil : '; print_r($id_in_hasil);
echo '<br /><br />';
echo 'id_progres : '; print_r($id_in_proses);
echo '<br /><br />';
echo 'Rute tercepat : ' . $_hasil[0];

```

5.3.9 Perhitungan total jarak seluruh rute

Tahap akhir adalah melakukan perhitungan kembali pada penyimpanan *temporary* hasil untuk mendapatkan total jarak dari seluruh *subtour* atau rute yang telah dibentuk. Dapat dilihat pada kode program pada Tabel 5.8.

Tabel 5.8 Kode Program Perhitungan Total Jarak seluruh Rute

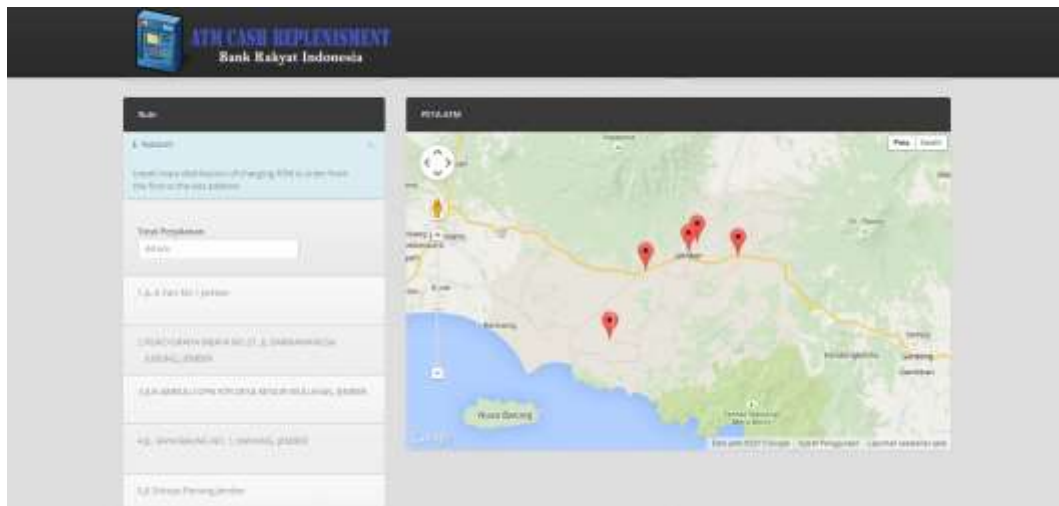
```

//menyimpan pada tabel hasil, dengan id=1 diinput pertama
$nomer = 1;
$jumlah=0;
$so = 0;
for ($i = 1; $i < count($_hasil); $i++) {
    echo ' --> ' . $_hasil[$i];
    $nomer++;
    $jumlah = $jumlah + $this->jarak_routing($_hasil[0], $_hasil[1]);
    echo ' --> ' . $jumlah;
    $so++;
}

$rute_perjalanan = array();
for ($i = 0; $i < count($_hasil); $i++) {
    array_push($rute_perjalanan, $this->m_view->ambil_combo("tb_atm", $_hasil[$i]));
}
$this->session->set_userdata("rute_perjalanan", $rute_perjalanan);
$this->session->set_userdata("total_jarak", $jumlah);

```

Hasil akhir adalah menampilkan halaman hasil *routing* dengan menampilkan total jarak, rute mana saja yang akan dituju dengan menggunakan alamat dan tampilan *marker* peta dari ATM mana saja yang telah dipilih menjadi objek. Hasil tampilan ini dapat dilihat pada Gambar 5.13.



Gambar 5.13 Tampilan Halaman Hasil Perhitungan Rute

5.4 Hasil Perbandingan Pendistribusian Pengisian ATM Secara Konvensional dan Menggunakan Sistem

Aplikasi Pendistribusian Pengisian ATM ini dibuat dengan tujuan untuk memudahkan bagian *IT & E Channel* dan bagian pendistribusian ATM agar dapat dengan mudah menjalankan tugas dengan menggunakan sistem yang sangat dibutuhkan dan belum ada sebelumnya untuk menyelesaikan permasalahan pendistribusian yang masih menggunakan teknik manual selama ini.

Proses pengambilan keputusan yang dilakukan selama ini adalah ketika ATM membunyikan alarm untuk memberi peringatan tentang sesuatu yang terjadi seperti uang di kas ATM telah mencapai batas, kertas *roll* yang habis, atau kamera pengaman yang rusak maka bagian *IT & E Channel* dan bagian pendistribusian ATM melakukan pencatatan secara manual pada kertas yang selanjutnya melakukan pemberangkatan menuju tempat ATM yang memberikan pemberitahuan. Pemberangkatan menuju lokasi ATM hanya dilakukan dengan proses pengambilan keputusan secara bersama untuk mendapatkan rute mana saja yang akan dilewati terlebih dahulu.

Hasil rute dari pengambilan keputusan secara bersama selanjutnya dibandingkan dengan hasil rute yang didapatkan dari sistem, dengan mengambil permisalan 5 objek ATM yakni ATM BRI CDM Cabang BRI Jember, ATM BRI Alfamart DR. Soebandi, ATM Bank BRI Alfamart Basuki Rahmat, ATM BRI Alfamart Kh. Shidiq, ATM BRI RS. Perkebunan (PTP X). Keputusan yang didapat menghasilkan rute perjalanan yang sama dengan menggunakan sistem, yakni dimulai dari ATM BRI CDM Cabang BRI Jember, menuju ATM Bank BRI Alfamart Basuki Rahmat, menuju ATM BRI Alfamart Kh. Shidiq, ATM BRI Alfamart DR.

Soebandi dan rute terakhir menuju ATM BRI RS. Perkebunan (PTP X), namun terkadang untuk objek ATM yang berbeda, terdapat perbedaan rute yang didapatkan ketika menggunakan keputusan dan dengan menggunakan sistem.

5.5. Pembahasan Aplikasi Pendistribusian Pengisian ATM

Proses pembahasan yang dilakukan oleh peneliti dengan bagian *IT & E Channel* mendapatkan hasil bahwa sistem yang telah dibuat dapat mempermudah proses pencatatan untuk mendapatkan urutan lokasi pendistribusian ATM dengan optimal. Namun dibutuhkannya *trigger* kepada sistem sebelumnya agar pencatatan ATM dapat dilakukan secara otomatis tanpa perlu melakukan penginputan secara manual, dan data yang tertera pada ATM perlu lebih diidentifikasi berdasarkan dengan jenis masing-masing ATM.

Penentuan rute untuk pendistribusian pengisian ATM juga sangat membantu memberikan rute yang optimal, namun untuk beberapa objek ATM rute yang didapatkan terkadang kurang sesuai dengan keputusan yang diambil secara bersama.

Aplikasi pendistribusian pengisian ATM sangat memudahkan user atau pengguna karena dengan melihat sistem tersebut, user mendapatkan beberapa kemudahan diantaranya adalah dengan menggunakan aplikasi ini, user dapat melihat seluruh lokasi ATM yang ada di Kabupaten Jember dengan menggunakan peta sehingga memudahkan proses pencarian lokasi ATM. Manajemen data ATM lebih terkomputasi dengan baik. Pencatatan objek ATM dapat dilakukan dengan mudah, dan menghasilkan rute dengan cepat dan tervisualisasi dengan menggunakan peta untuk melakukan pendistribusian pengisian ATM.

BAB 6. PENUTUP

Bab penutup merupakan gambaran tentang kesimpulan dari seluruh sistem yang telah dibangun oleh peneliti, dan diharapkan nantinya dari kesimpulan dan saran yang diberikan akan digunakan sebagai acuan untuk digunakan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

1. Aplikasi pendistribusian pengisian ATM pada PT. BRI (Persero) Tbk Cabang Jember telah dirancang dan dibangun dengan mempunyai 2 hak akses dan berbagai fitur. Hak akses pertama adalah hak akses untuk *user*, dimana *user* dapat menjalankan fitur untuk melihat peta ATM *user*, melihat data BRI dan melakukan perhitungan rute. Hak akses kedua adalah hak akses untuk admin, dimana admin dapat menjalankan fitur melihat peta ATM admin, melakukan perhitungan jarak, dan mengelola data ATM.
2. Pencarian optimasi rute untuk pendistribusian pengisian Anjungan Tunai Mandiri dengan menggunakan algoritma CIH adalah dengan cara menyisipkan objek ATM secara bergantian dengan menggunakan proses pengulangan hingga semua ATM telah masuk menjadi sebuah *subtour* atau rute dengan perhitungan jarak untuk mendapatkan total jarak yang terkecil.
3. Aplikasi pendistribusian pengisian Anjungan Tunai Mandiri dirancang dan dibangun dengan melalui beberapa tahapan yakni menganalisa data ATM dan data lokasi ATM yang nantinya akan dibutuhkan untuk melakukan sebuah perhitungan dengan menggunakan algoritma CIH yang dapat menghasilkan sebuah optimasi rute untuk pendistribusian pengisian ATM.

6.2 Saran

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

1. Memberikan *trigger* pada sistem lain yang terhubung dengan sistem ini.
2. Meningkatkan sistem dengan proses identifikasi terhadap data masing-masing ATM dengan lebih maksimal.

DAFTAR PUSTAKA

- Anwar, C. (2015, february 11). *Pengertian dan Kegunaan Kartu ATM dan Kartu Kredit*. Retrieved from Tips Aman Pemegang kartu Debit: <http://tipsamanpemegangkartudebit.blogspot.com/2013/06/pengertian-dan-kegunaan-kartu-atm-dan.html>
- Apriliani, I. (2011). Penyelesaian Travelling Salesman Problem (TSP) mengguakan Algoritma Semut dan Algoritma Cheapest Insertion Heuristic (CIH). *Majalah Ilmiah Matematika dan Statistika*, 7-8.
- Dharmmesta, B. (1999). *Saluran Pemasaran : Konsep dan Strategi Analisis Kuantitatif*. Yogyakarta: BPFE-Yogyakarta.
- Johnsonbaugh, R. (1998). *Matematika Diskrit Edisi 4 Jilid 1*. Jakarta: PT Prenhalliondo.
- Jonhson, D. S., & Mc Geoch, L. A. (1995). *The Travelling Salesman Problem : A Case Study in Local Optimization Aarts*. London: E. H. L. and Lenstra, J. K. (ed.) Local Search in Combinatorial Optimization. John Wiley and Sons. Ltd.
- Kotler, P. (2002). *Manajemen Pemasaran (Terjemahan)*. Jakarta: Prenhallindo.
- Kusrini, & Istiyanto, E. J. (2007). Penyelesaian Travelling salesman Problem dengan Algoritma Cheapest Insertion Heuristic dan Basis Data. *Jurnal Informatika*, 110-113.
- Masykur, F. (2014, Nopember). Implementasi Sistem Informasi Geografis Menggunakan Google Maps API Dalam Pemetaan Asal Mahasiswa. *Jurnal SIMETRIS*, 5, No.2.
- Nidhra, S., & Dondeti, J. (2012). BLACK BOX AND WHITE BOX TESTING TECHNIQUES – A LITERATURE REVIEW. *International Journal of Embedded Systems and Applications (IJESA)* , 33.
- Pressman, R. S. (1997). *Software Engineering : A Practioner's Approach*. 4th McGrawHill.
- Rothlauf, F. (2011). *Design of Modern Heuristic*. London: Springer.
- Slamin. (2009). *DESAIN JARINGAN : Pendekatan Teori Graf*. Jember: Jember University Press.
- Slamin. (2010). *Teori Graf dan Fenomena di Universitas Jember sebagai Model Beragam Aplikasinya*. Jember: Jember University Press.
- Wiyanti, D. T. (2013). Algoritma Optimasi untuk Penyelesaian Travelling Salesman Problem. *Jurnal Transformatika*, 1.

LAMPIRAN

1. LAMPIRAN A (*Usecase Skenario*)

1.1. *Usecase Skenario Login*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *login*.

Tabel 1.1 *Usecase Skenario Login*

Nama	<i>Login</i>
Trigger	<i>Admin</i>
Pre Condition	<i>Admin</i> menekan button “ <i>login</i> ”
Post Condition	<i>Admin</i> berhasil <i>login</i> atau masuk ke halaman system untuk <i>admin</i> .

SKENARIO NORMAL LOGIN

Aktor	Sistem
1. Menjalankan sistem	2. Menampilkan halaman <i>dashboard user</i>
3. Klik button “ <i>Login</i> ”	4. Menampilkan Form <i>Login</i>

SKENARIO NORMAL SIGN IN

Aktor	Sistem
5. Memasukkan <i>Username</i> dan <i>Password</i>	
6. Klik button “ <i>Login</i> ”	7. Melakukan autentifikasi ke dalam <i>database</i>
	8. Menampilkan halaman <i>dashboard admin</i>

SKENARIO NORMAL CANCEL

Aktor	Sistem
5. Klik button “ <i>Batal</i> ”	

6. Menampilkan halaman *dashboard user*

SKENARIO ALTERNATIF LOGIN

Jika aktor tidak memasukkan *username* dan *password*

Aktor	Sistem
5. <i>Username</i> (kosong) atau <i>Password</i> (kosong)	
6. Klik <i>button</i> “ <i>Login</i> ”	
	7. Mengecek masukan <i>username</i> dan <i>password</i> pada <i>database</i>
	8. Menampilkan <i>alert</i> “ Maaf! Isi dengan lengkap ' <i>username</i> ' dan ' <i>password</i> '”
	9. Pilih tanda ‘x’ atau close
	10. Menampilkan Form <i>Login</i>

Jika aktor memasukkan *username* dan *password* salah

Aktor	Sistem
5. Mengisi <i>username</i> dan <i>password</i>	
6. Klik <i>button</i> “ <i>Login</i> ”	
	7. Mengecek masukan <i>username</i> dan <i>password</i> pada <i>database</i>
	8. Menampilkan <i>alert</i> “ Maaf! ' <i>username</i> ' dan ' <i>password</i> ' tidak terdaftar”
	9. Pilih tanda ‘x’ atau close
	10. Menampilkan Form <i>Login</i>

1.2. *Usecase* Skenario *Logout*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *logout*.

Tabel 1.2 *Usecase* Skenario *Logout*

Nama	<i>Logout</i>
<i>Trigger</i>	<i>Admin</i>
<i>Pre Condition</i>	<i>Admin</i> menekan button " <i>logout</i> "
<i>Post Condition</i>	<i>Admin</i> berhasil <i>logout</i> atau keluar dari sistem.

SKENARIO NORMAL LOGOUT

Aktor	Sistem
1. Klik button " <i>Logout</i> " di bagian pojok kanan menu	2. Menghapus data <i>session</i>
	3. Menampilkan halaman <i>dashboard</i> user

1.3. *Usecase* Skenario Melihat Peta ATM *Admin*

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat peta ATM *Admin*.

Tabel 1.3 *Usecase* Skenario Melihat Peta ATM *Admin*

Nama	Melihat Peta ATM <i>Admin</i>
<i>Trigger</i>	<i>Admin</i>
<i>Pre Condition</i>	<i>Admin</i> mengakses sistem
<i>Post Condition</i>	<i>Admin</i> dapat melihat peta ATM yang ada di seluruh kawasan kecamatan Jember

SKENARIO NORMAL MELIHAT PETA ATM ADMIN

Aktor	Sistem
1. Menjalankan system	2. Mengambil data peta ATM dari database
	3. Menampilkan halaman <i>dashboard admin</i> yang terdapat peta yang berisi data ATM BRI dan keterangan sistem

SKENARIO NORMAL MELIHAT DATA DETAIL ATM ADMIN

Aktor	Sistem
4. Memilih salah satu <i>icon</i> ATM yang terdapat pada peta	5. Mengambil data detail ATM dari database
	6. Menampilkan nama dan alamat detail ATM sesuai dengan yang telah dipilih
7. Klik 'x' atau 'close' pada info yang tampil	8. Menampilkan halaman <i>dashboard admin</i>

1.4. *Usecase* Skenario Melihat Peta ATM User

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif usecase skenario melihat peta ATM User.

Tabel 1.4 *Usecase* Skenario Melihat Peta ATM User

Nama	Melihat Peta ATM User
<i>Trigger</i>	User
<i>Pre Condition</i>	User berhasil login
<i>Post Condition</i>	User dapat melihat peta ATM yang ada di seluruh kawasan kecamatan Jember

SKENARIO NORMAL MELIHAT PETA ATM USER

Aktor	Sistem
1. Menjalankan system	2. Mengambil data peta ATM dari database
	3. Menampilkan halaman <i>dashboard user</i> yang terdapat peta yang berisi data ATM BRI dan keterangan sistem

SKENARIO NORMAL MELIHAT DATA DETAIL ATM USER

Aktor	Sistem
4. Memilih salah satu <i>icon</i> ATM yang terdapat pada peta	5. Mengambil data detail ATM dari database
	6. Menampilkan nama dan alamat detail ATM
7. Klik ‘x’ atau ‘close’ pada info yang tampil	8. Menampilkan halaman <i>dashboard user</i>

1.5. *Usecase* Skenario Melihat Data BRI

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario melihat data BRI.

Tabel 1.5 *Usecase* Skenario Melihat Data BRI

Nama	Melihat Data BRI
<i>Trigger</i>	<i>User</i>
<i>Pre Condition</i>	<i>User</i> menekan menu “About BRI”
<i>Post Condition</i>	<i>User</i> dapat melihat data BRI yakni sejarah dan visi misi tentang BRI

SKENARIO NORMAL MELIHAT DATA BRI

Aktor	Sistem
1. Memilih menu “Tentang BRI”	2. Menampilkan <i>dropdown</i> menu “Sejarah BRI” dan “Visi dan Misi”

SKENARIO NORMAL MEMILIH MENU “SEJARAH”

Aktor	Sistem
3. Memilih menu “Sejarah BRI”	4. Menampilkan informasi tentang sejarah berdirinya bank BRI

SKENARIO NORMAL MEMILIH MENU “VISI & MISI”

Aktor	Sistem
3. Memilih menu “visi & misi”	4. Menampilkan informasi tentang visi & misi bank BRI

1.6. *Usecase* Skenario Perhitungan Jarak

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario perhitungan jarak.

Tabel 1.6 *Usecase* Skenario Perhitungan Jarak

Nama	Perhitungan Jarak
Trigger	Admin
Pre Condition	Admin menekan menu “Perhitungan Jarak”
Post Condition	Admin input data pada peta hingga menghasilkan jarak antar ATM yang dibutuhkan

SKENARIO NORMAL PERHITUNGAN JARAK

Aktor	Sistem
1. Memilih menu “Perhitungan Jarak”	2. Menampilkan halaman Perhitungan Jarak yang berisi pilihan Lokasi dan Bantuan. Lokasi berisi dropdown pilihan Tambah Lokasi Awal dan Tambah Lokasi Tujuan dan juga peta bank BRI di kabupaten Jember dan Bantuan berisi bantuan untuk menggunakan sistem
3. Klik button “Bantuan”	4. Sistem menampilkan bantuan berisi bantuan untuk menggunakan sistem

SKENARIO NORMAL PILIH LOKASI ATM

Aktor	Sistem
5. Memilih salah satu ATM yang ada pada <i>dropdown</i> lokasi awal	
6. Memilih salah satu ATM yang ada pada <i>dropdown</i> lokasi tujuan	
7. Klik button " <i>Hitung Jarak</i> "	8. Menampilkan data alamat lokasi awal, alamat lokasi tujuan, jarak dan waktu tempuh

SKENARIO NORMAL INPUT PERHITUNGAN JARAK

Aktor	Sistem
9. Menekan <i>button</i> "Input Perhitungan Jarak"	10. Sistem menampilkan <i>modal</i> dengan text field berupa lokasi awal, lokasi tujuan, jarak

SKENARIO NORMAL SIMPAN PERHITUNGAN JARAK

Aktor	Sistem
11. Menekan <i>button</i> "simpan"	12. Melakukan pengecekan pada database
	13. Menyimpan data pada database
	14. Menampilkan <i>alert</i> " Selamat! Jarak dari ATM berhasil di simpan"
15. Klik tanda 'x' atau <i>close</i> pada alert	16. Menampilkan halaman Perhitungan Jarak

SKENARIO NORMAL BATAL PERHITUNGAN JARAK

Aktor	Sistem
11. Menekan <i>button</i> “batal”	12. Menampilkan halaman Perhitungan Jarak

SKENARIO ALTERNATIF SIMPAN PERHITUNGAN JARAK

Jika aktor menginputkan jarak ATM yang telah ada pada database

Aktor	Sistem
11. Menekan <i>button</i> “simpan” (data ATM telah ada pada <i>database</i>)	12. Melakukan pengecekan pada database
	13. Menampilkan halaman <i>Calculate Distance</i> dan <i>alert</i> “Maaf! Jarak dari ATM telah diinputkan”
14. Klik tanda ‘x’ atau <i>close</i> pada alert	15. Menampilkan halaman Perhitungan Jarak

Jika aktor tidak mengisi form isian

Aktor	Sistem
11. Menekan <i>button</i> “simpan” (data ATM dan jarak tidak terisi)	12. Menampilkan <i>alert</i> “Maaf! Data tidak boleh kosong”
13. Klik tanda ‘x’ atau <i>close</i> pada alert	14. Menampilkan halaman Perhitungan Jarak

1.7. *Usecase* Skenario Manajemen Data ATM

Penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario manajemen data ATM

Tabel 1.7 *Usecase* Skenario Manajemen Data ATM

Nama	Manajemen Data ATM
<i>Trigger</i>	<i>Admin</i>
<i>Pre Condition</i>	<i>Admin</i> menekan menu “Data ATM”
<i>Post Condition</i>	<i>Admin</i> dapat memanajemen data ATM

SKENARIO NORMAL MANAJEMEN DATA ATM

Aktor	Sistem
1. Memilih menu “Data ATM”	2. Menampilkan halaman Data ATM berupa tabel yang berisi seluruh data ATM
3. Menekan <i>next</i> pada bagian bawah tabel	4. Menampilkan data ATM selanjutnya yang belum ditampilkan

SKENARIO NORMAL SHOW ATM

Aktor	Sistem
5. Menekan <i>dropdown</i> “ <i>show</i> ”	6. Menampilkan pilihan jumlah data ATM yang akan di tampilkan
7. Memilih salah satu pilhan dari <i>dropdown</i>	8. Menampilkan jumlah data ATM sesuai yang telah di pilih

SKENARIO NORMAL SEARCH

Aktor	Sistem
5. Menginputkan inisial nama ATM yang akan kita cari pada <i>text field search</i>	

6. Menampilkan nama ATM yang sesuai dengan inisial yang telah kita inputkan

SKENARIO NORMAL *ADD NEW ATM*

Aktor	Sistem
5. Menekan button “Tambah ATM Baru”	6. Menampilkan halaman tambah ATM baru

SKENARIO NORMAL *ADD SUBMIT*

Aktor	Sistem
7. Menginputkan ID ATM, Nama ATM dan alamat ATM yang baru yang di inputkan	
8. Menggeser <i>icon</i> ATM yang yang terdapat pada peta untuk memilih lokasi ATM baru yang akan di buat	9. Menampilkan <i>lattitude</i> dan <i>longitude</i> sesuai dengan lokasi yang di pilih
10. Klik <i>button</i> “simpan”	11. Menyimpan data pada database
	12. Menampilkan halaman Data ATM dan <i>alert</i> “ Selamat Berhasil Input ATM ”

SKENARIO NORMAL *ADD RESET*

Aktor	Sistem
7. Klik <i>button</i> “reset”	8. Menampilkan kembali form isian yang kosong

SKENARIO NORMAL EDIT ATM

Aktor	Sistem
5. Menekan <i>button</i> “ <i>Edit</i> ”	6. Menampilkan halaman <i>edit</i> ATM dan menampilkan data ATM yang akan di <i>edit</i>

SKENARIO NORMAL EDIT SUBMIT

Aktor	Sistem
7. Mengedit nama ATM dan alamat ATM yang baru	
8. Menggeser <i>icon</i> ATM yang yang terdapat pada peta untuk memilih lokasi ATM baru yang akan di perbaharui	9. Menampilkan <i>latitude</i> dan <i>longitude</i> sesuai dengan lokasi yang di pilih
10. Klik <i>button</i> “ <i>simpan</i> ”	11. Menyimpan data pada database
	12. Menampilkan halaman Data ATM dan <i>alert</i> “ Selamat Berhasil Input ATM ”

SKENARIO NORMAL EDIT RESET

Aktor	Sistem
7. Klik <i>button</i> “ <i>reset</i> ”	8. Menampilkan kembali form isian yang kosong

SKENARIO NORMAL DELETE ATM

Aktor	Sistem
5. Klik <i>button</i> “ <i>delete</i> ”	6. Menampilkan <i>warning</i> “ <i>Apakah kamu yakin akan menghapus data ATM ini?</i> ”

SKENARIO NORMAL OK DELETE ATM**Aktor****Aktor**

- | | |
|----------------------------|---|
| 7. Klik <i>button</i> "OK" | 8. Data terhapus dari database |
| | 9. Sistem menampilkan halaman data ATM dan <i>alert</i> "Selamat!! Data ATM berhasil dihapus" |

SKENARIO NORMAL CANCEL DELETE ATM**Aktor****Sistem**

- | | |
|-------------------------------|--|
| 7. Klik <i>button</i> "batal" | 8. Sistem menampilkan halaman data ATM |
|-------------------------------|--|

SKENARIO ALTERNATIF ADD SUBMIT

Jika aktor tidak menginputkan seluruh atau salah satu dari form isian

Aktor**Sistem**

- | | |
|---------------------------------|--|
| 10. Klik <i>button</i> "simpan" | 11. Menampilkan halaman Add New ATM dan <i>alert</i> "Maaf! Isi data dengan lengkap" |
|---------------------------------|--|

SKENARIO ALTERNATIF EDIT SUBMIT

Jika aktor tidak menginputkan seluruh atau salah satu dari form isian

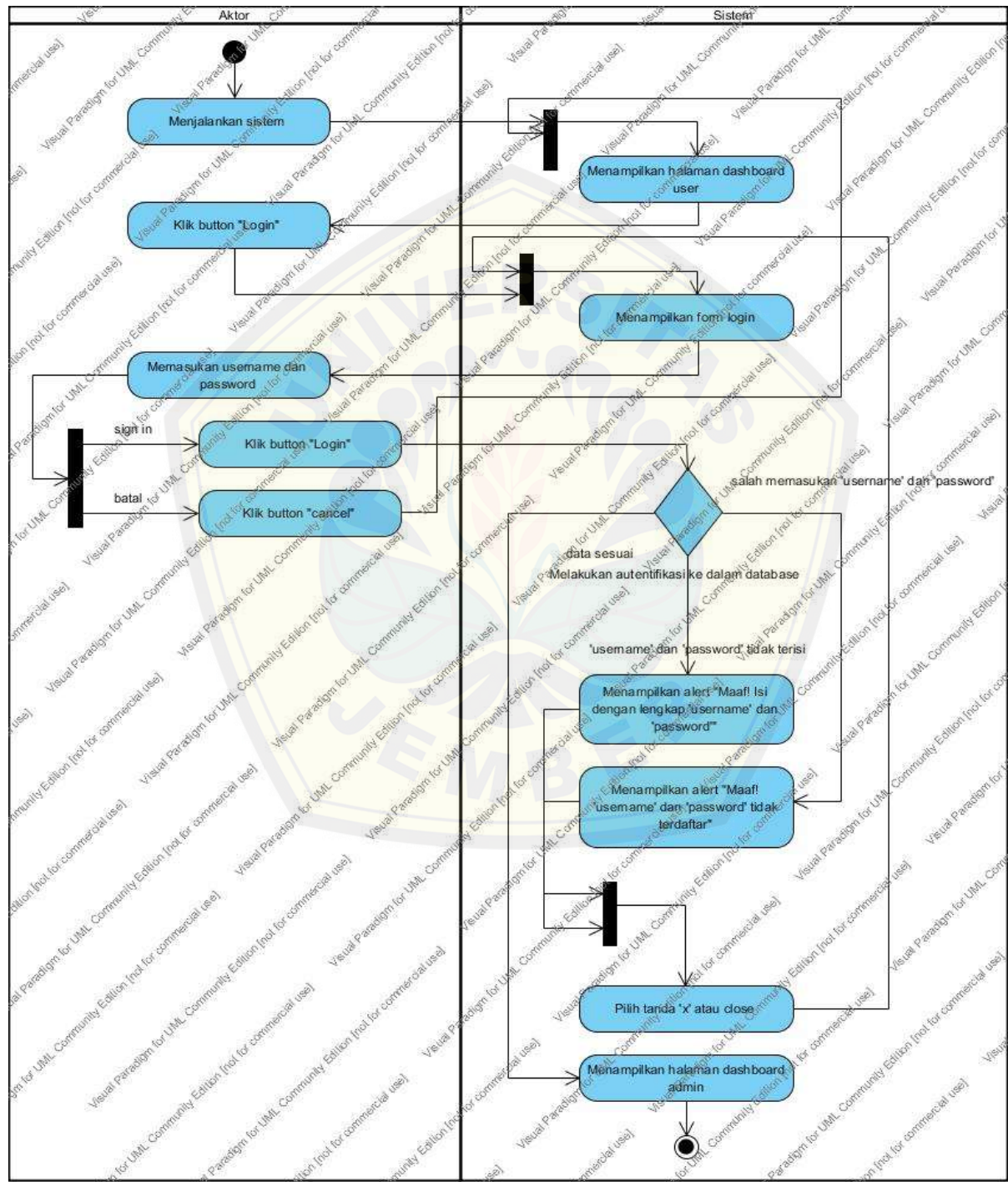
Aktor**Aktor**

- | | |
|---------------------------------|--|
| 10. Klik <i>button</i> "simpan" | 11. Menampilkan halaman Add New ATM dan <i>alert</i> "Maaf! Isi data dengan lengkap" |
|---------------------------------|--|

2. LAMPIRAN B (Activity Diagram)

2.1. Activity Diagram Login

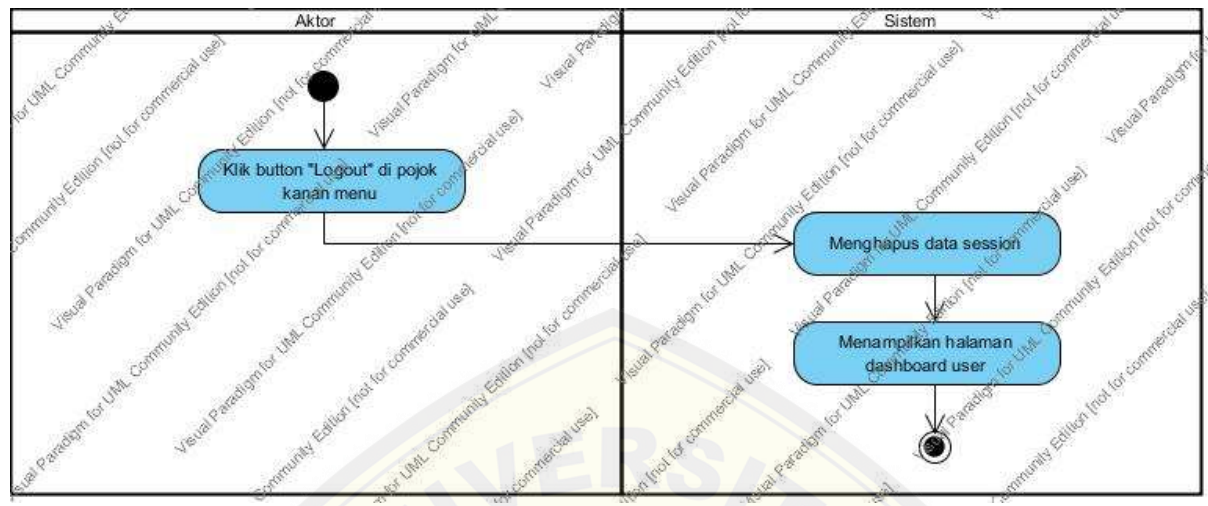
Activity diagram login dapat dilakukan oleh admin. Activity diagram login digunakan untuk proses login oleh admin untuk masuk ke halaman admin, yang dapat dilihat pada gambar di bawah.



Gambar 2.1 Activity Diagram Login

2.2. Activity Diagram Logout

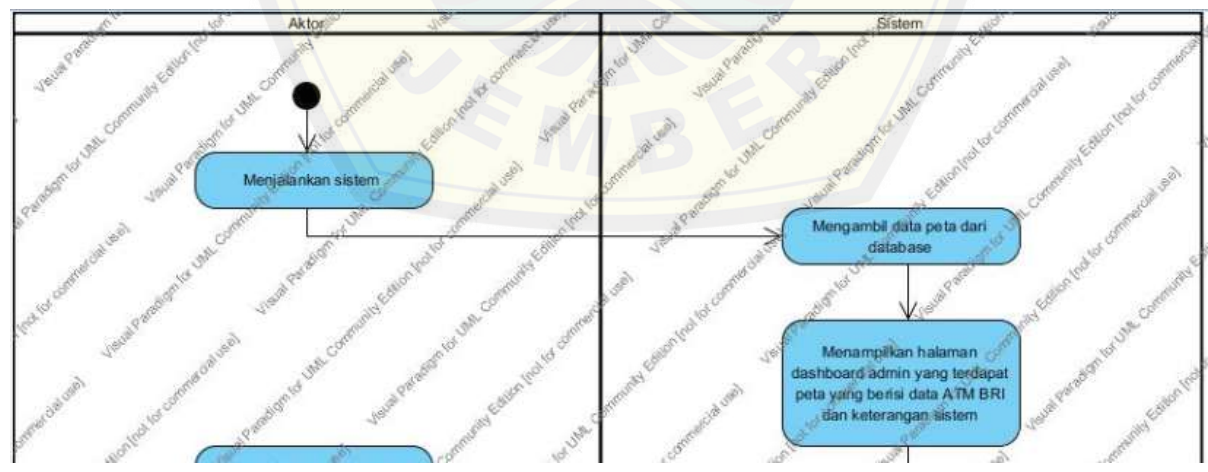
Activity diagram logout dapat dilakukan oleh admin. Activity diagram logout digunakan untuk proses logout oleh admin untuk keluar dari halaman admin, yang dapat dilihat pada gambar di bawah.

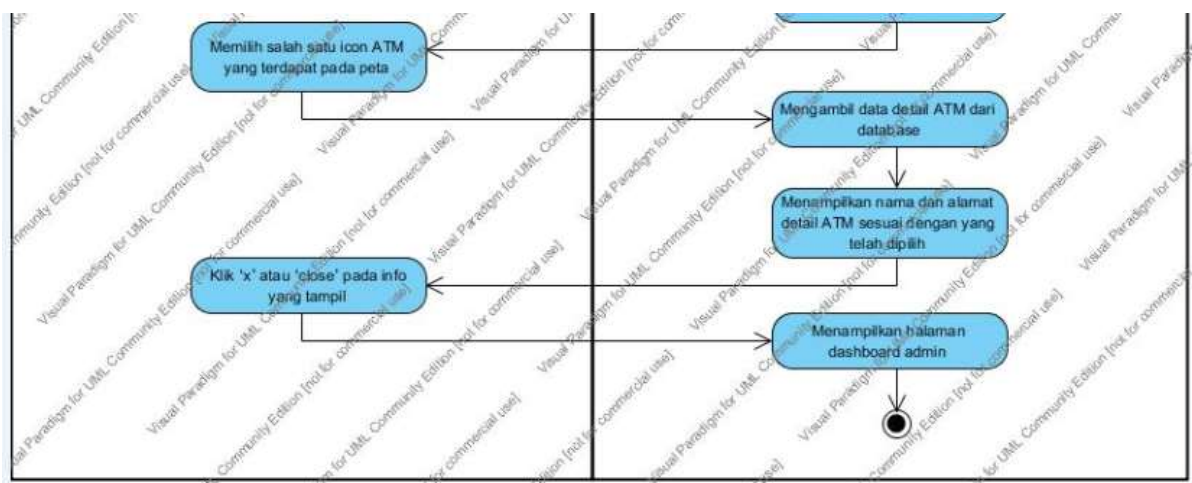


Gambar 2.2 Activity Diagram Logout

2.3. Activity Diagram Melihat Peta ATM Admin

Activity diagram melihat peta ATM admin dapat dilakukan oleh admin. Activity diagram melihat peta ATM admin dapat juga disebut dashboard yang digunakan untuk proses melihat peta seluruh ATM BRI yang ada di Kabupaten Jember, yang dapat dilihat pada gambar di bawah.

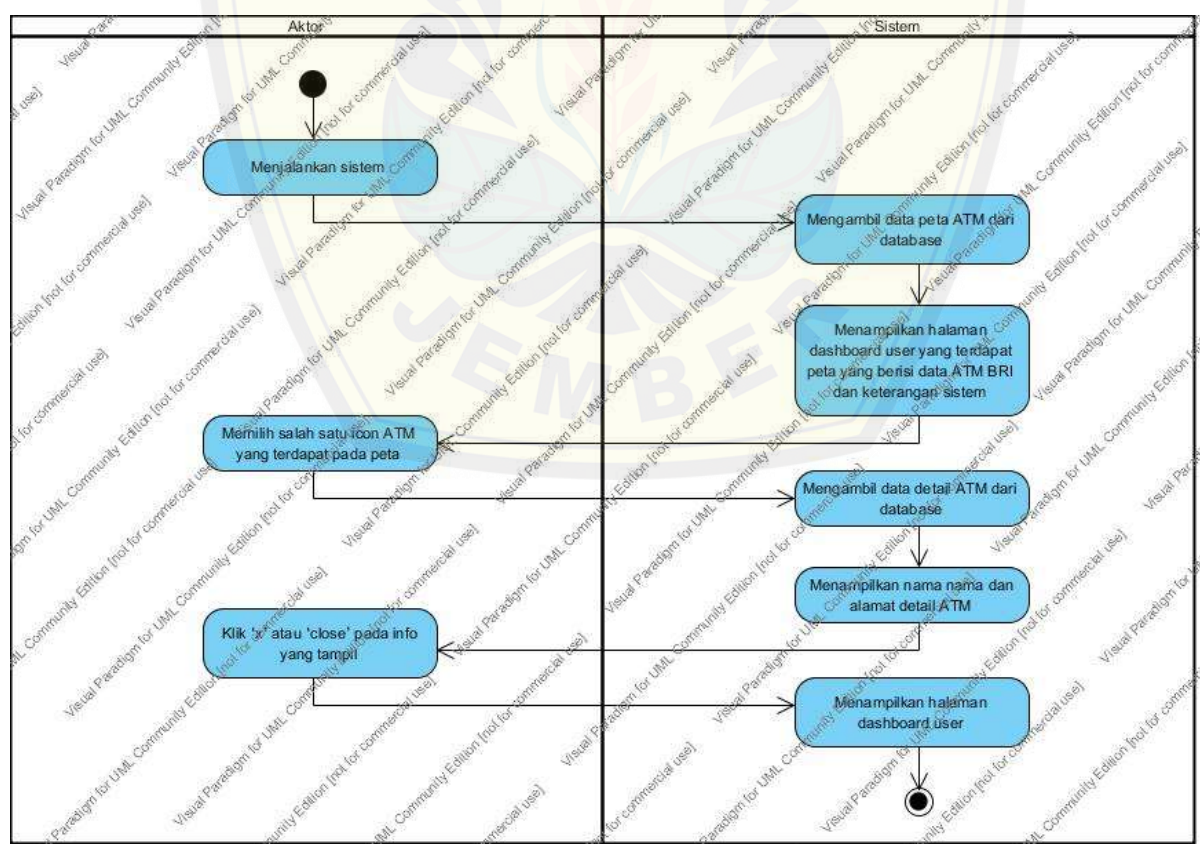




Gambar 2.3 Activity Diagram Melihat Peta ATM Admin

2.4. Activity Diagram Melihat Peta ATM User

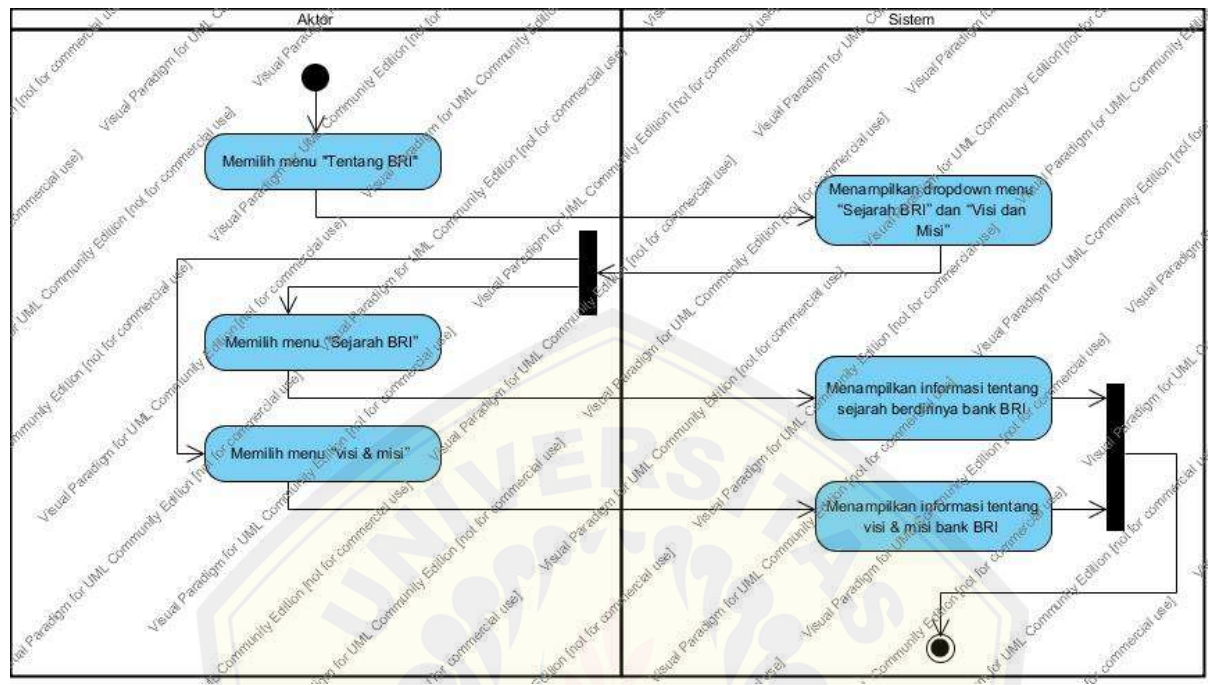
Activity diagram melihat peta ATM user dapat dilakukan oleh user. Activity diagram melihat peta ATM user dapat juga disebut dashboard yang digunakan untuk proses melihat peta seluruh ATM BRI yang ada di Kabupaten Jember, yang dapat dilihat pada gambar di bawah.



Gambar 2.4 Activity Diagram Melihat Peta ATM User

2.5. Activity Diagram Melihat Data BRI

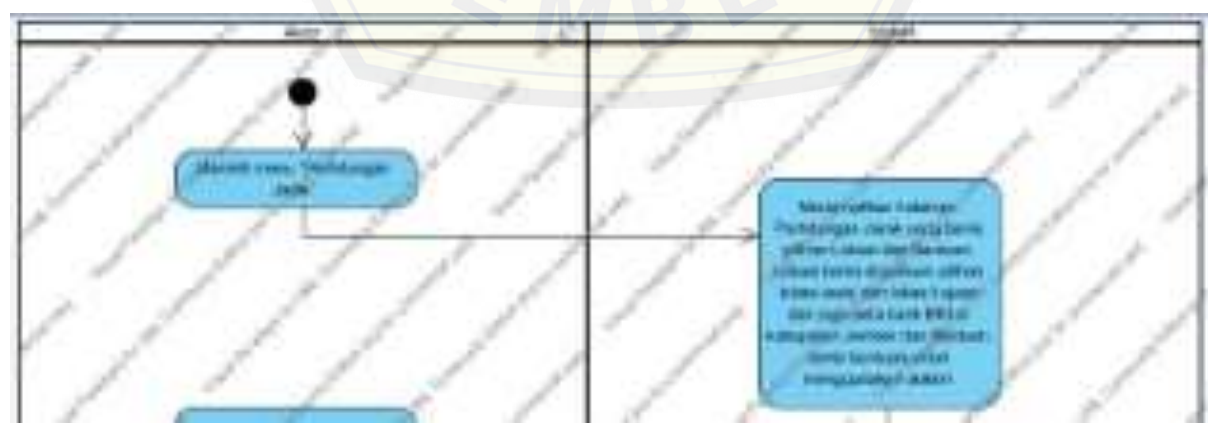
Activity diagram melihat data BRI dapat dilakukan oleh user. Activity diagram melihat data BRI digunakan untuk melihat sejarah, visi dan misi Bank BRI, yang dapat dilihat pada gambar di bawah.

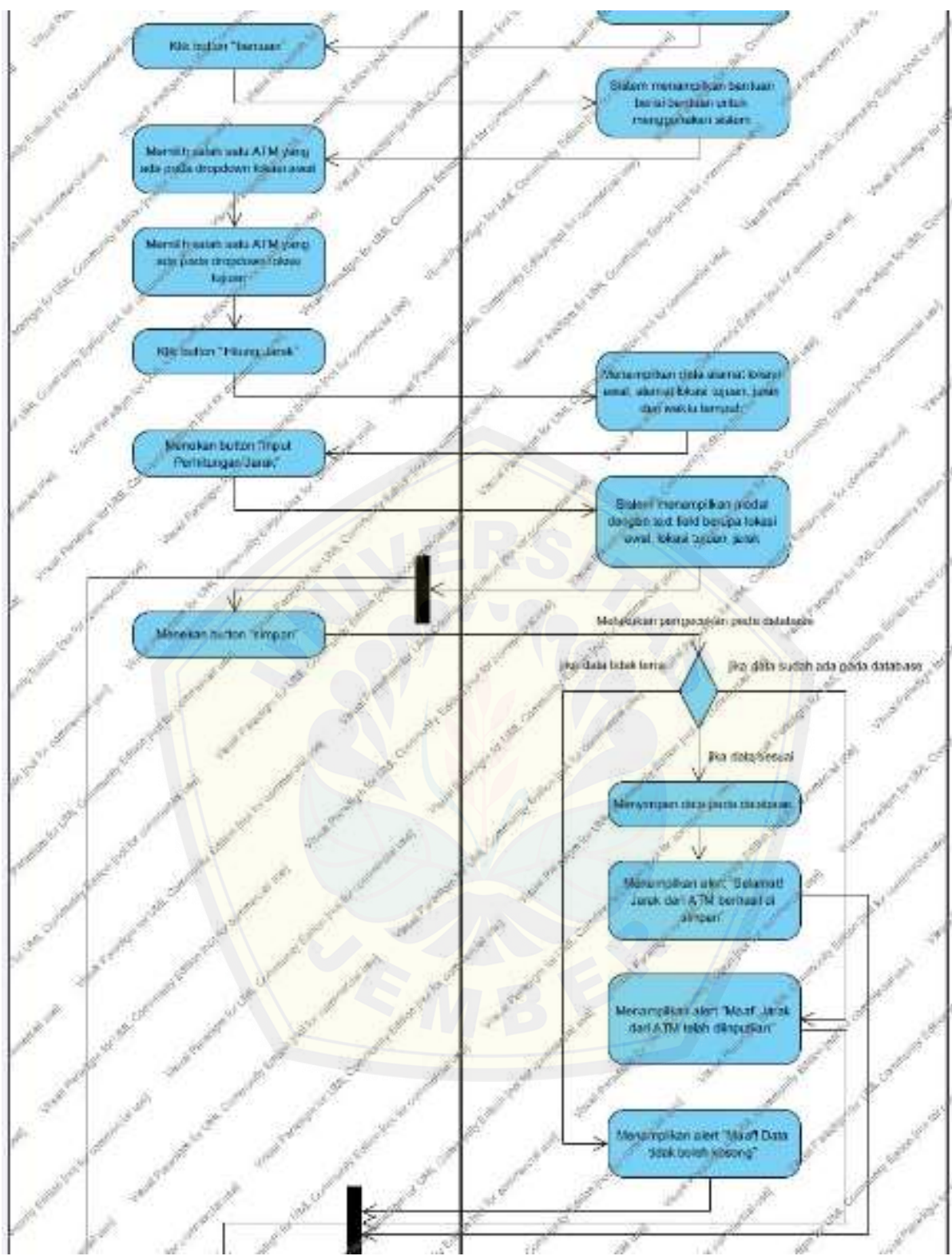


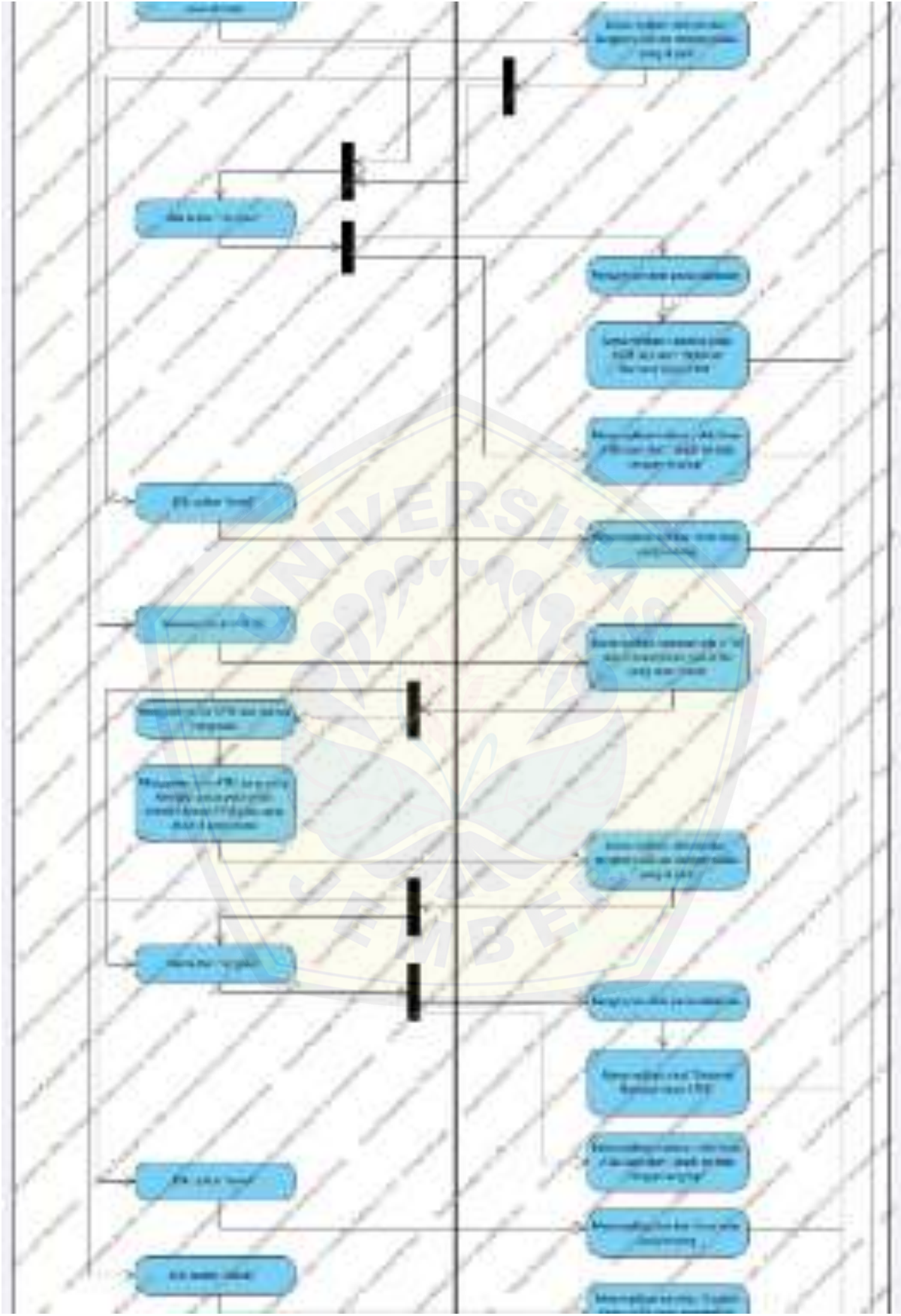
Gambar 2.5 Activity Diagram Melihat Data BRI

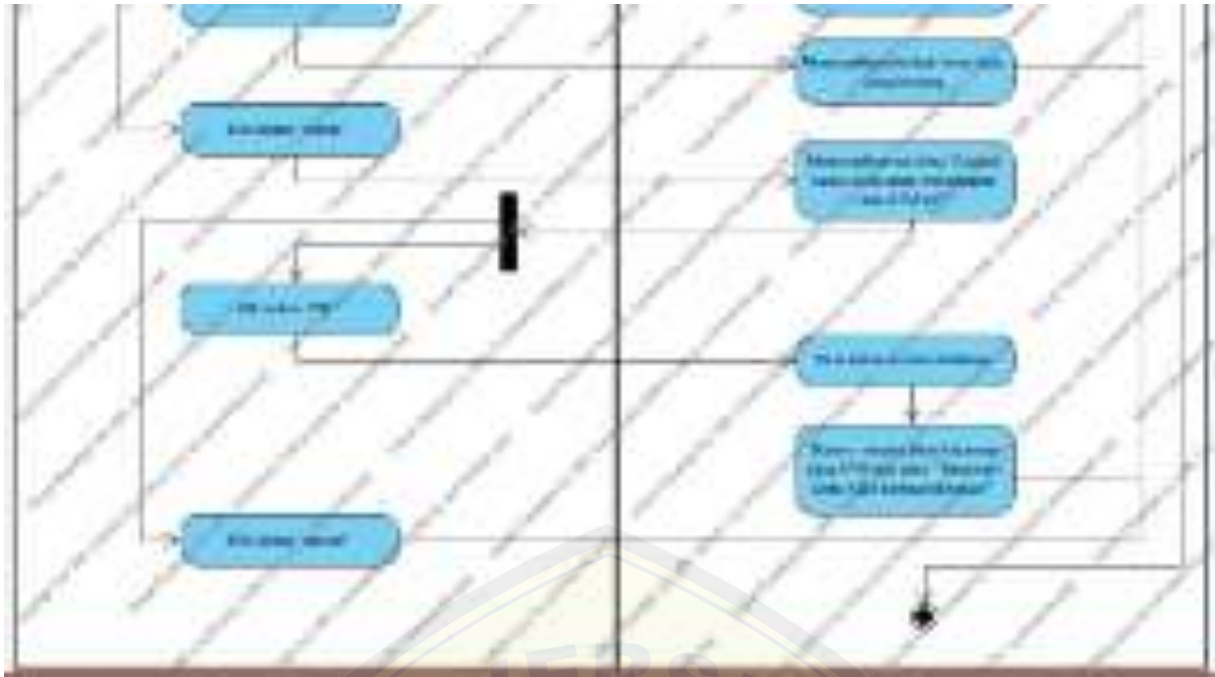
2.6. Activity Diagram Perhitungan Jarak

Activity diagram perhitungan jarak dapat dilakukan oleh admin. Activity diagram calculate distance digunakan untuk melakukan perhitungan antara 2 atm untuk mencari jarak yang selanjutnya akan disimpan pada database, yang dapat dilihat pada gambar di bawah.

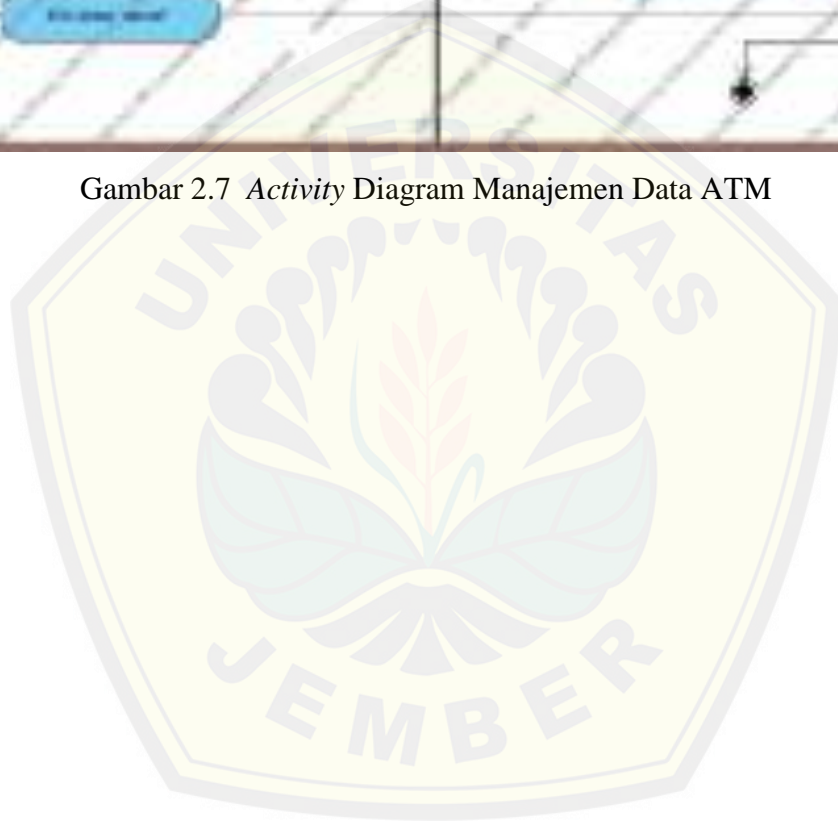






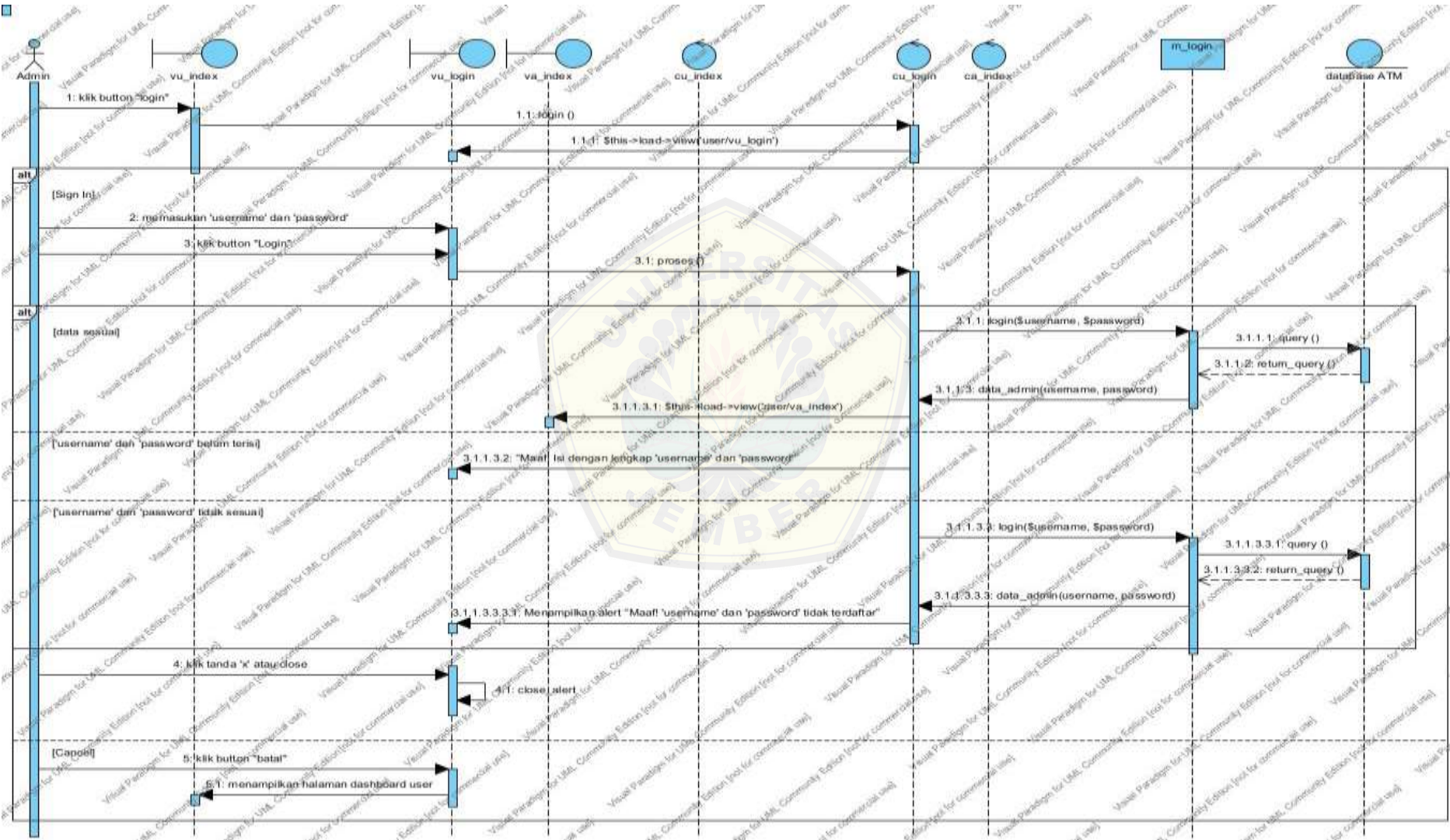


Gambar 2.7 Activity Diagram Manajemen Data ATM

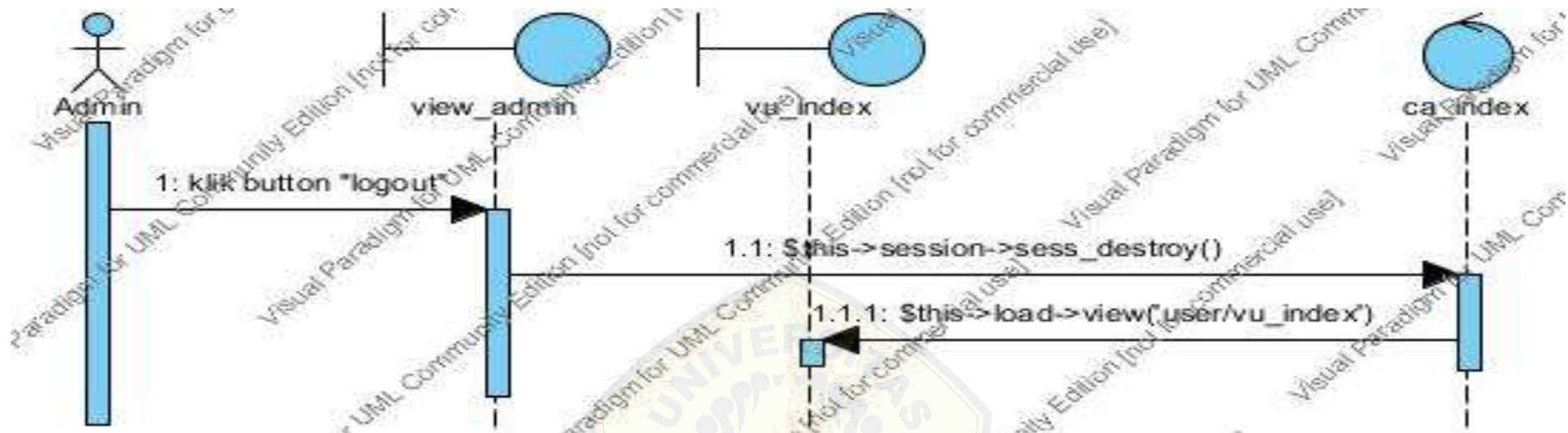


3. LAMPIRAN C (Sequence Diagram)

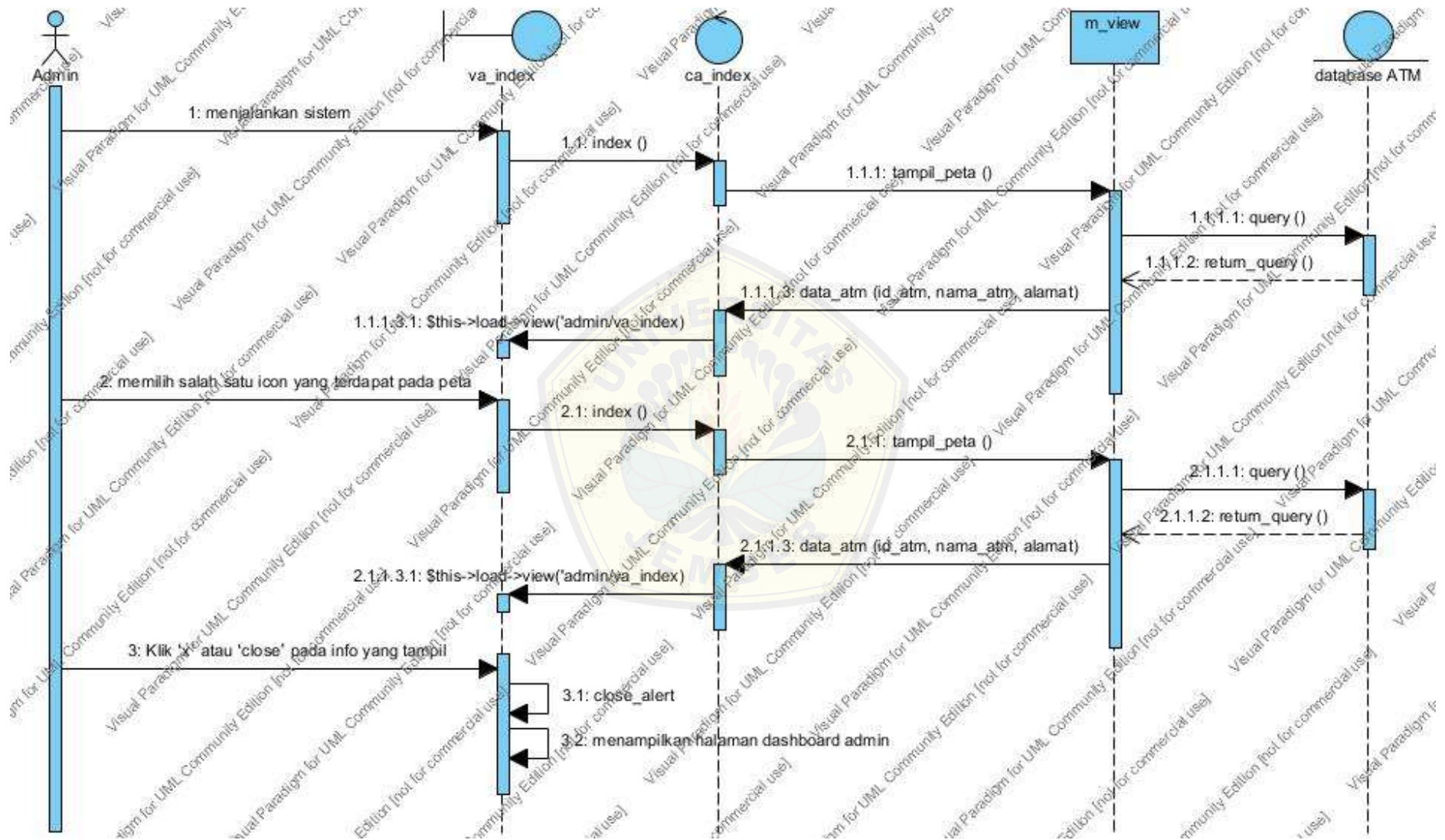
3.1. Sequence Diagram Login



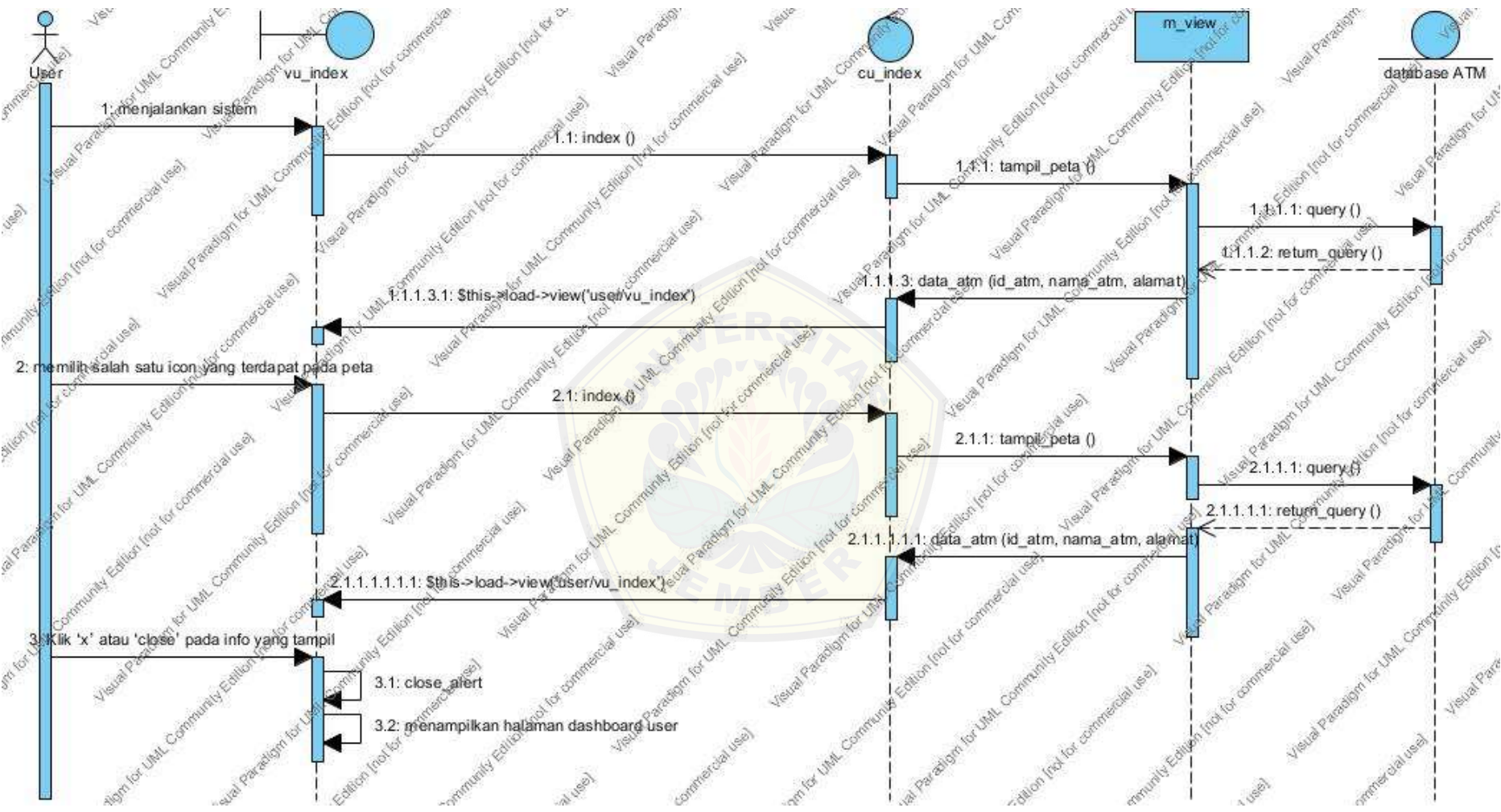
3.2. Sequence Diagram Logout



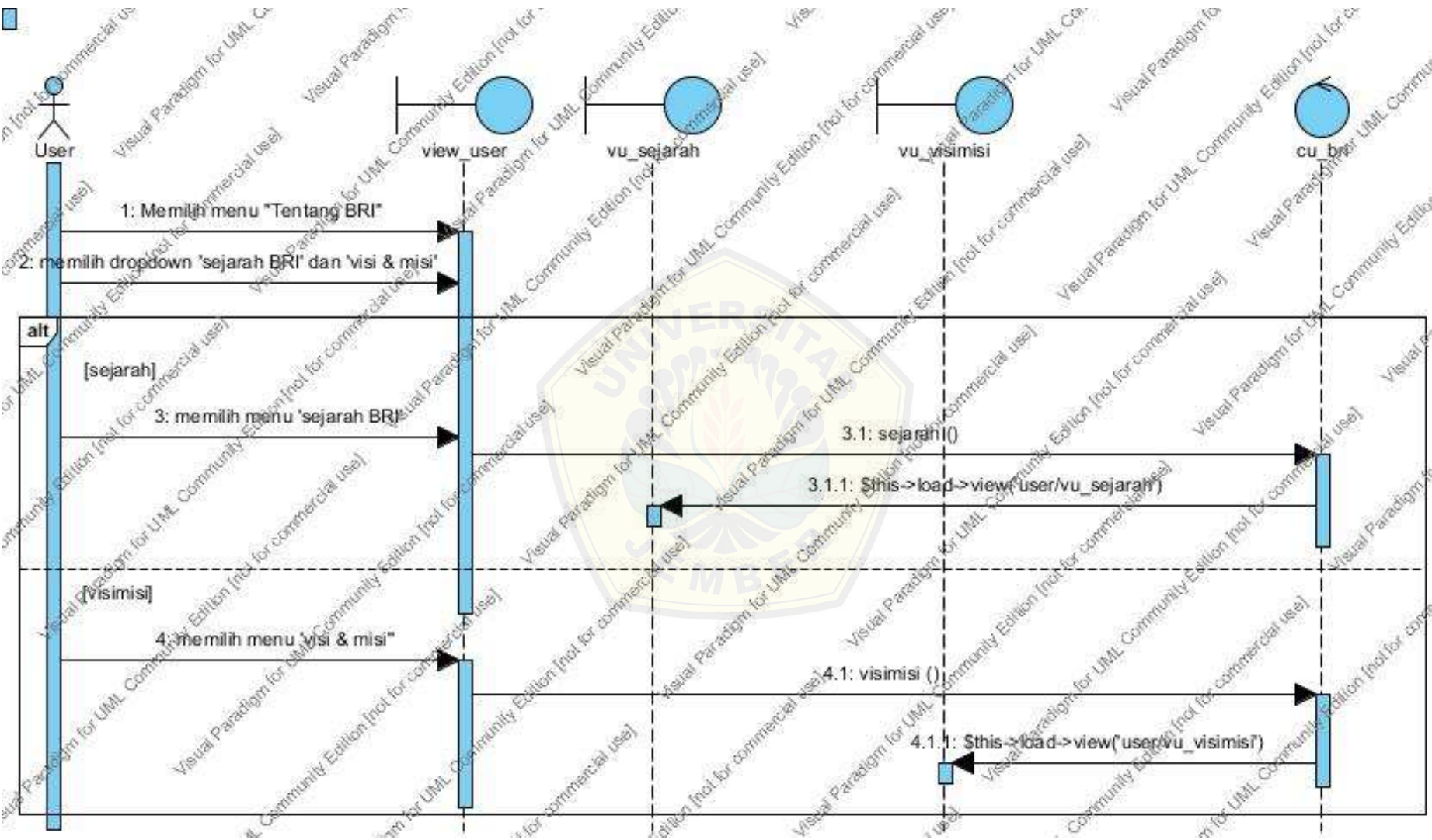
3.3. Sequence Diagram Melihat Peta ATM Admin



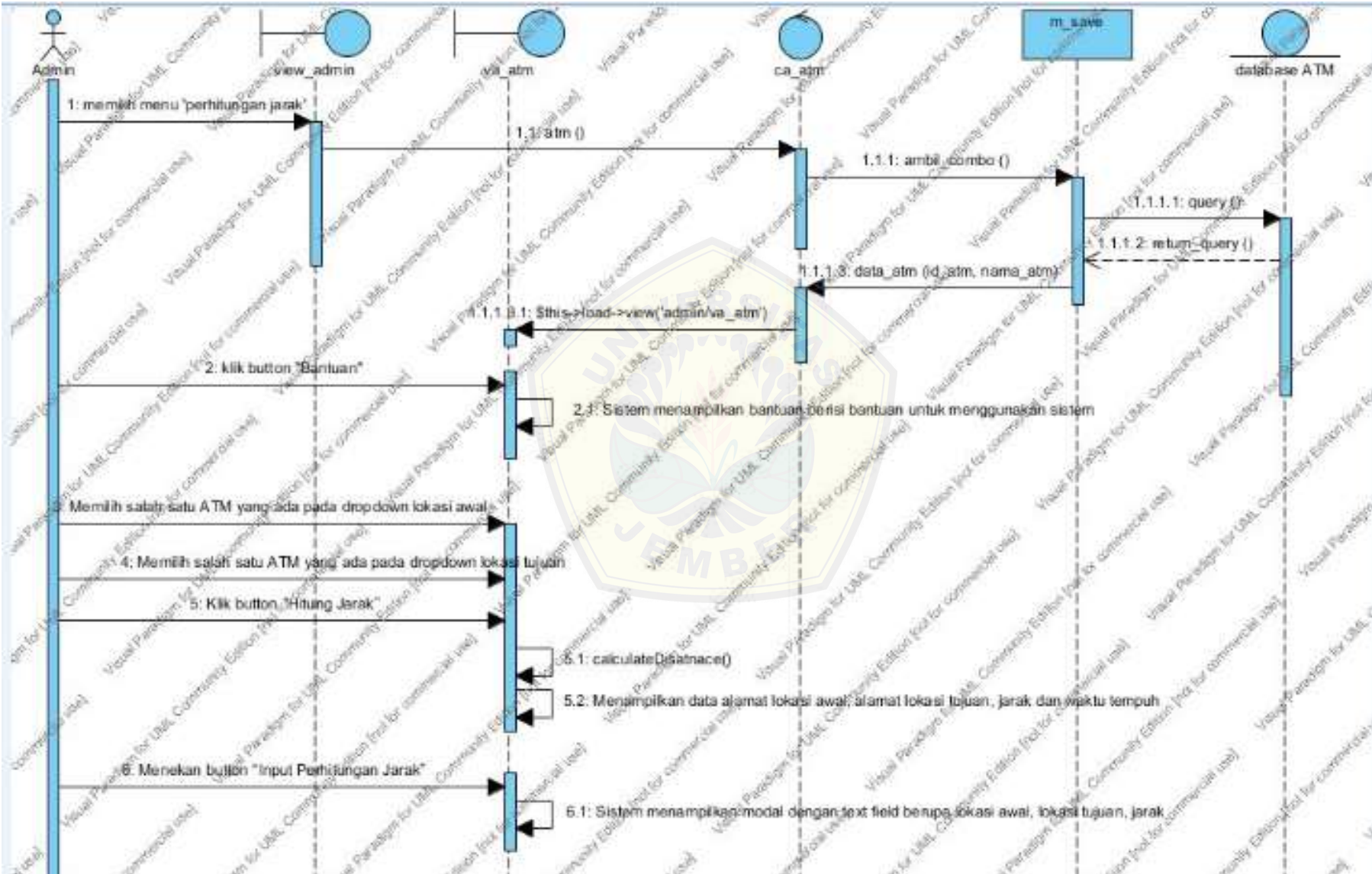
3.4. Sequence Diagram Melihat Peta ATM User

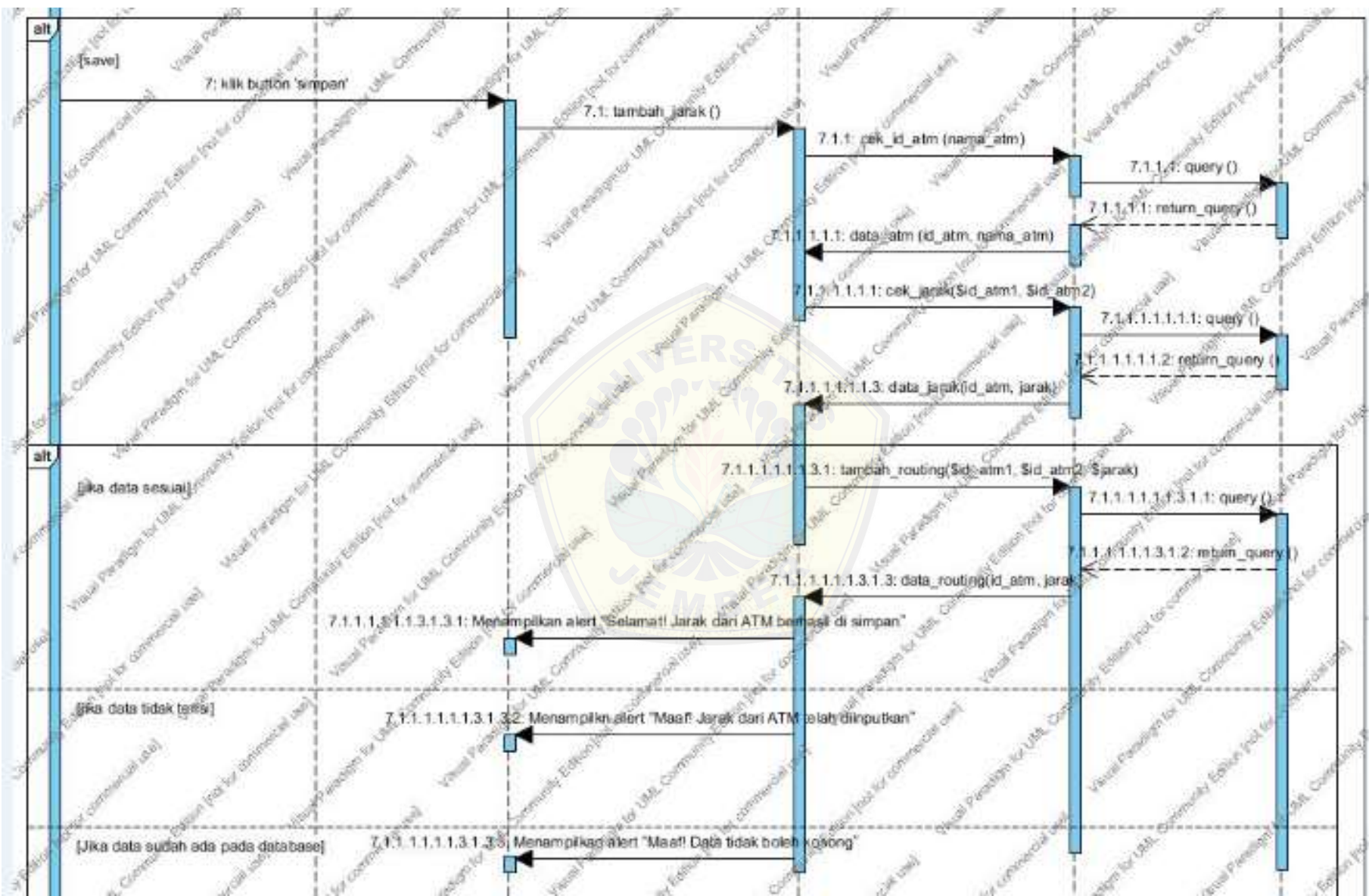


3.5. Sequence Diagram Melihat Data BRI



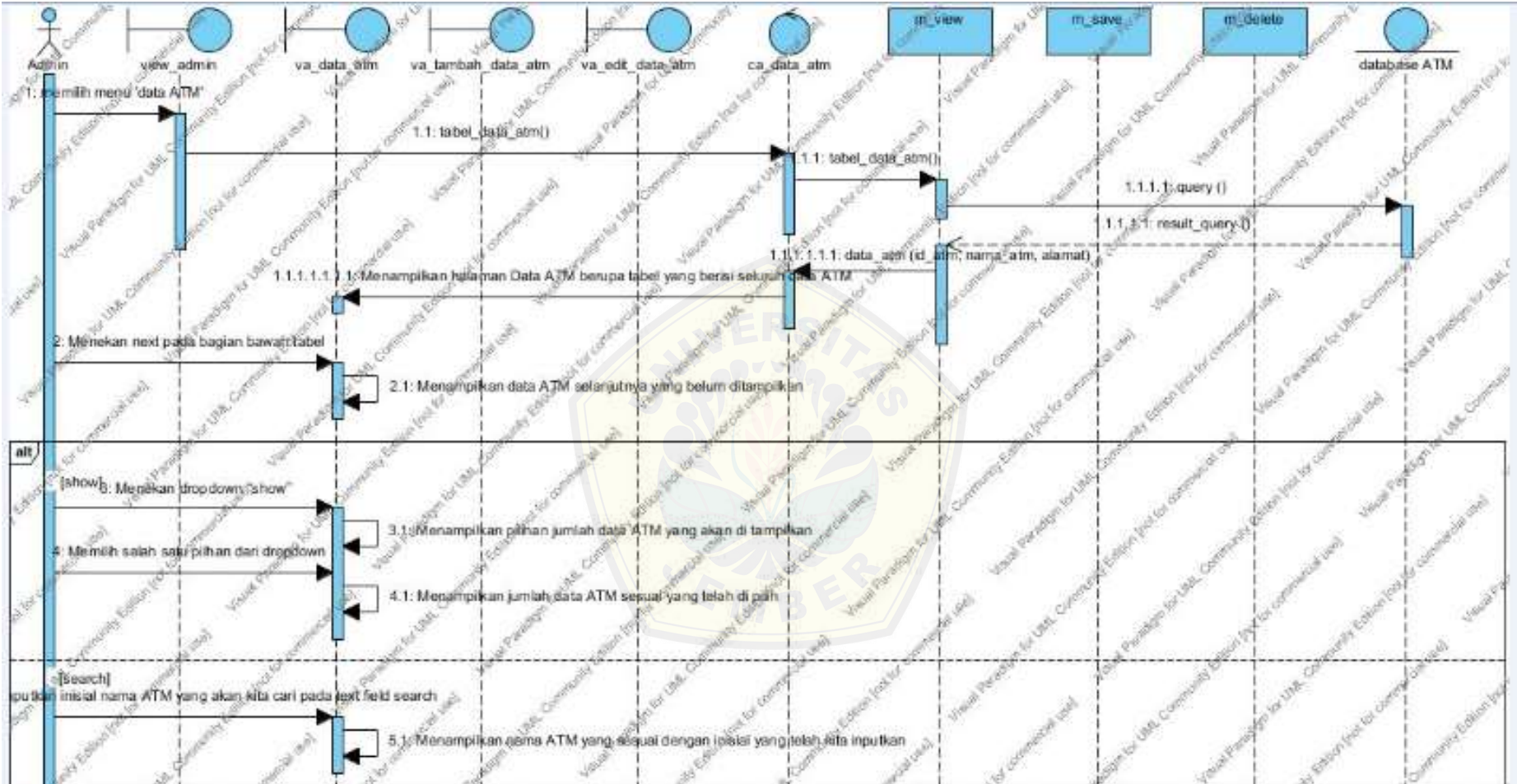
3.6. Sequence Diagram Perhitungan Jarak

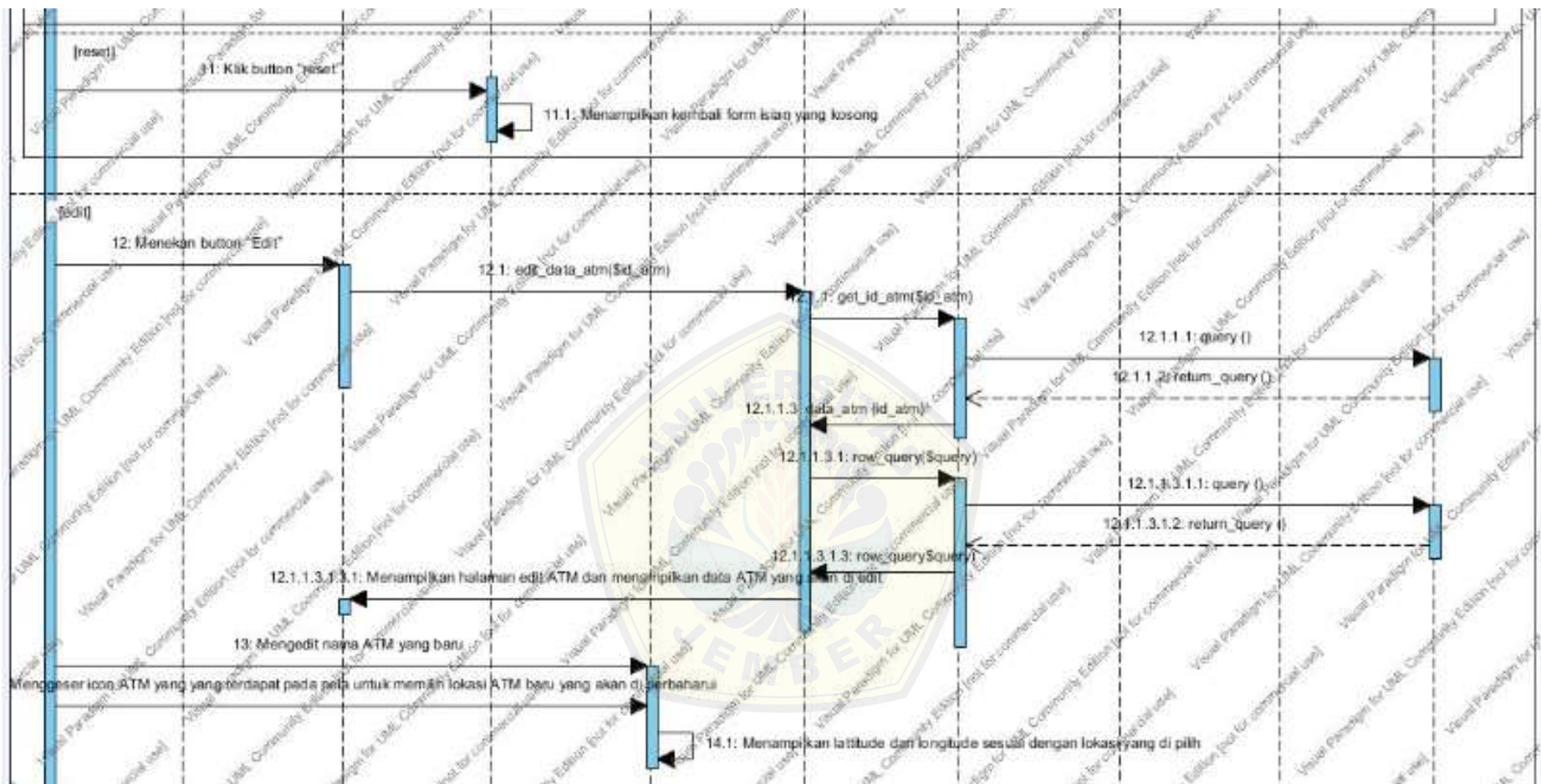


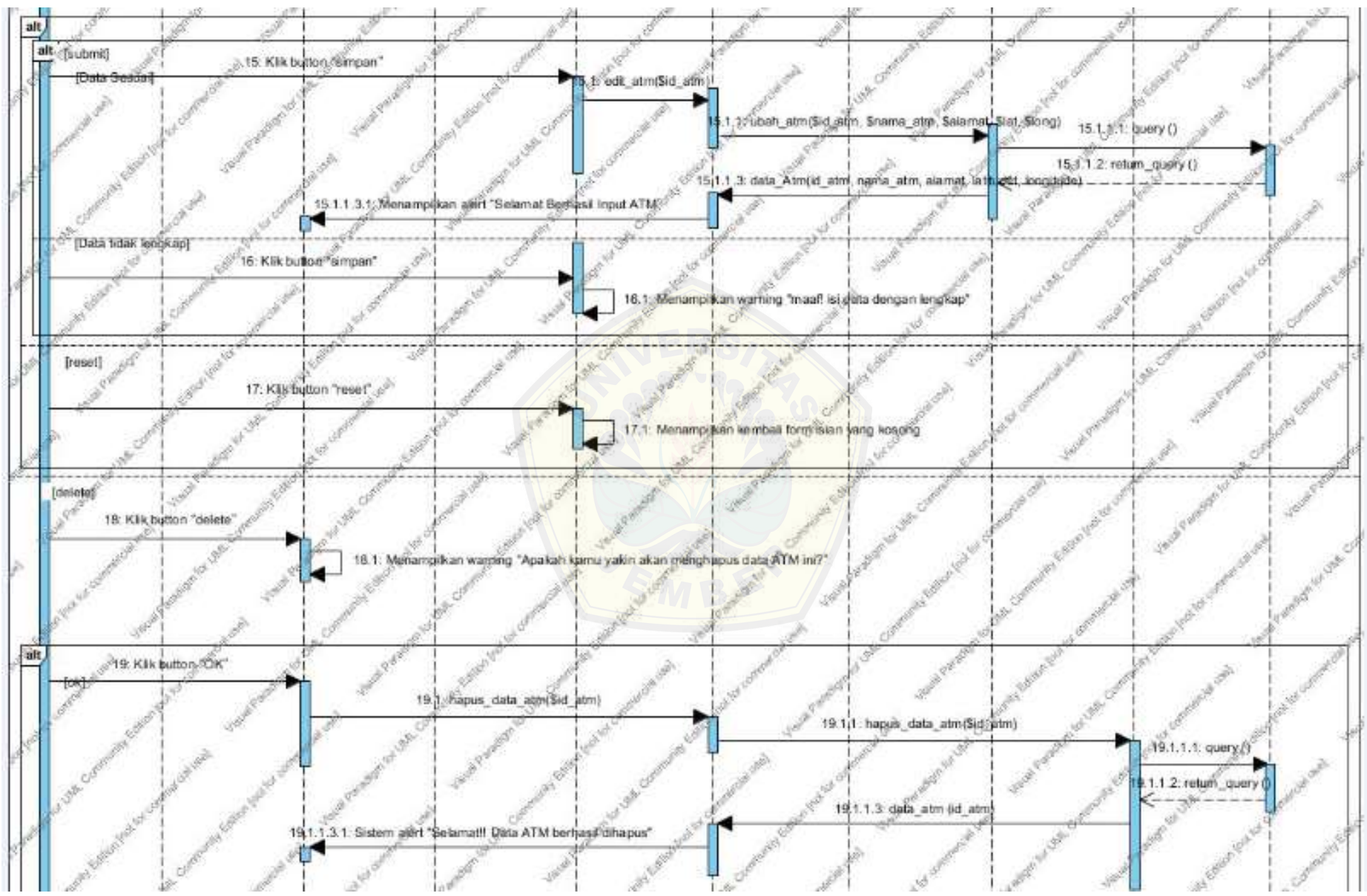




3.7. Sequence Diagram Manajemen Data ATM









4. LAMPIRAN D (Kode Program)

4.1. Kode Program Pada *Package Controllers*

1. *Class Ca_atm*

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

class Ca_atm extends CI_Controller {

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

    // untuk menu admin atm
    public function atm() {
        $this->load->library('googlemaps');
        $config['center'] = '-8.239383846027533, 113.6823472357595';
        $config['zoom'] = 10;
        $this->googlemaps->initialize($config);

        $data1['map'] = $this->googlemaps->create_map();
        $data1['option'] = $this->m_view->ambil_combo('tb_atm');
        $data['content'] = $this->load->view('admin/va_atm', $data1, true);
        $this->load->view('layout/temp_admin', $data);
    }

    // proses modal tambah jarak
    public function tambah_jarak() {
        $this->form_validation->set_rules('nama_atm1', 'nama_atm1', 'required');
        $this->form_validation->set_rules('nama_atm2', 'nama_atm2', 'required');
        $this->form_validation->set_rules('jarak', 'jarak', 'required');

        if ($this->form_validation->run()) {
            $nama1 = $this->input->post('nama_atm1', true);
            $nama2 = $this->input->post('nama_atm2', true);
            $jarak = $this->input->post('jarak', true);

            $a = $this->m_save->cek_id_atm($nama1);
            $b = $this->m_save->cek_id_atm($nama2);
            $sid_atm1 = $a->id_atm;
            $sid_atm2 = $b->id_atm;
            $cek = $this->m_save->cek_jarak($sid_atm1, $sid_atm2);

            if ($cek->jarak == null) {
                $query = $this->m_save->tambah_routing($sid_atm1, $sid_atm2, $jarak);
            }
            if ($query) {
                $this->session->set_userdata('operation', "sukses");
                $this->session->set_userdata('message', "Jarak dari ATM berhasil di simpan");
                redirect("admin/ca_atm/atm");
            } else {
                $this->session->set_userdata('operation', "gagal");
                $this->session->set_userdata('message', "Jarak dari ATM telah diinputkan");
                redirect("admin/ca_atm/atm");
            }
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Data tidak boleh kosong");
            redirect("admin/ca_atm/atm");
        }
    }
}
```


2. Class *Ca_data_atm*

```
<?php

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

class Ca_data_atm extends CI_Controller {

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

    // untuk menu admin data atm
    public function tabel_data_atm() {
        $data1['nilai'] = $this->m_view->tabel_data_atm();
        $data['content'] = $this->load->view('admin/va_data_atm', $data1, true);
        $this->load->view('layout/temp_admin', $data);
    }

    // untuk menampilkan admin tambah data atm
    public function tambah_data_atm() {
        $this->load->library('googlemaps');

        $config['center'] = '-8.17550025790394, 113.698826727947';
        $config['zoom'] = 10;
        $this->googlemaps->initialize($config);

        $marker = array();
        $marker['position'] = '-8.17550025790394, 113.698826727947';
        $marker['draggable'] = true;
        $marker['ondragend'] = '$("#lat").val(event.latLng.lat()); $("#long").val(event.latLng.lng());';

        $this->googlemaps->add_marker($marker);

        $data1['map'] = $this->googlemaps->create_map();
        $data['content'] = $this->load->view('admin/va_tambah_data_atm', $data1, true);
        $this->load->view('layout/temp_admin', $data);
    }

    // untuk menampilkan admin edit data atm
    public function edit_data_atm($id_atm) {
        $data_atm = $this->m_view->get_id_atm($id_atm);
        $this->load->library('googlemaps');
```

```

$lat = $data_atm->latitude;
$long = $data_atm->longitude;
$config['center'] = "$lat, $long";
$config['zoom'] = 15;
$this->googlemaps->initialize($config);

$skategori = $this->m_view->tampil_edit_peta($id_atm);

foreach ($skategori as $value) {
    $markerSkh = array();
    $markerSkh ['position'] = "$value->lat, $value->long";
    $markerSkh ['infowindow_content'] = "$value->nama_atm <br> $value->alamat";
    $this->googlemaps->add_marker($markerSkh);
}

$data1['map'] = $this->googlemaps->create_map();
$data1['data_atm'] = $this->m_view->row_query("select * from tb_atm where id_atm = " . $id_atm . " ");
$data1['data_id_atm'] = $this->m_view->get_id_atm($id_atm);
$data1['content'] = $this->load->view('admin/va_edit_data_atm', $data1, true);
$this->load->view('layout/temp_admin', $data);
}

// untuk menampilkan admin hapus data atm
public function hapus_data_atm($id_atm) {
    $query = $this->m_delete->hapus_data_atm($id_atm);
    if ($query) {
        $this->session->set_userdata('operation', "sukses");
        $this->session->set_userdata('message', "Data ATM berhasil dihapus");
        redirect("admin/ca_data_atm/tabel_data_atm");
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "Data ATM tidak berhasil dihapus");
        redirect("admin/ca_data_atm/tabel_data_atm");
    }
}

// untuk proses tambah atm
public function tambah_atm() {
    $this->form_validation->set_rules('id_atm', 'id_atm', 'required');
    $this->form_validation->set_rules('nama_atm', 'nama_atm', 'required');
    $this->form_validation->set_rules('alamat', 'alamat', 'required');
    $this->form_validation->set_rules('lat', 'lat', 'required');
    $this->form_validation->set_rules('long', 'long', 'required');
    if ($this->form_validation->run()) {
        $id_atm = $this->input->post('id_atm', true);
    }
}

```

```
$nama_atm = $this->input->post('nama_atm', true);
$alamat = $this->input->post('alamat', true);
$lat = $this->input->post('lat', true);
$long = $this->input->post('long', true);
$cek = $this->m_save->cek_tambah($id_atm);
if ($cek->id_atm == null) {
    $query = $this->m_save->tambah_atm($id_atm, $nama_atm, $alamat, $lat, $long);
}
if ($query) {
    $this->session->set_userdata('operation', "sukses");
    $this->session->set_userdata('message', "Berhasil Input ATM");
    redirect("admin/ca_data_atm/tabel_data_atm");
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Gagal Input ATM");
    redirect("admin/ca_data_atm/tambah_data_atm");
}
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Isi data dengan lengkap");
    redirect("admin/ca_data_atm/tambah_data_atm");
}
}

// untuk proses edit atm
public function edit_atm($id_atm) {
    $this->form_validation->set_rules('id_atm', 'id_atm', 'required');
    $this->form_validation->set_rules('nama_atm', 'nama_atm', 'required');
    $this->form_validation->set_rules('alamat', 'alamat', 'required');
    $this->form_validation->set_rules('lat', 'lat', 'required');
    $this->form_validation->set_rules('long', 'long', 'required');
    if ($this->form_validation->run()) {
        $id_atm = $this->input->post('id_atm', true);
        $nama_atm = $this->input->post('nama_atm', true);
        $alamat = $this->input->post('alamat', true);
        $lat = $this->input->post('lat', true);
        $long = $this->input->post('long', true);
        $query = $this->m_save->ubah_atm($id_atm, $nama_atm, $alamat, $lat, $long);
        if ($query) {
            $this->session->set_userdata('operation', "sukses");
            $this->session->set_userdata('message', "Berhasil mengubah data ATM");
            redirect("admin/ca_data_atm/tabel_data_atm");
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Gagal mengubah data ATM");
        }
    }
}
```

```

        redirect("admin/ca_data_atm/edit_data_atm/$id_atm");
    }
} else {
    $this->session->set_userdata('operation', "gagal");
    $this->session->set_userdata('message', "Isi data dengan lengkap");
    redirect("admin/ca_data_atm/edit_data_atm/$id_atm");
}
}
}
}

```

3. Class Ca_index

```

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

class Ca_index extends CI_Controller {

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

    // untuk menu admin index
    public function index() {
        $this->load->library('googlemaps');
        $config['center'] = '-8.239383846027533, 113.6823472357595';
        $config['zoom'] = 10;
        $this->googlemaps->initialize($config);

        $kategori = $this->m_view->tampil_peta();

        foreach ($kategori as $value) {
            $markerSkjh = array();
            $markerSkjh ['position'] = "$value->lat, $value->long";
            $markerSkjh ['infowindow_content'] = "$value->nama_atm <br> $value->alamat";
            $this->googlemaps->add_marker($markerSkjh);
        }

        $data1['map'] = $this->googlemaps->create_map();
        $data ['content'] = $this->load->view('admin/va_index', $data1, true);
        $this->load->view('layout/temp_admin', $data);
    }

    // untuk menu logout
    public function logout() {
        $this->session->sess_destroy();
        redirect('user/cu_index');
    }
}

```

4. Class Cu_bri

```

<?php

if (!defined('BASEPATH'))

```

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

class Cu_bri extends CI_Controller {

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

    // untuk menu user sejarah
    public function sejarah() {
        $data['content'] = $this->load->view('user/vu_sejarah', "", true);
        $this->load->view('layout/temp_user', $data);
    }

    // untuk menu user visimisi
    public function visimisi() {
        $data['content'] = $this->load->view('user/vu_visimisi', "", true);
        $this->load->view('layout/temp_user', $data);
    }
}
```

5. Class Cu_index

```
<?php

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

class Cu_index extends CI_Controller {

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

    // untuk menu index user
    public function index() {
        $this->load->library('googlemaps');
        $config['center'] = '-8.239383846027533, 113.6823472357595';
        $config['zoom'] = 10;
        $this->googlemaps->initialize($config);

        $kategori = $this->m_view->tampil_peta();

        foreach ($kategori as $value) {
            $markerSkh = array();
```

```

    $markerSklh ['position'] = "$value->lat, $value->long";
    $markerSklh ['infowindow_content'] = "$value->nama_atm </br> $value->alamat";
    $this->googlemaps->add_marker($markerSklh);
}

$data1['map'] = $this->googlemaps->create_map();
$data ['content'] = $this->load->view('user/vu_index', $data1, true);
$this->load->view('layout/temp_user', $data);
}

// untuk login
public function login() {
    $data['content'] = $this->load->view('user/vu_login', "", true);
    $this->load->view('layout/temp_login', $data);
}
}

```

6. Class Cu_login

```

<?php

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

class Cu_login extends CI_Controller {

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

    public function index() {
        $type = $this->session->userdata('username');
        $login = $this->session->userdata('login');
        if ($login != "") {

            redirect('admin/ca_index/index');
        } else {
            redirect('user/cu_index/login');
        }
    }

// untuk proses login
public function proses() {
    $this->form_validation->set_rules('username', 'username', 'required');
}

```

```

$this->form_validation->set_rules('password', 'password', 'required');
if ($this->form_validation->run()) {
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $query = $this->m_login->login($username, $password);
    if ($query == true) {
        redirect('user/cu_login');
    } else {
        $this->session->set_userdata('operation', "gagal");
        $this->session->set_userdata('message', "username' dan 'password' tidak terdaftar");
        redirect('user/cu_login');
    }
} else {
    $this->session->set_userdata('operation', "validasi");
    $this->session->set_userdata('message', "Isi dengan lengkap 'username' dan 'password'");
    redirect('user/cu_login');
}
}
}

```

4.2. Kode Program Pada *Package Models*

1. *Class m_delete*

```

<?php

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

class M_delete extends CI_Model {

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

    // hapus atm
    function hapus_data_atm($id_atm) {
        $query = $this->db->delete('tb_atm', array('id_atm' => $id_atm));
        if ($query) {
            return true;
        } else {
            return false;
        }
    }
}

```

2. *Class m_login*

```
<?php

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

class M_login extends CI_Model {

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

    // proses login
    function login($username, $password) {
        $this->db->where('username', $username);
        $this->db->where('password', $password);
        $query = $this->db->get('tb_admin');
        if ($query->num_rows > 0) {
            foreach ($query->result() as $row) {
                $data_ambil['login'] = 'ada';
                $this->session->set_userdata($data_ambil);
            }
            return true;
        } else {
            return false;
        }
    }
}
```

3. *Class m_save*

```
<?php

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

class M_save extends CI_Model {

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



```
// cek id pada ubah jarak
function cek_id_atm($nama) {
    $query = $this->db->query("select id_atm from tb_atm where nama_atm='$nama'");
    return $query->row();
}

//cek jarak pada routing atm
function cek_hasil_jarak($id1,$id2) {
    $query = $this->db->query("select jarak from tb_routing where id_atm1='$id1' and id_atm2='$id2'");
    return $query->row();
}

// cek jarak pada ubah jarak
function cek_jarak($id_atm1, $id_atm2) {
    $query = $this->db->query("select jarak from tb_routing where id_atm1='$id_atm1' and id_atm2='$id_atm2'");
    return $query->row();
}

// cek id atm pada tambah atm
function cek_tambah($id_atm) {
    $query = $this->db->query("select id_atm from tb_atm where id_atm='$id_atm'");
    return $query->row();
}

// tambah routing
function tambah_routing($id_atm1, $id_atm2, $jarak) {
    $this->db->set('id_atm1', $id_atm1);
    $this->db->set('id_atm2', $id_atm2);
    $this->db->set('jarak', $jarak);
    $query = $this->db->insert('tb_routing');
    if ($query) {
        return true;
    } else {
        return false;
    }
}

// tambah atm
function tambah_atm($id_atm, $nama_atm, $alamat, $lat, $long) {
    $query = $this->db->query("INSERT INTO tb_atm(id_atm, nama_atm, alamat, latitude, longitude ,geom)
        VALUES ('$id_atm', '$nama_atm', '$alamat', '$lat', '$long', ST_GeomFromText('POINT ($long $lat)'))");
    if ($query) {
        return true;
    } else {
        return false;
    }
}
```

```

    }
}

// ubah atm
function ubah_atm($id_atm, $nama_atm, $alamat, $lat, $long) {
    $query = $this->db->query("UPDATE tb_atm SET id_atm = '$id_atm', nama_atm = '$nama_atm', alamat = '$alamat',
latitude = '$lat', longitude = '$long', geom = ST_GeomFromText('POINT ($long $lat)') where id_atm = $id_atm ");

    if ($query) {
        return true;
    } else {
        return false;
    }
}

// tambah rute
function simpan_rute($nomer, $t_hasil) {
    $query = $this->db->query("INSERT INTO tb_hasil(nomer, id_atm)
VALUES ('$nomer', '$t_hasil')");

    if ($query) {
        return true;
    } else {
        return false;
    }
}
}

```

4. *Class m_umum*

```

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

class C_login extends CI_Controller {

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

    public function index() {
        $type = $this->session->userdata('id_hak_akses');
        $login = $this->session->userdata('login');
        if ($login != "") {
            if ($type == '1' || $type == '2') {
                redirect("c_home/home/$type");
            } else {
                redirect('c_login/logout');
            }
        }
        $this->load->view('login');
    }
}

```

```

}

public function proses() {
    $this->form_validation->set_rules('username', 'username', 'required');
    $this->form_validation->set_rules('password', 'password', 'required');
    if ($this->form_validation->run()) {
        $username = $this->input->post('username');
        $password = md5($this->input->post('password'));
        $query = $this->m_login->login($username, $password);
        if ($query == true) {
            redirect(base_url());
        } else {
            $this->session->set_userdata('operation', "gagal");
            $this->session->set_userdata('message', "Username dan Password tidak ditemukan");
            redirect(base_url());
        }
    } else {
        $this->session->set_userdata('operation', "validasi");
        $this->session->set_userdata('message', validation_errors());
        redirect(base_url());
    }
}

public function logout() {
    $this->session->sess_destroy();
    $this->session->unset_userdata('username');
    $this->session->unset_userdata('login');
    $this->session->unset_userdata('type');
    $this->session->unset_userdata('status');
    redirect(base_url());
}
}

```

5. *Class m_view*

```

<?php

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

class M_view extends CI_Model {

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

    // tampil peta
    function tampil_peta() {
        $query = $this->db->query('select nama_atm, alamat, st_y(st_astext(tb_atm.geom)) as lat, st_x(st_astext(tb_atm.geom))
as long from tb_atm');
        return $query->result();
    }

    // tampil peta hasil rute
    function tampil_rute() {
        $query = $this->db->query('select a.id_atm, b.alamat,b.nama_atm, st_y(st_astext(b.geom)) as lat,
st_x(st_astext(b.geom)) as long from tb_hasil a join tb_atm b on a.id_atm = b.id_atm');
        return $query->result();
    }

    // tampil peta edit atm
    function tampil_edit_peta($id_atm) {
        $query = $this->db->query('select nama_atm, alamat, st_y(st_astext(tb_atm.geom)) as lat, st_x(st_astext(tb_atm.geom))
as long from tb_atm where id_atm = ' . $id_atm);
    }
}

```

```
        return $query->result();
    }

// tampil peta
function tambah_marker($data) {
    $atm = array();
    for ($i = 0, $ln = count($data); $i < $ln; $i++) {
        $query = $this->db->query('select nama_atm, st_y(st_astext(tb_atm.geom)) as lat, st_x(st_astext(tb_atm.geom)) as
long from tb_atm where id_atm = ' . $data[$i]);
        $tmp = $query->result_array();
        array_push($atm, array("nama_atm" => $tmp[0]['nama_atm'], "lat" => $tmp[0]['lat'], "long" => $tmp[0]['long']));
    }
    return $atm;
}

// ambil combo atm
function ambil_combo($temp, $id = "") {
    if ($id == "") {
        $query = $this->db->query('select * from ' . $temp . ' order by id_atm asc');
        return $query->result();
    } else {
        $query = $this->db->query('select * from ' . $temp . ' where id_atm = ' . $id);
        return $query->result_array();
    }
}

// tabel data atm
function tabel_data_atm() {
    $query = $this->db->query('select * from tb_atm order by id_atm asc');
    return $query->result();
}

// id atm
function get_id_atm($id_atm) {
    $this->db->where('id_atm', $id_atm);
    $query = $this->db->get('tb_atm');
    return $query->row();
}

// row query
function row_query($query) {
    $query = $this->db->query($query);
    return $query->row();
}

// jarak routing
function routing($atm1, $atm2) {
    $this->db->where(array('id_atm1' => $atm1, 'id_atm2' => $atm2));
    return $this->db->get('tb_routing')->result_array();
}
}
```

5. LAMPIRAN E (Pengujian *Black Box*)

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Login	Menjalankan sistem	a) Menampilkan halaman <i>dashboard user</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik <i>button</i> “ <i>Login</i> ”	b) Menampilkan Form <i>Login</i>	
	Sign In	Memasukkan <i>Username</i> dan <i>Password</i> dan klik <i>button</i> “ <i>Login</i> ”	a) Melakukan autentifikasi ke dalam <i>database</i> b) Menampilkan halaman <i>dashboard admin</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik <i>button</i> “ <i>Login</i> ” ketika tidak memasukan <i>username</i> dan <i>password</i>	a) Mengecek masukan <i>username</i> dan <i>password</i> pada <i>database</i> b) Menampilkan <i>alert</i> “ Maaf! Isi dengan lengkap 'username' dan 'password'” c) Pilih tanda ‘x’ atau close d) Menampilkan Form <i>Login</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik <i>button</i> “ <i>Login</i> ” ketika salah memasukan <i>username</i> dan <i>password</i>	a) Mengecek masukan <i>username</i> dan <i>password</i> pada <i>database</i> b) Menampilkan <i>alert</i> “ Maaf! 'username' dan 'password' tidak terdaftar” c) Pilih tanda ‘x’ atau close d) Menampilkan Form <i>Login</i>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Cancel	Klik <i>button</i> “ <i>Batal</i> ”	b) Menampilkan halaman <i>dashboard user</i>	<input checked="" type="checkbox"/> Berhasil

				[] Gagal
2.	Logout	Klik button “Logout” di bagian pojok kanan menu	a) Menghapus data <i>session</i> b) Menampilkan halaman <i>dashboard</i> user	[√] Berhasil [] Gagal
3.	Melihat Peta ATM Admin	Menjalankan system	a) Mengambil data peta ATM dari database b) Menampilkan halaman <i>dashboard admin</i> yang terdapat peta yang berisi data ATM BRI dan keterangan sistem	[√] Berhasil [] Gagal
	Melihat Data Detail ATM Admin	Memilih salah satu <i>icon</i> ATM yang terdapat pada peta	a) Mengambil data detail ATM dari database b) Menampilkan nama dan alamat detail ATM sesuai dengan yang telah dipilih	[√] Berhasil [] Gagal
		Klik ‘x’ atau ‘close’ pada info yang tampil	a) Menampilkan halaman <i>dashboard admin</i>	[√] Berhasil [] Gagal
4.	Melihat Peta ATM User	Menjalankan system	a) Mengambil data peta ATM dari database b) Menampilkan halaman <i>dashboard user</i> yang terdapat peta yang berisi data ATM BRI dan keterangan sistem	[√] Berhasil [] Gagal
	Melihat Data Detail ATM User	Memilih salah satu <i>icon</i> ATM yang terdapat pada peta	a) Mengambil data detail ATM dari database b) Menampilkan nama dan alamat detail ATM sesuai dengan yang telah dipilih	[√] Berhasil [] Gagal
		Klik ‘x’ atau ‘close’ pada info yang tampil	a) Menampilkan halaman <i>dashboard user</i>	[√] Berhasil [] Gagal

5.	Melihat Data BRI	Memilih menu “Tentang BRI”	a) Menampilkan <i>dropdown</i> menu “Sejarah BRI” dan “Visi dan Misi”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Memilih menu “Sejarah BRI”	a) Menampilkan informasi tentang sejarah berdirinya bank BRI	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Memilih menu “visi & misi”	a) Menampilkan informasi tentang visi & misi bank BRI	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
6.	Perhitungan Jarak	Memilih menu “Perhitungan Jarak”	a) Menampilkan halaman Perhitungan Jarak yang berisi pilihan Lokasi dan Bantuan. Lokasi berisi <i>dropdown</i> pilihan Tambah Lokasi Awal dan Tambah Lokasi Tujuan dan juga peta bank BRI di kabupaten Jember dan Bantuan berisi bantuan untuk menggunakan sistem	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik button “Bantuan”	a) Sistem menampilkan bantuan berisi bantuan untuk menggunakan sistem	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Pilih Lokasi ATM	Memilih salah satu ATM yang ada pada <i>dropdown</i> lokasi awal Memilih salah satu ATM yang ada pada <i>dropdown</i> lokasi tujuan	a) Menampilkan data alamat lokasi awal, alamat lokasi tujuan, jarak dan waktu tempuh	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		Klik button " <i>Hitung Jarak</i> "		
	Input Perhitungan Jarak	Menekan <i>button</i> "Input Perhitungan Jarak"	a) Sistem menampilkan <i>modal</i> dengan text field berupa lokasi awal, lokasi tujuan, jarak	[√] Berhasil [] Gagal
	Simpan Perhitungan Jarak	Menekan <i>button</i> "simpan"	a) Melakukan pengecekan pada database b) Menyimpan data pada database c) Menampilkan <i>alert</i> " Selamat! Jarak dari ATM berhasil di simpan "	[√] Berhasil [] Gagal
		Klik tanda 'x' atau <i>close</i> pada alert	a) Menampilkan halaman Perhitungan Jarak	[√] Berhasil [] Gagal
	Simpan Perhitungan Jarak	Menekan <i>button</i> "simpan" ketika jarak ATM yang diinputkan telah ada pada database	a) Melakukan pengecekan pada database b) Menyimpan data pada database c) Menampilkan halaman <i>Calculate Distance</i> dan <i>alert</i> " Maaf! Jarak dari ATM telah diinputkan "	[√] Berhasil [] Gagal
		Klik tanda 'x' atau <i>close</i> pada alert	a) Menampilkan halaman Perhitungan Jarak	[√] Berhasil [] Gagal
	Simpan Perhitungan Jarak	Menekan <i>button</i> "simpan" ketika ada data yang kosong	a) Melakukan pengecekan pada database b) Menyimpan data pada database c) Menampilkan <i>alert</i> " Maaf! Data tidak boleh kosong "	[√] Berhasil [] Gagal

		Klik tanda 'x' atau <i>close</i> pada alert	d) Menampilkan halaman Perhitungan Jarak	
	Batal Perhitungan Jarak	Menekan <i>button</i> "batal"	a) Menampilkan halaman Perhitungan Jarak	[√] Berhasil [] Gagal
7.	Manajemen Data ATM	Memilih menu "Data ATM"	a) Menampilkan halaman Data ATM berupa tabel yang berisi seluruh data ATM	[√] Berhasil [] Gagal
		Menekan <i>next</i> pada bagian bawah tabel	a) Menampilkan data ATM selanjutnya yang belum ditampilkan	[√] Berhasil [] Gagal
	Show ATM	Menekan <i>dropdown</i> "show"	a) Menampilkan pilihan jumlah data ATM yang akan di tampilkan	[√] Berhasil [] Gagal
		Memilih salah satu pilhan dari <i>dropdown</i>	a) Menampilkan jumlah data ATM sesuai yang telah di pilih	[√] Berhasil [] Gagal
	Search	Menginputkan inisial nama ATM yang akan kita cari pada <i>text field search</i>	a) Menampilkan nama ATM yang sesuai dengan inisial yang telah kita inputkan	[√] Berhasil [] Gagal
	Add New ATM	Menekan button "Tambah ATM Baru"	a) Menampilkan halaman tambah ATM baru	[√] Berhasil [] Gagal
	Add Submit	Menginputkan ID ATM, Nama ATM dan alamat ATM yang baru yang di inputkan	a) Menampilkan <i>latitude</i> dan <i>longitude</i> sesuai dengan lokasi yang di pilih	[√] Berhasil [] Gagal

		Menggeser <i>icon</i> ATM yang yang terdapat pada peta untuk memilih lokasi ATM baru yang akan di buat		
		Klik <i>button</i> “simpan”	a) Menyimpan data pada database b) Menampilkan halaman Data ATM dan <i>alert</i> “ Selamat Berhasil Input ATM ”	[√] Berhasil [] Gagal
		Klik <i>button</i> “simpan” ketika ada data yang kosong	a) Menyimpan data pada database b) Menampilkan halaman Add New ATM dan <i>alert</i> “ Maaf! Isi data dengan lengkap”	[√] Berhasil [] Gagal
	Add Reset	Klik <i>button</i> “reset”	a) Menampilkan kembali form isian yang kosong	[√] Berhasil [] Gagal
	Edit ATM	Menekan <i>button</i> “Edit”	a) Menampilkan halaman <i>edit</i> ATM dan menampilkan data ATM yang akan di <i>edit</i>	[√] Berhasil [] Gagal
	Edit Submit	Mengedit nama ATM dan alamat ATM yang baru Menggeser <i>icon</i> ATM yang yang terdapat pada peta untuk memilih lokasi ATM baru yang akan di perbaharui	a) Menampilkan <i>latitude</i> dan <i>longitude</i> sesuai dengan lokasi yang di pilih	[√] Berhasil [] Gagal
		Klik <i>button</i> “simpan”	a) Menyimpan data pada database	[√] Berhasil

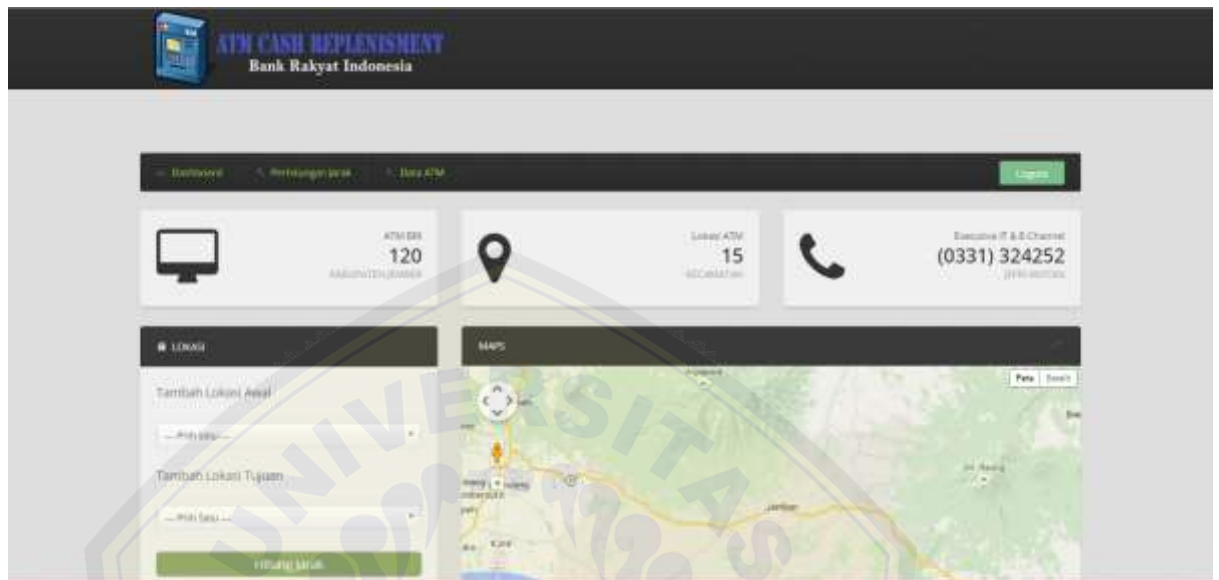
			b) Menampilkan halaman Data ATM dan <i>alert</i> “ Selamat Berhasil Input ATM ”	[] Gagal
		Klik <i>button</i> “simpan”	a) Menyimpan data pada database b) Menampilkan halaman Add New ATM dan <i>alert</i> “ Maaf! Isi data dengan lengkap”	[√] Berhasil [] Gagal
	Edit Reset	Klik <i>button</i> “reset”	a) Menampilkan kembali form isian yang kosong	[√] Berhasil [] Gagal
	Delete ATM	Klik <i>button</i> “delete”	a) Menampilkan <i>warning</i> “Apakah kamu yakin akan menghapus data ATM ini?”	[√] Berhasil [] Gagal
	OK Delete ATM	Klik <i>button</i> “OK”	a) Data terhapus dari database b) Sistem menampilkan halaman data ATM dan <i>alert</i> “ Selamat!! Data ATM berhasil dihapus”	[√] Berhasil [] Gagal
	Cancel Delete ATM	Klik <i>button</i> “batal”	a) Sistem menampilkan halaman data ATM	[√] Berhasil [] Gagal
	Manajemen Data ATM	Memilih menu “Data ATM”	a) Menampilkan halaman Data ATM berupa tabel yang berisi seluruh data ATM	[√] Berhasil [] Gagal
		Menekan <i>next</i> pada bagian bawah tabel	a) Menampilkan data ATM selanjutnya yang belum ditampilkan	[√] Berhasil [] Gagal
	Show ATM	Menekan <i>dropdown</i> “show”	a) Menampilkan pilihan jumlah data ATM yang akan di tampilkan	[√] Berhasil [] Gagal

		Memilih salah satu pilhan dari <i>dropdown</i>	b) Menampilkan jumlah data ATM sesuai yang telah di pilih	
--	--	--	---	--

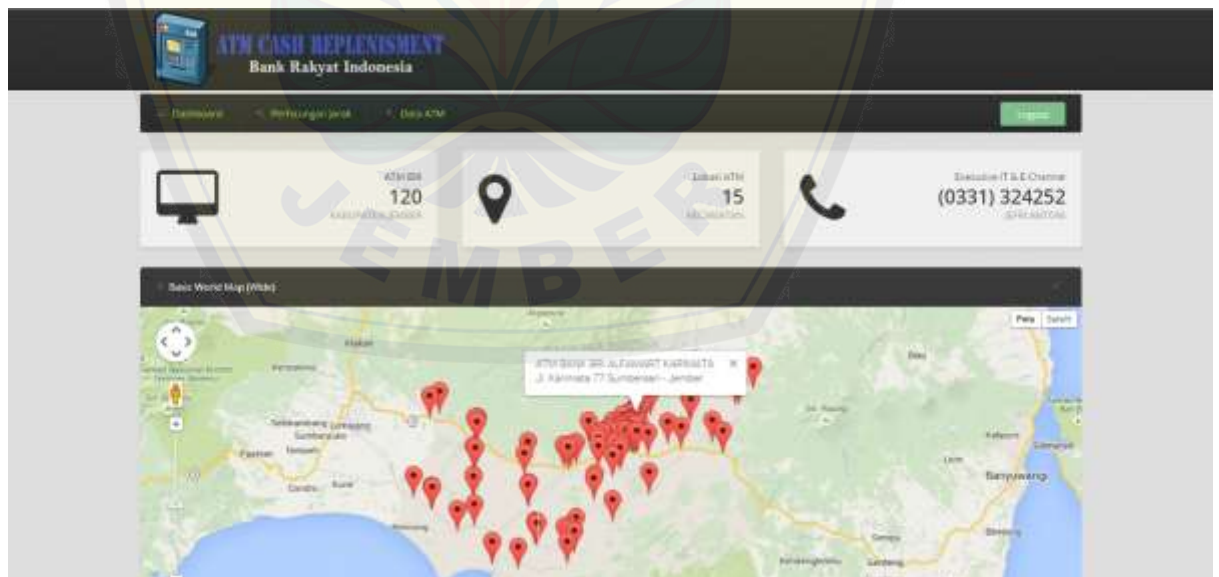


6. LAMPIRAN F (Hasil Pembuatan Sistem)

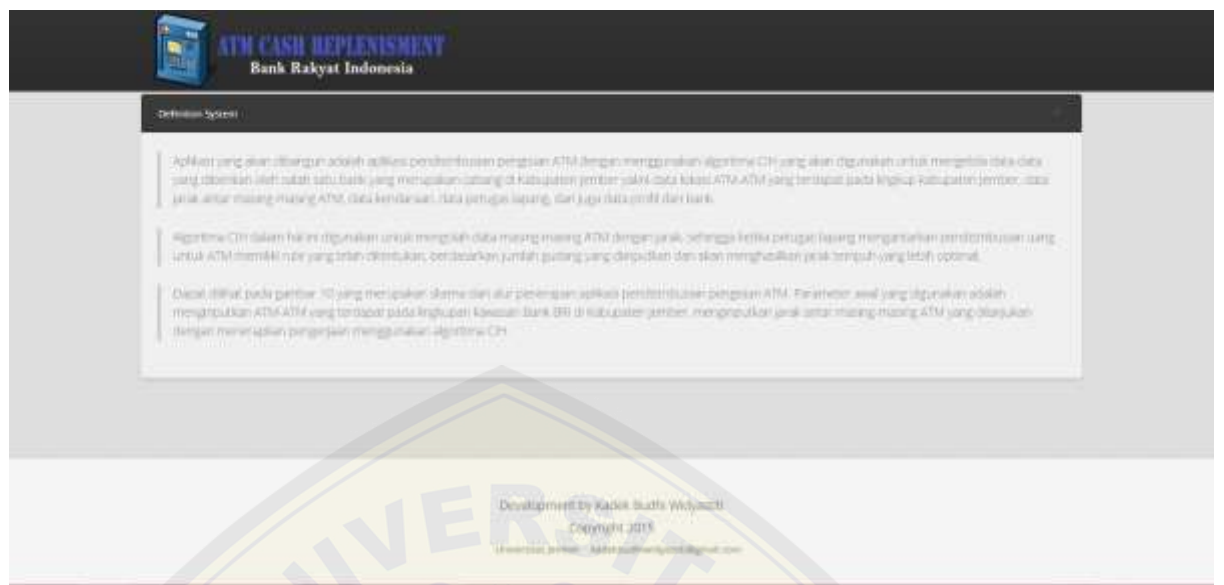
6.1 Tampilan Fitur Logout



6.2 Tampilan Fitur Melihat Peta ATM User



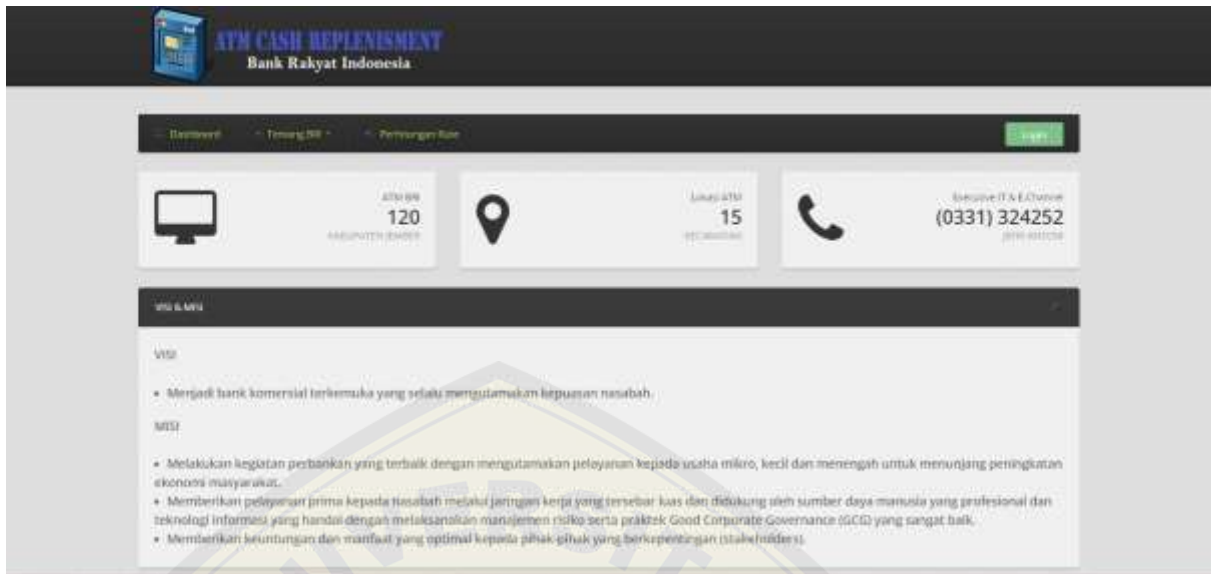
6.3 Tampilan Fitur Melihat Peta ATM User dengan Keterangan



6.4 Tampilan Fitur Melihat Data BRI Sejarah



6.5 Tampilan Fitur Melihat Data BRI Visi dan Misi



The screenshot displays the 'ATM CASH REPLENISHMENT' page for Bank Rakyat Indonesia. The page features a navigation bar with 'Beranda', 'Tentang BRI', and 'Perbankan BRI'. Below the navigation bar, there are three main service tiles: 'ATM BRI 120 RASUNTEN ENDEH', 'Lokasi ATM 15 RASUNTEN ENDEH', and 'Service IT & E-Channel (0331) 324252 (0331) 881004'. The 'Visi & Misi' section is highlighted, showing the following content:

Visi

- Menjadi bank komersial terkemuka yang selalu mengutamakan kepuasan nasabah.

Misi

- Melakukan kegiatan perbankan yang terbaik dengan mengutamakan pelayanan kepada usaha mikro, kecil dan menengah untuk menunjang pertumbuhan ekonomi masyarakat.
- Memberikan pelayanan prima kepada nasabah melalui jaringan kerja yang tersebar luas dan didukung oleh sumber daya manusia yang profesional dan teknologi informasi yang handal dengan melaksanakan manajemen risiko serta praktik Good Corporate Governance (GCG) yang sangat baik.
- Memberikan keuntungan dan manfaat yang optimal kepada pihak-pihak yang berkepentingan (stakeholders).

