

repository.ub.ac.id

**RANCANG BANGUN APLIKASI ANDROID UNTUK  
PENCARIAN LOKASI ATM TERDEKAT DI KOTA MALANG**

**SKRIPSI**

Sebagai salah satu syarat untuk memperoleh  
gelar Sarjana dalam bidang Ilmu Komputer



Disusun oleh :

**YUSTIKA RILLANINGRUM**

**NIM. 0810963077**

**PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER  
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA**

**MALANG**

**2014**



**LEMBAR PERSETUJUAN**

**RANCANG BANGUN APLIKASI ANDROID PENCARIAN LOKASI ATM  
TERDEKAT DI KOTA MALANG**

**SKRIPSI**

**LABORATORIUM**

**PEMROGRAMAN APLIKASI PERANGKAT BERGERAK**

**Untuk memenuhi persyaratan mencapai gelar Sarjana Komputer**



Disusun oleh :

**YUSTIKA RILLANINGRUM**

**NIM. 0810963077**

Skripsi ini telah diperiksa dan disetujui oleh dosen pembimbing

Pada tanggal 21 Agustus 2014

Pembimbing I,

Pembimbing II,

**Dr. Eng Herman Tolle, ST., MT**

**NIP. 197408232000121001**

**Drs. Marji, MT**

**NIP. 196708011992031001**

**LEMBAR PENGESAHAN**

**RANCANG BANGUN APLIKASI ANDROID PENCARIAN LOKASI ATM  
TERDEKAT DI KOTA MALANG**

**SKRIPSI**

Sebagai salah satu syarat untuk memperoleh  
gelar Sarjana dalam bidang Ilmu Komputer

Disusun Oleh:  
**YUSTIKA RILLANINGRUM**  
**NIM. 0810963077**

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 21 Agustus 2014

Penguji I,

Penguji II,

**Aryo Pinandito, S.T., M.MT.**  
**NIK. 83051916110374**

**Agi Putra Kharisma, S.T., M.T.**  
**NIP.**

Penguji III,

**Eriq Muhammad Adams, J.ST.,M.Kom**  
**NIP. 198504102012121001**

Mengetahui,  
Ketua Program Studi Informatika / Ilmu Komputer

**Drs. Marji, M.T.**  
**NIP. 196708011992031001**

## PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, Agustus 2014

Mahasiswa

**Yustika Rillaningrum**

**NIM. 0810963077**

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmat, pertolongan, dan petunjuk-Nya sehingga Penulis dapat menyelesaikan skripsi dengan judul : **“RANCANG BANGUN APLIKASI ANDROID UNTUK PENCARIAN LOKASI ATM TERDEKAT DI KOTA MALANG ”**.

Skripsi ini diajukan sebagai syarat ujian seminar skripsi dalam rangka untuk memperoleh gelar Sarjana Komputer di Program Studi Informatika dan Ilmu Komputer, Universitas Brawijaya Malang. Atas terselesainya skripsi ini, Penulis mengucapkan terima kasih kepada:

1. Dr. Eng Herman Tolle, ST., MT. selaku Dosen Pembimbing Skripsi I yang telah meluangkan waktu memberikan pengarahan dan bimbingan kepada penulis.
2. Drs. Marji, MT. selaku Dosen Pembimbing Skripsi II yang telah meluangkan waktu membimbing dan memberikan arahan kepada penulis dan selaku Ketua Program Studi Teknik Informas Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
3. Nurul Hidayat, S.Pd., MSc. selaku dosen pembimbing akademik, yang telah membantu dan selalu memberikan nasehat kepada penulis.
4. Segenap Bapak dan Ibu Dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya.
5. Segenap Bapak dan Ibu dosen Fakultas MIPA yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan.
6. Segenap staf dan karyawan di Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya yang telah membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
7. Segenap staf dan karyawan di Fakultas MIPA Universitas Brawijaya yang telah banyak membantu Penulis dalam menempuh pendidikan.
8. Secara khusus Penulis ingin mengucapkan banyak terima kasih kepada orang tua, Ir. Hari Prasetyo, MP dan Ir. Mamik Chuzaimah yang telah memberi

motivasi, kasih sayang serta dukungan moril dan materil kepada Penulis. Kedua saudara laki-laki Penulis, Hashfi Mafazi dan Ardhy Ihza Mahindra yang telah memberikan semangat sampai akhir pengerjaan tugas akhir ini.

9. Seluruh teman seperjuangan Ilkomp 2008 yang telah banyak memberikan segala cerita, kenangan, support, dan doa dalam menempuh pendidikan bersama-sama di Universitas Brawijaya.
10. Ferdiansyah Setyanto Andoko, yang selama ini telah memberikan support, dukungan, motivasi dan doa kepada Penulis. Dan juga kepada Pritha SAF, Mutya Fani Atsomya, Nasrul Akhmad Hidayat, Alfath Dahana, Eko Alfiyanto, Hari Prisbiantoro, Dito Ferandy Raditya, M. Rizky Indra, Pradana Tommy, Aditya Wira, Ria Kurnianti, Arianti Silvia, dan Senggek selaku guru yang selalu setia menemani, mendengarkan, memberikan motivasi dan mengajarkan semangat kepada Penulis.
11. Semua pihak yang telah membantu terselesaikannya skripsi ini yang tidak dapat kami sebutkan satu per satu.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan jauh dari sempurna, karena keterbatasan materi dan pengetahuan yang dimiliki penulis. Maka, saran dan kritik yang membangun dari semua pihak sangat diharapkan demi penyempurnaan selanjutnya. Semoga skripsi ini dapat bermanfaat dan berguna bagi semua pihak, baik penulis maupun pembaca.

Malang, Agustus 2014

Penulis

## ABSTRAK

**Yustika Rillaningrum. 2014. Rancang Bangun Aplikasi Android Untuk Pencarian Lokasi ATM Terdekat di Kota Malang.**

ATM (*Automated Teller Machine*) merupakan salah satu fasilitas umum dengan teknologi terkini yang digunakan masyarakat untuk melakukan transaksi keuangan tanpa harus pergi ke bank. Saat ini, mesin ATM telah tersebar di berbagai sudut daerah, tidak terkecuali di kota Malang. Banyaknya jumlah ATM di kota Malang dengan lokasinya yang tersebar, terkadang membuat sebagian masyarakat merasa kesulitan untuk mencarinya. Oleh karena itu penulis merancang suatu aplikasi pencarian lokasi ATM berbasis Android *mobile* dengan mengaplikasikan LBS (*Location Based Service*). Aplikasi ini berpotensi mempermudah pengguna Android *mobile* yang ada di kota Malang untuk mencari posisi saat ini, kemudian dari posisi sekarang, pengguna dapat menemukan ATM terdekat yang diinginkan. Pengembangan sistem menggunakan model *hybrid mobile development* pada Android *Smartphone*, dengan *framework* PhoneGap, HTML5, CSS3, dan JavaScript, serta penyimpanan data secara *online*. Penerapan metode Haversine pada fitur pencarian lokasi ATM terdekat juga sangat berguna dalam mencari jarak terdekat antara pengguna dengan lokasi ATM. Setelah melakukan pengujian, aplikasi pencarian lokasi ATM sudah berjalan dengan baik dan berhasil dari sisi *administrator* maupun *user*. Aplikasi pencarian lokasi ATM terdekat di kota Malang dapat berjalan pada sistem operasi Android versi 2.3.5, versi 4.1.2, dan versi 4.2.2.

**Kata kunci :** ATM, PhoneGap, *Hybrid Mobile Application*, Algoritma Haversine

## ABSTRACT

**Yustika Rillaningrum. 2014. *Design of Android Mobile Application for Search Nearby ATM Locations in Malang City.***

*ATM (Automated Teller Machine) is one of the public facilities with updated technology that allows people to conduct financial transactions without having to go to the bank. Currently, ATM machines have been scattered in various corners of the region, not least in Malang city. A large number of ATM in Malang with scattered location, sometime make some people difficult to find it. Therefore, the authors designed a Android mobile application based on Android mobile ATM locations by applying LBS (Location Based Service). This application has the potential to simplify the Android mobile users in Malang city to locate the current position, then from the current position, the user can find the nearest ATM. Development using a hybrid mobile development on Android Smartphone with framework PhoneGap, HTML5, CSS3, and JavaScript as well as online data storage. Application of Haversine method in the search nearest ATM location feature is also very useful for finding the shortest distance between user and ATM location. After testing, the application of search nearest ATM locations has been running well and managed from the administrator and user. Android Application of search nearest ATM locations in Malang can be run on the Android version 2.3.5, version 4.1.2, dan version 4.2.2.*

**Keywords :** ATM, PhoneGap, Hybrid Mobile Application, Haversine Algorithm





DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN .....	iii
PERNYATAAN ORISINALITAS SKRIPSI .....	iv
KATA PENGANTAR .....	v
ABSTRAK .....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xv
DAFTAR RUMUS .....	xvii
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	4
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1 ATM ( <i>Auto Teller Machine</i> ).....	5
2.1.1 Jenis Mesin ATM.....	5
2.1.2 Manfaat ATM.....	6
2.2 Sistem Operasi Android .....	7
2.2.1 Pengertian Android Platform .....	7
2.2.2 Fitur dan Arsitektur Android.....	8
2.3 HTML 5.....	11
2.4 CSS3.....	13
2.5 JavaScript .....	14
2.5.1 Tag JavaScript.....	14



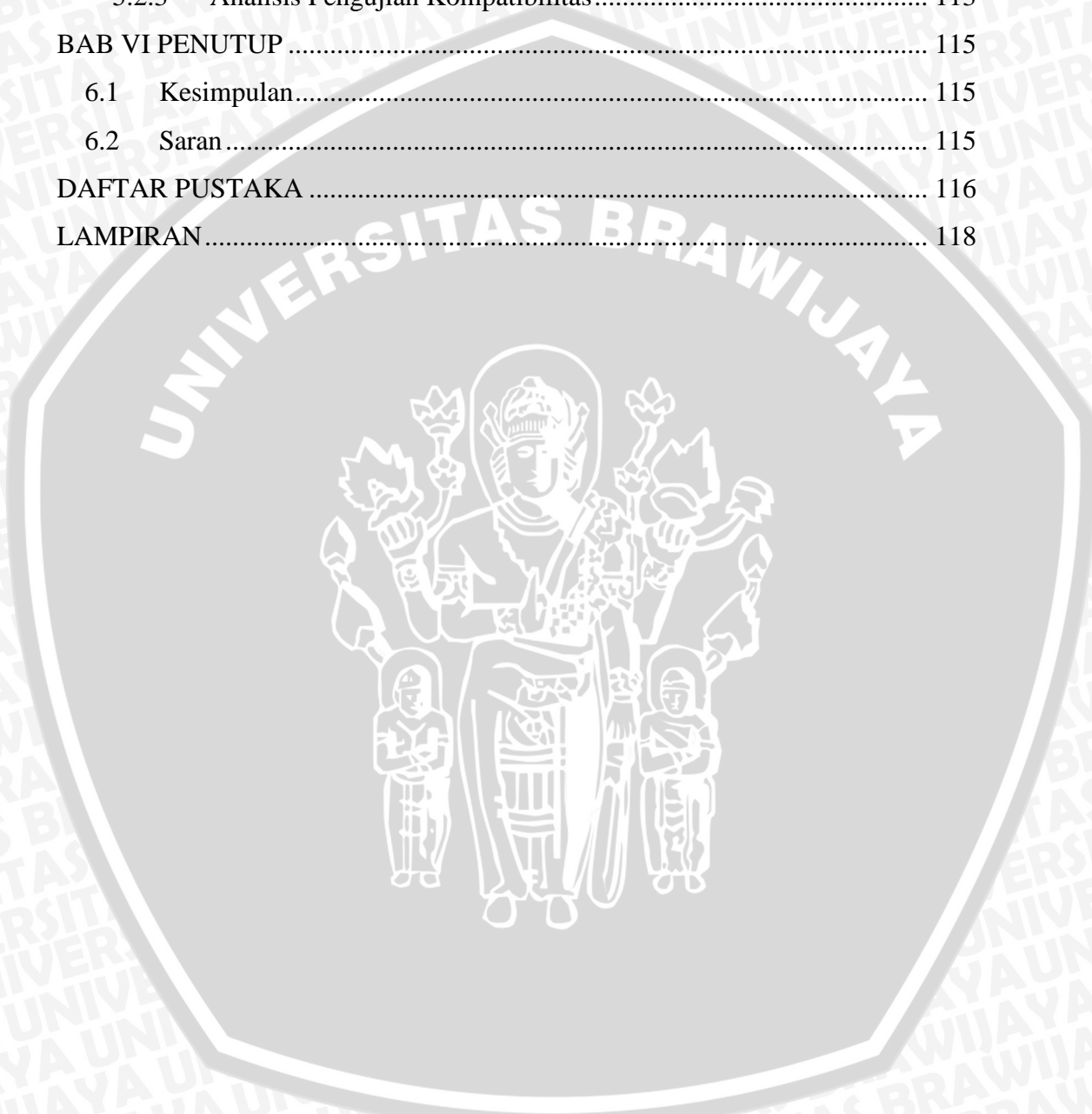
2.5.2	Penulisan Statement .....	14
2.6	jQuery .....	15
2.7	Ratchet .....	15
2.8	PhoneGap .....	16
2.9	PHP dan MySQL .....	17
2.10	<i>Location Based Service</i> .....	19
2.11	Google Map .....	20
2.12	Google Map API .....	20
2.13	Algoritma Haversine .....	21
2.14	JSON .....	23
2.15	Teknik Pengujian .....	23
2.15.1	<i>White-Box Testing</i> .....	24
2.15.2	<i>Black-Box Testing</i> .....	26
<b>BAB III METODOLOGI PENELITIAN</b> .....		27
3.1	Studi Literatur .....	28
3.2	Pengumpulan Data .....	28
3.3	Analisis Kebutuhan .....	28
3.4	Perancangan Sistem .....	29
3.5	Implementasi .....	29
3.6	Pengujian dan Analisis .....	30
3.7	Kesimpulan dan Saran .....	31
<b>BAB IV PERANCANGAN DAN IMPLEMENTASI</b> .....		32
4.1	Perancangan .....	32
4.2	Perancangan Umum Sistem .....	33
4.3	Analisis Kebutuhan .....	34
4.3.1	Gambaran Umum Aplikasi .....	34
4.3.1.1	Deskripsi Aplikasi .....	35
4.3.1.2	Lingkungan Aplikasi .....	35
4.3.2	Identifikasi Aktor .....	35
4.3.3	Analisis Kebutuhan Fungsional .....	36
4.3.4	Analisis Kebutuhan Non Fungsional .....	50
4.4	Perancangan Perangkat Lunak .....	51



4.4.1	Perancangan UML ( <i>Unified Modelling Language</i> ) .....	51
4.4.1.1	Perancangan <i>Activity Diagram</i> .....	51
4.4.2	Perancangan Arsitektur Sistem .....	63
4.4.3	Perancangan Basis Data .....	64
4.4.4	Perancangan <i>Class Diagram</i> .....	66
4.4.5	..... Perancangan Algoritma Pencarian ATM Terdekat dengan Haversine .....	67
4.4.5	Perancangan Antarmuka .....	70
4.4.4.1	Perancangan Tampilan .....	70
4.5	Implementasi .....	78
4.5.1	Spesifikasi Sistem .....	78
4.5.1.1	Spesifikasi Perangkat Keras .....	79
4.5.1.2	Spesifikasi Perangkat Lunak .....	79
4.5.2	Batasan Implementasi .....	80
4.5.3	Implementasi Format Data JSON .....	81
4.5.4	Implementasi Algoritma .....	83
4.5.4.1	Implementasi Algoritma Haversine .....	83
4.5.4.2	Implementasi Algoritma Tambah Data ATM .....	84
4.5.5	Implementasi Antarmuka Aplikasi .....	85
4.5.5.1	Implementasi <i>Back-End (Server)</i> .....	85
4.5.5.2	Implementasi <i>Front-End</i> .....	88
BAB V PENGUJIAN DAN ANALISIS .....		96
5.1	Pengujian .....	96
5.1.1	Pengujian Unit .....	97
5.1.1.1	Pengujian Unit Algoritma Menambah Data ATM .....	97
5.1.2	Pengujian Validasi .....	100
5.1.2.1	Kasus Uji Validasi Tampilan Admin .....	100
5.1.2.2	Kasus Uji Validasi Tampilan Pengguna .....	102
5.1.2.3	Hasil Pengujian Validasi .....	108
5.1.3	Pengujian Kompatibilitas .....	111
5.1.3.1	Pengujian Kompatibilitas Android Versi 2.3.5 .....	111
5.1.3.2	Pengujian Kompatibilitas Android Versi 4.1.2 .....	111



5.1.3.3	Pengujian Kompatibilitas Android Versi 4.2.2.....	112
5.2	Analisis.....	113
5.2.1	Analisis Pengujian Unit .....	113
5.2.2	Analisis Pengujian Validasi .....	113
5.2.3	Analisis Pengujian Kompatibilitas.....	113
BAB VI PENUTUP .....		115
6.1	Kesimpulan.....	115
6.2	Saran.....	115
DAFTAR PUSTAKA .....		116
LAMPIRAN.....		118



## DAFTAR GAMBAR

Gambar 2.1	Arsitektur Android .....	9
Gambar 2.2	Logo HTML 5 .....	13
Gambar 2.3	Logo CSS3 .....	13
Gambar 2.4	Logo jQuery .....	15
Gambar 2.5	Segitiga Bola yang Diselesaikan dengan Rumus <i>Haversine</i> .....	22
Gambar 2.6	Transformasi <i>Flow Chart</i> ke <i>Flow Graph</i> .....	25
Gambar 3.1	Diagram Tahapan Proses dalam Penelitian.....	27
Gambar 4. 1	Diagram Perancangan .....	33
Gambar 4. 2	Perancangan Umum sistem .....	34
Gambar 4. 3	Diagram <i>Use Case</i> .....	38
Gambar 4. 4	<i>Activity Diagram</i> untuk Lihat ATM Terdekat.....	52
Gambar 4. 5	<i>Activity Diagram</i> untuk Lihat Detail ATM.....	53
Gambar 4. 6	<i>Activity Diagram</i> untuk Daftar Lokasi.....	54
Gambar 4. 7	<i>Activity Diagram</i> untuk Lihat Peta Lokasi.....	55
Gambar 4. 8	<i>Activity Diagram</i> untuk Lihat ATM pada Peta .....	56
Gambar 4. 9	<i>Activity Diagram</i> untuk Lihat Jalur.....	57
Gambar 4. 10	<i>Activity Diagram</i> untuk Lihat Rute .....	58
Gambar 4. 11	<i>Activity Diagram</i> untuk Home .....	59
Gambar 4. 12	<i>Activity Diagram</i> Tentang Aplikasi .....	60
Gambar 4. 13	<i>Sequence Diagram</i> Lihat ATM Terdekat.....	61
Gambar 4. 14	<i>Sequence Diagram</i> Lihat Daftar Lokasi.....	62
Gambar 4. 15	<i>Sequence Diagram</i> Peta Lokasi.....	63
Gambar 4. 16	Perancangan Arsitektur Sistem Aplikasi.....	64
Gambar 4. 17	<i>Entity Relational Diagram</i> Sistem Pencarian Lokasi ATM Terdekat .....	65
Gambar 4. 18	<i>Class Diagram</i> Aplikasi Pencarian Lokasi ATM Terdekat .....	67
Gambar 4.19	<i>Flow Chart</i> Pencarian ATM Terdekat Menggunakan Haversine ...	68
Gambar 4.20	Antarmuka Halaman <i>Splash Screen</i> .....	71
Gambar 4.21	Antarmuka Halaman Utama ( <i>Home</i> ).....	72

Gambar 4.22 Antarmuka Halaman ATM Terdekat .....	73
Gambar 4.23 Antarmuka Halaman Detail ATM dari ATM Terdekat .....	73
Gambar 4.24 Antarmuka Halaman Peta Lokasi.....	74
Gambar 4.25 Antarmuka Halaman Daftar Lokasi .....	75
Gambar 4.26 Antarmuka Halaman Daftar Lokasi ATM Dari Bank yang Terpilih .....	76
Gambar 4.27 Antarmuka Halaman Detail ATM dari Daftar Lokasi .....	76
Gambar 4.28 Antarmuka Halaman Tentang Aplikasi.....	77
Gambar 4.29 Diagram Pohon Implementasi.....	78
Gambar 4.30 Algoritma Menambah Data ATM .....	85
Gambar 4.31 Halaman Login Admin.....	86
Gambar 4.32 Halaman Login Admin Bila Terjadi Kesalahan .....	86
Gambar 4.33 Halaman <i>Home</i> Admin.....	87
Gambar 4.34 Halaman Admin Menu <i>Category</i> .....	87
Gambar 4.35 Halaman Admin Menu <i>Location List</i> .....	88
Gambar 4.36 Halaman <i>Update</i> Data ATM pada Menu <i>Location List</i> .....	88
Gambar 4.37 Tampilan Halaman <i>Home</i> .....	89
Gambar 4.38 Tampilan Halaman ATM Terdekat.....	90
Gambar 4.39 Tampilan Halaman Detail ATM .....	90
Gambar 4.40 Tampilan Halaman Lihat Peta.....	91
Gambar 4.41 Tampilan Halaman Lihat Jalur.....	91
Gambar 4.42 Tampilan Halaman Lihat Rute .....	92
Gambar 4.43 Tampilan Halaman Daftar Lokasi .....	93
Gambar 4.44 Tampilan Halaman <i>Search</i> ATM .....	93
Gambar 4.45 Tampilan Halaman Peta Lokasi .....	94
Gambar 4.46 Tampilan Halaman Tentang Aplikasi .....	95
Gambar 5.1 Diagram Pohon Pengujian dan Analisis.....	96
Gambar 5.2 Pengujian Unit Algoritma Menambah Data ATM.....	98
Gambar 5.3 Pemodelan <i>Flow Graph</i> Algoritma Menambah Data ATM.....	99



DAFTAR TABEL

Tabel 2. 1 Fitur-fitur PhoneGap .....	16
Tabel 4. 1 Identifikasi Aktor .....	36
Tabel 4. 2 Spesifikasi Kebutuhan Fungsional.....	36
Tabel 4. 3 Skenario <i>Use Case Login</i> .....	39
Tabel 4. 4 Skenario <i>Use Case Input</i> Data Lokasi ATM .....	39
Tabel 4. 5 Skenario <i>Use Case Ubah</i> Data Lokasi ATM .....	40
Tabel 4. 6 Skenario <i>Use Case Hapus</i> Data Lokasi ATM.....	41
Tabel 4. 7 Skenario <i>Use Case Home</i> .....	41
Tabel 4. 8 Skenario <i>Use Case Lihat</i> ATM Terdekat.....	42
Tabel 4. 9 Skenario <i>Use Case Pilih</i> ATM.....	43
Tabel 4. 10 Skenario <i>Use Case Lihat</i> Detail ATM .....	44
Tabel 4. 11 Skenario <i>Use Case Lihat</i> Peta .....	45
Tabel 4. 12 Skenario <i>Use Case Lihat</i> Jalur .....	45
Tabel 4. 13 Skenario <i>Use Case Lihat</i> Rute .....	46
Tabel 4. 14 Skenario <i>Use Case Lihat</i> Daftar Lokasi.....	47
Tabel 4. 15 Skenario <i>Use Case Search</i> ATM .....	48
Tabel 4. 16 Skenario <i>Use Case Lihat</i> Peta Lokasi.....	49
Tabel 4. 17 Skenario <i>Use Case Tentang</i> Aplikasi .....	50
Tabel 4. 18 Analisis Kebutuhan Non Fungsional .....	51
Tabel 4. 19 Struktur Tabel ATM.....	65
Tabel 4. 20 Struktur Tabel Kategori .....	66
Tabel 4. 21 Struktur Tabel <i>User</i> .....	66
Tabel 4. 22 Koordinat Antara UB dengan ATM BCA Kiosk Blimbing.....	68
Tabel 4. 23 Koordinat Antara Batu Secret Zoo dengan ATM BCA Kiosk Blimbing.....	69
Tabel 4. 24 Spesifikasi Perangkat Keras Komputer.....	79
Tabel 4. 25 Spesifikasi Perangkat Keras <i>Smartphone</i> Android .....	79
Tabel 4. 26 Spesifikasi Perangkat Lunak Komputer.....	79
Tabel 4. 27 Spesifikasi Perangkat Lunak <i>Smartphone</i> Android .....	80

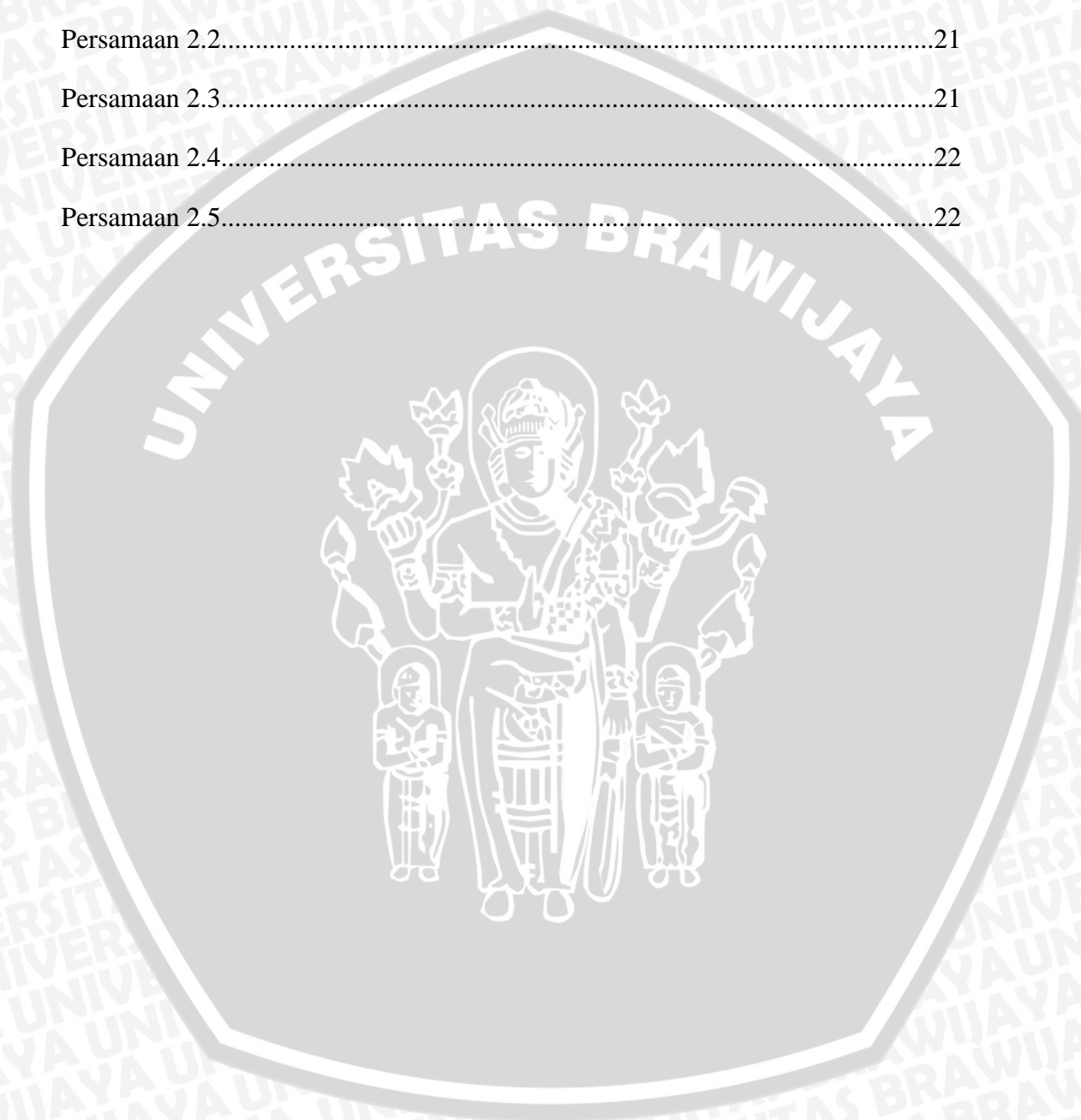


Tabel 4. 28 Kode Program Service <i>Get_List</i> .....	81
Tabel 4. 29 Kode Program Function <i>Get_List</i> .....	81
Tabel 4. 30 Struktur Data <i>Service Get_List</i> .....	82
Tabel 4. 31 Implementasi Algoritma pada Menu ATM Terdekat .....	83
Tabel 5. 1 Kasus Uji Algoritma Menambah ATM .....	100
Tabel 5. 2 Kasus Uji Validasi <i>Login</i> (data normal) .....	100
Tabel 5. 3 Kasus Uji Validasi <i>Login</i> (data salah).....	101
Tabel 5. 4 Kasus Uji Validasi <i>Input</i> Data ATM.....	101
Tabel 5. 5 Kasus Uji Validasi Ubah Data ATM .....	102
Tabel 5. 6 Kasus Uji Validasi Hapus Data ATM.....	102
Tabel 5. 7 Kasus Uji Validasi Buka Aplikasi .....	103
Tabel 5. 8 Kasus Uji Validasi Lihat Halaman <i>Home</i> .....	103
Tabel 5. 9 Kasus Uji Validasi Lihat ATM Terdekat.....	103
Tabel 5. 10 Kasus Uji Validasi Pilih ATM.....	104
Tabel 5. 11 Kasus Uji Validasi Lihat Detail ATM .....	104
Tabel 5. 12 Kasus Uji Validasi Lihat Peta .....	105
Tabel 5. 13 Kasus Uji Validasi Lihat Jalur .....	105
Tabel 5. 14 Kasus Uji Validasi Lihat Rute .....	106
Tabel 5. 15 Kasus Uji Validasi Lihat Daftar Lokasi.....	106
Tabel 5. 16 Kasus Uji Validasi <i>Search</i> ATM .....	107
Tabel 5. 17 Kasus Uji Validasi Lihat Peta Lokasi .....	107
Tabel 5. 18 Kasus Uji Validasi Lihat Tentang Aplikasi .....	108
Tabel 5. 19 Hasil Pengujian Validasi.....	108
Tabel 5. 20 Kasus Uji dan Hasil Pengujian Kompatibilitas pada Android versi 2.3.5.....	111
Tabel 5. 21 Kasus Uji dan Hasil Pengujian Kompatibilitas pada Android Versi 4.1.2.....	112
Tabel 5. 22 Kasus Uji dan Hasil Pengujian Kompatibilitas pada Android Versi 4.2.2.....	112



## DAFTAR RUMUS

Persamaan 2.1.....	21
Persamaan 2.2.....	21
Persamaan 2.3.....	21
Persamaan 2.4.....	22
Persamaan 2.5.....	22



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Semakin pesatnya perkembangan teknologi informasi telah mendorong masyarakat untuk berlomba-lomba menerapkan sistem informasi dalam berbagai aspek kehidupan. Saat ini kebutuhan untuk memperoleh informasi telah menjadi kebutuhan pokok masyarakat. Salah satu informasi yang dibutuhkan masyarakat adalah kebutuhan informasi yang berkaitan dengan tempat-tempat fasilitas umum. ATM (*Automated Teller Machine*) atau Anjungan Tunai Mandiri) merupakan salah satu fasilitas umum dalam bentuk perangkat komputerisasi dari suatu lembaga keuangan (bank) dalam upaya menyediakan layanan transaksi keuangan [SUR-12], kini telah tersebar di berbagai sudut daerah di wilayah Negara Indonesia dalam jumlah yang relatif banyak, tak terkecuali di kota Malang.

Malang adalah salah satu kota di Propinsi Jawa Timur yang terbesar setelah kota Surabaya, baik dalam segi wilayah maupun populasi. Malang memiliki julukan sebagai kota bunga, kota pendidikan, kota industri, dan kota pariwisata. Selain itu, dari situs resmi kota Malang, dikemukakan bahwa Malang memiliki situasi kota yang tenang, penduduknya yang ramah, harga makanan yang relatif murah, serta fasilitas pariwisata dan pendidikan yang sangat memadai. Keistimewaan inilah yang menjadikan kota Malang ramai dikunjungi penduduk dari dalam negeri maupun luar negeri, sehingga populasi masyarakat di kota Malang semakin berkembang.

Berkembangnya populasi masyarakat di kota Malang berbanding lurus dengan tersebarnya fasilitas ATM di berbagai sudut daerah di Kota Malang, mulai dari ATM Mandiri, BCA, BNI, BRI, BTN, Danamon, serta ATM dari bank lainnya. Namun karena banyaknya jumlah fasilitas ATM yang ada di kota Malang serta lokasinya yang tersebar di segala penjuru daerah kota Malang, membuat sebagian masyarakat, mungkin dalam keadaan terdesak, mengalami kesulitan dalam menemukan posisi atau letak ATM terdekat. Hal ini mungkin dikarenakan sebagian masyarakat belum mengerti area jalanan kota Malang ataupun merupakan masyarakat pendatang.

Perkembangan teknologi informasi pada perangkat bergerak yang saat ini berkembang pesat menjadi salah satu sarana bagi masyarakat luas untuk saling bertukar informasi. Perangkat bergerak sudah menjadi alat interaksi umum yang dilakukan oleh masyarakat untuk memenuhi kebutuhan sehari-hari. Pada masa kini mobilitas seseorang yang cepat berdampak pula pada mobilitas aplikasi serta perangkat pendukung lainnya, sehingga menyebabkan aplikasi bergerak terus meningkat termasuk sistem operasi yang digunakan, salah satunya adalah Android [SUR-12].

Android saat ini sangat populer di kalangan masyarakat dan mendukung fitur *touch screen* beserta fitur canggih lainnya. Android juga mendukung pengembangan aplikasi secara terbuka, sehingga banyak *programmer* tertarik untuk terlibat dalam pengembangan aplikasi-aplikasi yang dapat digunakan pada perangkat *mobile* dengan sistem Android. Munculnya ponsel Android yang semakin memanjakan penggunanya dalam mengikuti perkembangan teknologi aplikasi *mobile*, telah mendorong penulis untuk merancang suatu sistem pengolahan data dan informasi yang meliputi pencarian lokasi ATM terdekat di kota Malang. Pada aplikasi digunakan algoritma Haversine sebagai estimasi dalam perhitungan jarak antara dua titik GPS, sehingga didapatkan daftar ATM terdekat dari posisi pengguna. Penerapan teknologi yang tepat guna dan tepat sasaran diharapkan dapat memudahkan masyarakat pengguna Android di kota Malang dalam mencari lokasi ATM yang paling terdekat dari lokasi pengguna saat itu.

## 1.2 Rumusan Masalah

Permasalahan yang akan dibahas pada tugas akhir ini adalah bagaimana merancang, membangun, dan menguji sistem aplikasi Android pencarian lokasi ATM terdekat di kota Malang.

## 1.3 Batasan Masalah

Batasan masalah yang perlu diajukan dalam pengembangan aplikasi perangkat lunak skripsi ini antara lain :

1. Pada aplikasi ini digunakan data lokasi ATM dari 10 jenis bank yang ada di kota Malang, yaitu :
  - a. Bank Danamon
  - b. Bank Mandiri
  - c. Bank Bukopin
  - d. Bank Panin
  - e. Bank Rakyat Indonesia
  - f. BTN
  - g. Bank Niaga
  - h. BNI
  - i. BII
  - j. Bank Permata
2. Algoritma Haversine digunakan hanya sebagai estimasi untuk menghitung jarak antar dua posisi dalam garis lurus, dan bukan jarak sebenarnya berdasarkan rute.
3. Aplikasi sisi *server/admin* dibangun dengan menggunakan bahasa pemrograman HTML5, CSS3, dan JavaScript.
4. Aplikasi sisi klien/*user* dibangun dengan pendekatan web *hybrid* menggunakan bahasa pemrograman HTML5, CSS3, JavaScript, dan digunakan *framework* PhoneGap untuk mengubah menjadi aplikasi Android.

#### 1.4 Tujuan

Tujuan dari penelitian tugas akhir ini adalah menghasilkan sebuah aplikasi Android yang mempunyai kapabilitas menyimpan data lokasi ATM serta dapat menampilkan data dengan atraktif.

#### 1.5 Manfaat

Manfaat yang dapat diperoleh dari penelitian tugas akhir ini antara lain :

1. memudahkan pengguna Android, khususnya yang ada di kota Malang, untuk mencari dan menemukan lokasi ATM yang terdekat dari posisi pengguna.

2. Aplikasi ini berbasis *mobile* sehingga bisa diakses dimanapun dan kapanpun

### 1.6 Sistematika Penulisan

Sistematika penulisan dalam tugas akhir ini adalah sebagai berikut :

#### **BAB I PENDAHULUAN**

Memuat latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan.

#### **BAB II TINJAUAN PUSTAKA**

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari penelitian tugas akhir ini.

#### **BAB III METODOLOGI PENELITIAN**

Menjelaskan bagaimana metodologi untuk perancangan dan pembuatan aplikasi Android *mobile* pencarian lokasi ATM terdekat di kota Malang.

#### **BAB IV PERANCANGAN DAN IMPLEMENTASI**

Membahas analisis sistem, perancangan sistem, dan implementasi dari sistem aplikasi. Selain itu juga membahas metode pengujian yang sesuai dengan teori yang ada.

#### **BAB V PENGUJIAN DAN ANALISIS**

Menjelaskan hasil pengujian aplikasi berdasarkan parameter-parameter yang telah ditetapkan dan kemudian dilakukan analisa terhadap hasil pengujian.

#### **BAB VI PENUTUP**

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian program, serta saran-saran untuk pengembangan lebih lanjut.

## BAB II

### TINJAUAN PUSTAKA

Pada bab ini akan diuraikan mengenai teori-teori dasar pembuatan aplikasi sebagai acuan dalam perancangan sistem yang akan dijelaskan pada bab 3 yaitu metodologi penelitian. Teori-teori dasar yang akan diuraikan antara lain penjelasan tentang ATM (*Auto Teller Machine*), Android, LBS, HTML, CSS, JavaScript, jQuery, PhoneGap, Google MAP API, JSON serta teori dasar lainnya mengenai konsep pengembangan aplikasi perangkat bergerak khususnya mengenai aspek-aspek yang akan dikembangkan dalam sistem *mobile* ini.

#### 2.1 ATM (*Auto Teller Machine*)

ATM (*Automated Teller Machine*) merupakan sebuah perangkat komputerisasi yang digunakan oleh suatu lembaga keuangan (bank) dalam upaya menyediakan layanan transaksi keuangan (pengambilan uang) di tempat umum tanpa membutuhkan adanya pegawai bank (*teller*) [SUR-12].

Pada mulanya penyediaan ATM adalah untuk memudahkan layanan pengambilan uang dari tabungan nasabah, akan tetapi seiring dengan perkembangan teknologi dan kebutuhan akan peningkatan layanan kepada para nasabah, penggunaan ATM telah meluas tidak hanya sebatas pengambilan uang saja. Saat ini sudah memungkinkan bagi para nasabah untuk melakukan transfer uang, pembayaran, pengecekan saldo, dan transaksi keuangan lainnya cukup dengan menggunakan ATM. Secara umum, teknologi pada ATM merupakan suatu bentuk jaringan komputer yang tersebar. Teknik pengamanan yang diterapkan pada ATM adalah dengan penggunaan *personal identification number* (PIN) sehingga hanya orang tertentu saja yang dapat mengakses ataupun melakukan transaksi pada ATM.

##### 2.1.1 Jenis Mesin ATM

Adapun jenis mesin ATM adalah sebagai berikut [SUR-12] :

1. ATM Tunai

Adalah ATM yang dapat melakukan transaksi pengambilan tunai dan transaksi lain seperti *inquiry* saldo rekening, transfer, ganti PIN, pembelian dan pembayaran tagihan.

2. ATM Non Tunai

Adalah ATM yang hanya dapat melakukan transaksi seluruh transaksi non tunai seperti *inquiry* saldo rekening, transfer, ganti PIN, pembelian dan pembayaran tagihan dan tidak dapat melakukan transaksi pengambilan tunai. Secara fisik bentuk ATM Non Tunai berbeda dengan ATM biasa, karena ATM Non Tunai tidak memiliki bentuk fisik mesin seperti mesin ATM pada umumnya dan hanya berupa layar monitor dan papan ketik (*keyboard*) pada bagian depan layar.

3. ATM Setoran Tunai (*CDM – Cash Deposit Machine*)

Adalah Mesin ATM yang dapat menerima setoran tunai dan *inquiry* saldo rekening. ATM Setoran Tunai tidak dapat melakukan transaksi pengambilan tunai, transfer, ganti PIN, pembelian dan pembayaran tagihan. Secara fisik bentuk ATM Setoran Tunai hampir sama dengan ATM biasa, namun pada sisi depan terdapat lubang atau laci yang digunakan untuk meletakkan uang yang akan disetor.

4. *Drive Thru* Mesin

Adalah ATM yang dapat melakukan seluruh transaksi seperti halnya ATM biasa meliputi penarikan tunai, *inquiry* saldo, transfer, ganti PIN, pembelian dan pembayaran tagihan. ATM *Drive Thru* memiliki keunikan dalam lokasi dan bentuk karena didesain khusus untuk dapat melayani nasabah tanpa harus turun dari mobil.

### 2.1.2 Manfaat ATM

Ada banyak manfaat ATM bagi nasabah, perkembangan teknologi juga memberikan kesempatan bagi teknologi perbankan untuk menambah fitur-fitur baru bagi produk ATMnya. Adapun manfaat ATM adalah :

1. Dapat menarik uang tunai dalam 24 jam, sehingga nasabah tidak lagi tergantung jam pelayanan bank atau hari libur untuk mengambil uang.

2. ATM dapat menghemat waktu dan biaya tanpa harus datang langsung ke bank yang bersangkutan.
3. Lebih mudah dicapai karena mesin ATM tersedia di berbagai lokasi yang strategis.
4. ATM digunakan dengan sistem LAN dapat menggunakan adapter untuk menyediakan kecepatan transmisi.
5. ATM mampu menangani semua jenis trafik komunikasi (*voice*, *data*, *image*, *video*, suara dengan kecepatan tinggi, multimedia, dan sebagainya) dalam satu saluran dan dengan kecepatan tinggi.

## 2.2 Sistem Operasi Android

### 2.2.1 Pengertian Android Platform

Android merupakan OS *Mobile* yang tumbuh di tengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti *Windows Mobile*, *i-Phone OS*, *Symbian*, dan masih banyak lagi yang juga menawarkan kekayaan isi dan keoptimalan berjalan diatas perangkat *hardware* yang ada. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk *platform* mereka.

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android menjadi pesaing utama dari Apple pada sistem operasi tablet PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan di atas adalah karena Android itu sendiri adalah *platform* yang sangat lengkap baik itu sistem operasinya, aplikasi dan *tool* pengembangan, *market* aplikasi Android serta dukungan yang sangat tinggi dari komunitas *open source* di dunia, sehingga Android terus berkembang pesat dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia. [HER-11]

Selain itu Android juga dipuji sebagai *platform mobile* pertama yang lengkap, terbuka, dan bebas [MUL-10].



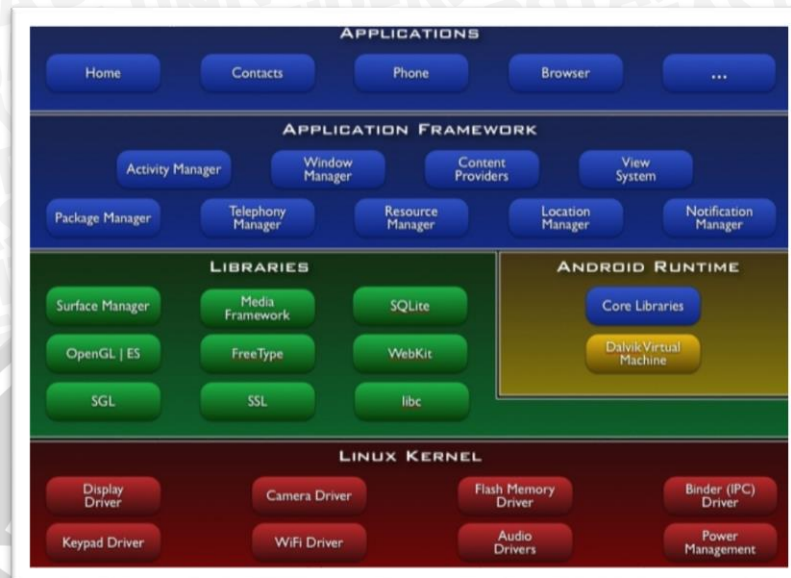
1. Lengkap (*Complete Platform*) : Para *designer* dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan platform Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.
2. Terbuka (*Open Source Platform*) : Platform Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri mengembangkan Linux Kernel 2.6
3. *Free* (*Free Platform*) : Android adalah *platform* atau aplikasi yang bebas untuk *develop*. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apa pun.

### 2.2.2 Fitur dan Arsitektur Android

Fitur yang tersedia pada Android adalah [NAZ-11] :

1. *Framework* aplikasi : memungkinkan penggunaan dan pemindahan dari komponen yang tersedia.
2. *Davilk Virtual Machine* : *Virtual machine* yang dioptimalkan untuk perangkat *mobile*. DVM menggunakan kernel linux untuk menangani fungsionalitas tingkat rendah, termasuk keamanan, *threading*, dan proses serta manajemen memori [MUL-10].
3. Grafik : grafik 2D dan grafik 3D yang didasarkan pada *library* OpenGL.
4. SQLite : untuk penyimpanan data.
5. Media pendukung : audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
6. GSM, Bluetooth, EDGE, 3G, dan WiFi (tergantung *hardware*).
7. Kamera, *Global Positioning System* (GPS), compass, dan accelerometer (tergantung *hardware*).
8. Lingkungan pengembangan yang kaya, termasuk emulator, peralatan *debugging*, dan *plugin* untuk Eclipse IDE.

Sistem operasi Android dibangun berdasarkan kernel Linux dan memiliki arsitektur sesuai Gambar 2.1.



**Gambar 2.1** Arsitektur Android

Sumber : [AZZ-12]

**a. Applications**

Puncak dari diagram arsitektur Android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi di balik lapisan aplikasi. Lapisan ini berjalan dalam *Android runtime* dengan menggunakan kelas dan *service* yang tersedia pada *framework* aplikasi. Lapisan aplikasi Android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada Android semua aplikasi, baik aplikasi inti maupun aplikasi pihak ketiga berjalan di atas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama. [AZZ-12]

**b. Applications Framework**

Pada lapisan ini pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi ini dirancang supaya kita dengan mudah dapat menggunakan kembali

komponen yang sudah digunakan (*reuse*). Komponen-komponen yang termasuk di dalam *Applications Framework* adalah sebagai berikut [AZZ-12] :

1. *Content Provider*

Berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.

2. *Resource Manager*

Untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya di luar kode program, seperti karakter, grafik, dan *file layout*.

3. *Notification Manager*

Mencakup berbagai macam peringatan seperti pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada status bar.

4. *Location Manager*

Berfungsi untuk memberikan informasi detail mengenai lokasi perangkat Android berada.

5. *Activity Manager*

Berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan "*Backstack*" untuk navigasi penggunaan.

c. *Libraries*

*Libraries* ini adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, *layer* ini meliputi berbagai *library* C/C++ inti seperti [AZZ-12] :

1. *Libraries* media untuk pemutaran media audio dan video.

2. *Libraries* untuk menjalankan tampilan.

3. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.

4. *Libraries SQLite* untuk dukungan *database* yang digunakan pada aplikasi.

5. *Libraries SSL* dan WebKit terintegrasi dengan *web browser* dan *security*.

6. *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embedded web view*.

7. *Libraries 3D* yang mencakup implementasi OpenGL ES 1.0 API's.

**d. *Android Runtime***

*Layer* yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi linux. DVM (*Dalvik Virtual Machine*) merupakan mesin yang berbentuk dasar kerangka aplikasi Android. Di dalam *Android Run Time* dibagi menjadi dua bagian yaitu [AZZ-12]:

1. *Core Libraries*: Aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh *core libraries*.
2. *Dalvik Virtual Machine* (DVM) : Virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

**e. *Linux Kernel***

Linux kernel adalah *layer* dimana inti dari sistem operasi dari Android itu berada. Berisi *file-file* sistem yang mengatur *system processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi Android lainnya. Linux kernel yang digunakan Android adalah linux kernel *release 2.6*.

**2.3 HTML 5**

Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML (*HyperText Markup Language*) adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML sat ini merupakan standard internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989. [W3C-14]

HTML adalah bahasa dari *World Wide Web* (www) yang dipergunakan untuk menyusun dan membentuk dokumen agar dapat ditampilkan pada program web *browser*. HTML juga dapat disebut sebagai protokol yang digunakan untuk

mentransfer data atau dokumen dari web *server* ke *browser*. HTML inilah yang menjadi dasar bila akan menjelajah internet dan melihat halaman web yang menarik.

Selama tahun 90-an telah muncul berbagai versi HTML dengan berbagai versi HTML dengan berbagai macam fiturnya. Versi HTML yang ada tersebut antara lain:

1. Versi 1.0

Merupakan versi pertama setelah lahirnya HTML dan mempunyai kemampuan seperti untuk *heading*, *paragraph*, *hypertext*, *list*, cetak tebal (*bold*), dan cetak miring (*italic*) pada teks. Selain itu juga dukungan peletakan *image* pada dokumen dengan tidak memperbolehkan peletakan teks di sekelilingnya (*wrapping*).

2. Versi 2.0

Dirilis pada tanggal 14 Januari 1996 dengan beberapa kemampuan tambahan seperti penambahan *form*. Hal ini menjadi pionir untuk membuat sebuah web yang interaktif.

3. Versi 3.0

Dirilis pada tanggal 18 Desember 1997, yang dikenal dengan HTML+. Versi ini memiliki beberapa fitur tambahan seperti fitur *table* dalam paragraf.

4. Versi 3.2

Dirilis pada bulan Mei 1998, merupakan versi terbaru penyempurnaan dari versi 3.0.

5. Versi 4.0

Dirilis pada tanggal 24 Desember 1999 dengan penambahan beberapa fitur seperti adanya *link*, *imagemaps*, *image*, dan lain-lain.

6. Versi 5.0

Dirilis pada tahun 2009 yang menjadi standar baru untuk HTML, XHTML, dan DOM HTML. HTML5 ini merupakan proyek kerja sama antara W3C (*World Wide Web Consortium*) dan WHATG (*Web Hypertext Application Technology Working Group*). [W3C-14]



Gambar 2.2 Logo HTML 5

Sumber : [W3C-14]

#### 2.4 CSS3

World Wide Web Consortium menjadikan CSS (*Cascading Style-Sheet*) sebagai bahasa pemrograman standart dalam pengaturan tampilan web, mengatur *layout* (letak web), serta format huruf. CSS merupakan *file* yang ditambahkan ke dalam *website* untuk mengatur *style website* agar terlihat seragam. Hampir semua *programmer website* sekarang menggunakan *file* CSS karena sangat mudah ketika harus mengubah keseluruhan atau sebagian tampilan pada *website*. Seorang *programmer* tidak perlu membuka satu per satu dan mengubah *layout* yang diinginkan melainkan hanya dengan membuat satu *file* CSS saja maka perubahan dapat diterapkan dalam semua halaman *website*. [W3C-14]

CSS3 adalah standar untuk CSS yang paling baru, banyak batasan *website* desain yang dahulu tidak bisa ditangani oleh CSS sekarang bisa dilakukan dengan CSS3. CSS3 sepenuhnya kompatibel dan pelengkap dari CSS lama, sehingga *browser* akan selalu mendukung CSS lama dan kita tidak harus mengubah desain yang sudah ada.



Gambar 2.3 Logo CSS3

Sumber : [W3C-14]

## 2.5 JavaScript

JavaScript adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar *browser* populer [SUN-07]. JavaScript dapat mempermudah fungsionalitas pada pembuatan *website*, serta dapat membuat *website* lebih interaktif dan animatif.

### 2.5.1 Tag JavaScript

Penulisan perintah JavaScript dapat dituliskan bersama dengan file HTML secara langsung atau dapat dituliskan pada *file* tersendiri dengan ekstensi *.js*. Untuk penulisan *tag* JavaScript sebagai deklarasi awal untuk memulai penulisan jika JavaScript dituliskan bersama dengan file HTML adalah sebagai berikut :

```
<script>
    //kode JavaScript
</script>
```

Atau

```
<script type="text/JavaScript">
    //kode JavaScript
</script>
```

Apabila perintah JavaScript dituliskan pada dokumen khusus yaitu *.js* maka untuk menggunakannya harus dilakukan pemanggilan dokumen JavaScript tersebut dengan perintah sebagai berikut :

```
<script type="text/JavaScript"
src="nama_file.js"></script>
```

### 2.5.2 Penulisan Statement

*Statement* JavaScript adalah sebuah perintah yang ditujukan *browser*. *Statement* akan memberitahukan pada *browser* apa yang harus dilakukan. *Statement* berupa satu baris perintah, contoh penulisan *statement* pada JavaScript misalnya

```
document.write("<h1>Hello JavaScript</h1>");
```

Keterangan :

*Statement* di atas berarti memerintahkan browser untuk menuliskan “Hello JavaScript” dengan *tag* h1 pada halaman web.

Perintah pada JavaScript bersifat *case sensitive* yang membedakan huruf besar dan huruf kecil, misalnya **write** dengan **Write** merupakan perintah yang berbeda. Hal ini berbeda dengan perintah HTML yang bersifat *case insensitive*, misalnya <h1> dengan <H1> adalah perintah yang sama.

Kode JavaScript merupakan rangkaian dari statemen-statementen JavaScript. Penulisan kode JavaScript dapat dituliskan dalam sebuah blok maupun tanpa sebuah blok. Tujuan dari penulisan JavaScript dalam sebuah blok adalah untuk mengklarifikasikan fungsionalitas kode JavaScript yang berbeda agar dapat dijalankan secara bersamaan [SUN-07].

## 2.6 jQuery

jQuery adalah sebuah *library* JavaScript yang sangat ringkas dan sederhana untuk memanipulasi komponen di dokumen HTML, menangani *event*, animasi, efek dan memproses interaksi ajax. jQuery dirancang sedemikian rupa supaya membuat program menggunakan JavaScript menjadi relatif sangat mudah. Sesuai slogan nya, *write less, do more*. Menulis kode lebih sedikit, tetapi melakukan pekerjaan lebih banyak [WAH-13].



Gambar 2.4 Logo jQuery

Sumber: [JQU-14]

## 2.7 Ratchet

Ratchet merupakan sebuah *front-end framework* yang memiliki koleksi *user interface* dan *plugin* JavaScript yang digunakan untuk membangun aplikasi *mobile* sederhana dengan HTML, CSS, dan komponen JavaScript. *Framework* ini juga menggunakan *plugin* yang memiliki syntax sama seperti jQuery. [PET-14]



## 2.8 PhoneGap

PhoneGap adalah *framework open source* untuk membuat aplikasi pada berbagai macam *platform* seperti Android, Blackberry, Iphone atau Windows Phone menggunakan HTML5, JQuery, JQuery Mobile, dan CSS3 [EDD-14]. Di mana inputan atau bahan baku dari aplikasi yang akan dibuat dengan *library* PhoneGap adalah sebuah *file web* [EMS-13].

PhoneGap dikontribusikan untuk *Apache Software Foundation* (ASF) di bawah nama Apache Cordova. PhoneGap memungkinkan *developer* untuk membuat aplikasi *native* untuk perangkat *mobile*, dengan *script* kode yang bersifat *open source*. *User Interface* untuk aplikasi PhoneGap dibuat menggunakan HTML, CSS dan JavaScripts. *UI layer* dari sebuah aplikasi PhoneGap adalah sebuah tampilan web *browser* yang memenuhi 100% tinggi layar dan 100% lebar layar. [EMS-13]

Ada beberapa fitur yang dapat diberikan oleh PhoneGap diantaranya adalah *Accelerometer*, *Camera*, *Compass*, *Contacts*, *File*, dan lain-lain.

**Tabel 2.1** Fitur-fitur PhoneGap

	iPhone/ iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 6.0+	WebOS	Windows Phone 7+ 8	Symbian	Bada
Accelerometer	√	√	√	√	√	√	√	√
Camera	√	√	√	√	√	√	√	√
Compass		√	√	X	√	√	X	√
Contact	√	√	√	√	X	√	√	√
File	√	√	√	√	X	√	X	X
Geolocation	√	√	√	√	√	√	√	√
Media	√	√	√	X	X	√	X	X
Network	√	√	√	√	√	√	√	√
Notification (Alert)	√	√	√	√	√	√	√	√
Notification (Sound)	√	√	√	√	√	√	√	√
Notification (Vibration)	√	√	√	√	√	√	√	√
Storage	√	√	√	√	√	√	X	X

Keterangan:

√ - supported feature

X - unsupported feature due to hardware or software restrictions

Sumber : [EMS-13]

## 2.9 PHP dan MySQL

PHP (*Hypertext Preprocessor*) adalah bahasa komputer yang dibuat untuk pengembangan web dinamis. Pada umumnya PHP digunakan di *server* namun juga dapat berdiri sendiri sebagai aplikasi *graphical* [PHP-08].

Pada tugas akhir ini digunakan PHP sebagai bahasa pemrograman dan MySQL sebagai database sisi server untuk menyimpan informasi mengenai data lokasi ATM. Penggunaan PHP dan MySQL dipilih karena PHP dan MySQL memiliki beberapa kelebihan seperti dinyatakan oleh Nugroho, B (2008) [NUG-08] kelebihannya antara lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan dimana-mana dari mulai IIS sampai dengan Apache, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

Sedangkan *database* MySQL memiliki beberapa kelebihan yaitu:

1. *Portability*  
MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. *Open Source*  
MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. *Multiuser*  
MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

6. *Command dan functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.

7. *Security*

MySQL memiliki beberapa lapisan sekuritas seperti *level subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

8. *Scalability dan limits*

MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, *Unix socket (UNIX)*, atau *Named Pipes (NT)*.

10. *Localization*

MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. *Interface*

MySQL memiliki *interface* (antarmuka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Clients dan tools*

MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

### 13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan *database* lainnya semacam PostgreSQL ataupun Oracle.

## 2.10 Location Based Service

*Location Based Service* (LBS) atau Layanan Berbasis Lokasi merupakan layanan informasi yang dapat diakses melalui perangkat *mobile* melalui jaringan seluler dan memiliki kemampuan untuk memanfaatkan lokasi posisi perangkat *mobile* [VIR-01]. Layanan berbasis lokasi ini menggunakan teknologi GPS (*Global Positioning System*), teknologi ini memungkinkan para pengguna dapat memperoleh informasi lokasi sesuai dengan kebutuhannya. Dua unsur utama LBS adalah [ROM-12]:

### a. Location Manager (API Maps)

Menyediakan tools atau resource untuk LBS, *Application Programming Interface* (API) *Maps* menyediakan fasilitas untuk menampilkan, memanipulasi peta beserta fitur-fitur lainnya, seperti tampilan satelit, jalan, maupun gabungannya. Paket ini berada pada [com.google.android.maps](http://com.google.android.maps)

### b. Location Providers (API Location)

Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. API *Location* berhubungan dengan data GPS dan data lokasi *real-time*. API *Location* berada pada paket Android yaitu dalam paket `android.location`. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini, *track* gerakan atau perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

LBS dapat diklasifikasikan menjadi tiga jenis yaitu *local information*, *traffic and tracking information*, dan *general services*. *Local information* memungkinkan pengguna untuk mencari layanan di sekitar mereka. *Traffic and tracking information* berfokus kepada pelacakan aset atau manusia. Sedangkan *general services* tidak menyediakan informasi ke pengguna, namun menggunakan data lokasi pengguna, contohnya layanan *emergency*.

### 2.11 Google Map

Google Map adalah layanan gratis peta dan pemetaan *digital* yang bisa dimanfaatkan untuk mengamati peta dunia melalui *browser* web. Google Map memanfaatkan teknologi digital *imaging*, seperti foto satelit, sehingga kita dapat melihat bagaimana *landscape* planet bumi apabila dilihat dari luar angkasa. Layanan ini gratis dan dapat ditemukan di <http://maps.google.com>. [MAP-14]

### 2.12 Google Map API

API atau *Application Programming Interface* merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Dengan adanya API ini, maka memudahkan *programmer* untuk membongkar suatu *software* untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan *programmer* menggunakan *system function*. Proses ini dikelola melalui *operating system*. Keunggulan dari API ini adalah memungkinkan suatu aplikasi dengan aplikasi lainnya dapat saling berhubungan dan berinteraksi. Bahasa pemrograman yang digunakan oleh Google Maps yang terdiri dari HTML, JavaScript, dan AJAX serta XML, memungkinkan untuk menampilkan peta Google Maps di aplikasi lain. [SVE-14]

Pada Google Maps API terdapat empat jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah [MAP-14] :

1. ROADMAP, untuk menampilkan peta biasa 2 dimensi.
2. SATELLITE, untuk menampilkan foto satelit .
3. TERRAIN, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai.
4. HYBRID, akan menunjukkan foto satelit yang di atasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota).

### 2.13 Algoritma Haversine

*Haversine* formula adalah persamaan penting dalam navigasi untuk menghitung jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur (longitude) dan lintang (latitude) [GIN-10].

Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi [ESS-11]

Untuk menghitung jarak antara dua GeoPoints rumus Haversine adalah sebagai berikut :

$$a = (\sin(lat_2 - lat_1/2)) \cdot (\sin(lat_2 - lat_1/2)) + \cos(lat_1) \cdot \cos(lat_2) \cdot (\sin(long_2 - long_1/2)) \cdot (\sin(long_2 - long_1/2)) \quad (2.1)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (2.2)$$

$$d = R \cdot c \quad (2.3)$$

Dimana,

R = Radius bumi (6,371 km)

d = Jarak (km)

$lat_1$  = Latitude objek 1

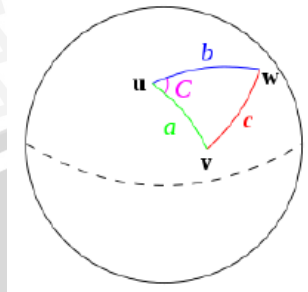
$lat_2$  = Latitude objek 2

$long_1$  = Longitude objek 1

$long_2$  = Longitude objek 2

Ini adalah kasus khusus dari sebuah formula yang lebih umum dalam trigonometri lingkaran bola, Haversine formula menghubungkan sisi dan sudut dari sebuah segitiga bola. Dalam sebuah unit bola, sebah segitiga pada permukaan bola didefinisikan oleh lingkaran besar yang menghubungkan tiga titik yaitu u, v, dan w pada bola. Jika panjang dari tiga sisi adalah **a** (dari u ke v), **b** (dari u ke w), dan **c** (dari v ke w), dan sudut yang berlawanan dari c adalah C. Karena ini adalah sebuah unit lingkaran bola, sehingga panjang **a**, **b**, dan **c** hanya sama dengan sudut (dalam radian) berdasarkan pada sisi-sisi dari pusat lingkaran bola (pada lingkaran

tak penuh, masing-masing panjang busur sama dengan sudut pusat dikalikan dengan jari-jari bola) [WIK-14].



**Gambar 2.5** Segitiga Bola yang Diselesaikan dengan Rumus *Haversine*

**Sumber :** [WIK-14]

Untuk mendapatkan rumus *haversine*, secara sederhana kita mempertimbangkan sebuah kasus khusus dimana *u* adalah kutub utara, sementara *v* dan *w* adalah dua titik yang dipisahkan oleh *d* yang akan ditentukan. Dalam hal ini *c* adalah *d/R* yang diinginkan [WIK-14].

Rumusan perhitungan jarak antara dua lokasi yang berbeda ini bisa diterapkan juga pada sintak SQL, misalkan pada aplikasi pencarian lokasi ATM ini, bisa digabungkan dengan *get GPS current location* terhadap lokasi targetnya. Dimana lokasi target diambil dari *database* yang ada pada sistem. Cara menghitungnya menggunakan fungsi pada Persamaan 2.4 dan Persamaan 2.5. Dalam sintak ini koordinat *latitude* dan *longitude* harus dikonversi dari derajat ke radian dengan mengalikan  $\pi/180$ . [ESS-11]

- *Distance* dalam satuan km :

$$Distance = (((\text{acos}(\sin(".$lat." * \pi/180)) * \sin((\text{latitude} * \pi/180)) + \cos(".$lat." * \pi/180)) * \cos((\text{latitude} * \pi/180)) * \cos((".$long." - \text{longitude}) * \pi/180)))) * 180 / \pi) * 60 * 1.1515 * 1.609344) \dots \dots \dots (2.4)$$

- *Distance* dalam satuan mil :

$$Distance = (((\text{acos}(\sin(".$lat." * \pi/180)) * \sin((\text{latitude} * \pi/180)) + \cos(".$lat." * \pi/180)) * \cos((\text{latitude} * \pi/180)) * \cos((".$long." - \text{longitude}) * \pi/180)))) * 180 / \pi) * 60 * 1.1515) \dots \dots \dots (2.5)$$

Keterangan :

Distance = jarak terdekat antara pengguna dengan lokasi ATM yang dituju

\$lat = Latitude dari lokasi ATM yang diambil dari *database* sistem

\$long = Longitude dari lokasi ATM yang diambil dari *database* sistem

Latitude = Latitude dari lokasi pengguna

Longitude = Longitude dari lokasi pengguna

Pi = nilai jari-jari ( $\pi$ ) = 3,14

## 2.14 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari JavaScript syandar ECMA-262 Edisi ke-3 pada bulan Desember 1999. JSON merupakan format teks yang yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, dan lain-lain. Oleh karena itu menjadikan JSON ideal sebagai bahasa pertukaran data. JSON terdiri dari dua struktur : [JSO-14]

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*structure*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai *array*, *vector* dan *list*.

## 2.15 Teknik Pengujian

Pengujian perangkat lunak memerlukan perancangan kasus uji (*test case*) agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan kasus uji telah berevolusi. Metode-metode ini menyediakan *developer* pendekatan sistematis untuk pengujian. Terlebih lagi metode-metode ini menyediakan mekanisme yang dapat membantu memastikan kelengkapan dari pengujian dan menyediakan kemungkinan tertinggi untuk

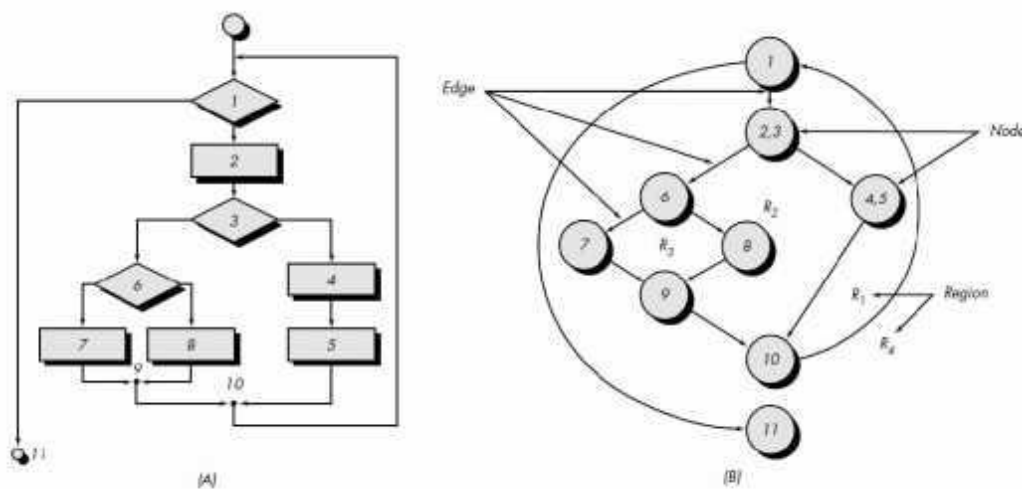


menemukan kesalahan-kesalahan dalam perangkat lunak [PRE-01]. Teknik atau metode perancangan kasus uji yang digunakan adalah *Black-Box testing* dan *white-box testing*

### 2.15.1 White-Box Testing

*White-box testing* atau *glass-box testing* merupakan sebuah metode perancangan kasus uji yang menggunakan struktur kontrol dari perancangan prosedural untuk memperoleh kasus uji [PRE-01]. Ada dua jenis pengujian yang termasuk *white-box testing* yaitu *basis path testing* dan *control structure testing*.

Pada skripsi ini menggunakan *basis path testing* yang diusulkan pertama kali oleh Tom McCabe [PRE-01]. *Basis path testing* ini memungkinkan perancang kasus uji memperoleh ukuran kompleksitas logis dari sebuah perancangan prosedural dan menggunakan pengukuran ini sebagai pedoman untuk mendefinisikan *basis set* dari jalur eksekusi (*execution path*). *Test case* yang dilakukan untuk menggunakan *basis set* tersebut dijamin untuk menggunakan setiap *statement* di dalam program paling tidak sekali selama pengujian. Sebelum metode *basis path* dapat diperkenalkan, notasi sederhana untuk representasi aliran kontrol yang disebut diagram alir (*flow graph*) harus diperkenalkan. Setiap representasi desain prosedural yang berupa *flow chart* dapat diterjemahkan ke dalam *flow graph*. Gambar 2.9 menunjukkan transformasi *flow chart* ke *flow graph*. Setelah *flow graph* didefinisikan maka harus ditentukan ukuran kompleksitas (*cyclomatic complexity*).



**Gambar 2.6** Transformasi *Flow Chart* ke *Flow Graph*

Sumber : [PRE-01]

*Cyclomatic complexity* adalah metrik perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Bila metrik ini digunakan dalam konteks metode pengujian *basis path*, maka nilai yang dihitung untuk *cyclomatic complexity* menentukan jumlah jalur independen (*independent path*) dalam *basis set* suatu program dan memberi batas atas bagi jumlah pengujian yang harus dilakukan untuk memastikan bahwa semua *statement* telah dieksekusi sedikitnya satu kali.

Jalur independen adalah jalur yang melalui program yang mengenalkan sedikitnya satu rangkaian *statement* proses baru atau suatu kondisi baru. Untuk menentukan *cyclomatic complexity* bisa dilakukan dengan beberapa cara, diantaranya [PRE-01]:

1. Jumlah region pada flow graph sesuai dengan cyclomatic complexity.
2. Cyclomatic complexity  $V(G)$ , untuk grafik  $G$  adalah  $V(G) = E - N + 2$ , dimana  $E$  adalah jumlah edge, dan  $N$  adalah jumlah node.

$V(G) = P + 1$ , dimana  $P$  adalah jumlah *predicate node* yaitu *node* yang merupakan kondisi (ada 2 atau lebih *edge* akan keluar *node* ini).

### 2.15.2 *Black-Box Testing*

*Black-Box testing* dapat mengidentifikasi kesalahan yang berhubungan dengan kesalahan fungsionalitas perangkat lunak yang tampak dalam kesalahan output. Pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada output yang dihasilkan yang merespon *input* yang dipilih dan kondisi dieksekusi.

Pengujian yang dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu. Pengujian yang dilakukan untuk *interface* perangkat lunak, pengujian ini dilakukan untuk memperlihatkan bahwa fungsi-fungsi bekerja dengan baik dalam arti masukan yang diterima dengan benar dan keluaran yang dihasilkan benar-benar tepat, pengintegrasian dari eksternal data berjalan dengan baik [LES-13].

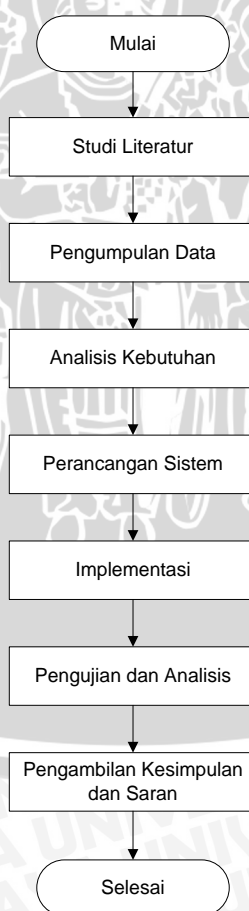


### BAB III

## METODOLOGI PENELITIAN

Penelitian yang diusulkan bersifat rancang bangun dan analisis implementasi hasil rancang bangun. Pendekatan yang digunakan dalam rancang bangun aplikasi ini adalah pendekatan *user/customer oriented*, yaitu pendekatan dari sudut pandang pengguna atau *user experience* yang dititikberatkan pada fungsionalitas (*usability*) dari aplikasi sebagai suatu sistem penyampaian informasi yang atraktif, mudah digunakan, dan membantu *user* secara penuh.

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam perancangan, implementasi dan pengujian dari aplikasi perangkat lunak yang akan dikembangkan. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan perangkat lunak selanjutnya. Berikut ini merupakan diagram alir runtutan pengerjaan penelitian ini :



**Gambar 3. 1** Diagram Tahapan Proses dalam Penelitian

### 3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan tugas akhir. Teori-teori pendukung tersebut meliputi :

- a. Pengenalan ATM
- b. Sistem Operasi Android
- c. Aplikasi Android Berbasis Web
- d. *Location Based Service* (LBS)
- e. Pengenalan HTML5
- f. Pengenalan CSS3
- g. Pengenalan JavaScript
- h. Ratchet
- i. PhoneGap
- j. Pengenalan jQuery
- k. Pengenalan PHP dan MySQL
- l. Google Map
- m. Google Map API
- n. Algoritma Haversine
- o. Teknik Pengujian *Black-Box* dan *White-Box*

### 3.2 Pengumpulan Data

Perangkat lunak yang dikembangkan dalam tugas akhir ini adalah Aplikasi Android Pencarian Lokasi ATM Terdekat di kota Malang. Adapun teknik yang digunakan dalam pengumpulan data yaitu dengan riset pustaka. Riset pustaka dilakukan dengan cara mengumpulkan data-data ATM yang diperoleh dari website resmi dari bank yang bersangkutan. Data ATM yang digunakan untuk sistem pencarian lokasi ATM terdekat akan dilampirkan pada Lampiran 1.

### 3.3 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (*requirements*) sistem dan siapa saja yang terlibat di dalamnya. Analisis juga dilakukan untuk mengetahui kondisi lapangan

yang ada sehingga dapat diketahui implementasi perangkat lunak yang akan digunakan. Metode analisis yang digunakan adalah *Object Oriented Analysis* dengan menggunakan bahasa pemodelan UML (*Unified Modeling Language*).

*Use case* diagram digunakan untuk mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas sistem dari perspektif *end-user*. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (*requirements*) perangkat lunak yang kemudian akan dimodelkan dalam *use case* diagram.

Tahap analisis kebutuhan terdiri atas empat langkah yaitu melakukan penjabaran tentang gambaran umum aplikasi, melakukan proses identifikasi aktor yang terlibat di dalam aplikasi, membuat daftar kebutuhan pengguna menggunakan pemodelan diagram *use case*. Proses analisis kebutuhan ini diawali dengan penjabaran gambaran umum aplikasi Android *mobile* pencarian lokasi ATM, penjabaran tentang daftar kebutuhan dan kemudian memodelkannya ke dalam diagram *use case*. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

### 3.4 Perancangan Sistem

Perancangan perangkat lunak dilakukan setelah semua kebutuhan perangkat lunak didapatkan melalui tahap analisis kebutuhan. Perancangan perangkat lunak dilakukan berdasarkan *object oriented analysis* dan *object oriented design* yaitu menggunakan pemodelan UML (*Unified Modeling Language*). Perancangan dimulai dari perancangan alur atau aktifitas yang dilakukan *user* secara *procedural* yang dimodelkan dalam *activity* diagram. Selanjutnya, dilakukan perancangan *sequence* diagram, untuk menggambarkan perilaku sistem secara dinamis dan memperlihatkan interaksi dari objek-objek yang terlibat di dalamnya. Perancangan algoritma Haversine sebagai cara perhitungan jarak terdekat juga dilakukan pada tahap ini. Kemudian tahap perancangan dilanjutkan dengan perancangan antarmuka pengguna.

### 3.5 Implementasi

Pada tahap ini dilakukan proses implementasi dalam aplikasi komputer

menggunakan bahasa pemrograman yang telah dipilih yang sesuai dengan analisis dan perancangan yang sudah dilakukan. Implementasi perangkat lunak mengacu kepada perancangan perangkat lunak. Implementasi perangkat lunak diawali dengan penjabaran spesifikasi lingkungan perancangan perangkat lunak. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman HTML, CSS, dan JavaScript. Implementasi antarmuka berdasarkan perancangan yang telah dilakukan. Pada tahap akhir dilakukan implementasi simulasi pada *hardware* secara langsung menggunakan *Android mobile device*.

### 3.6 Pengujian dan Analisis

Pengujian perangkat lunak pada tugas akhir ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang telah dianalisa sebelumnya. Strategi pengujian perangkat lunak yang akan digunakan yaitu pengujian validasi dan pengujian kompatibilitas.

Proses pengujian perangkat lunak pada tugas akhir ini terdapat tiga tahap (strategi) yaitu pengujian unit, pengujian validasi, dan pengujian kompatibilitas. Pada pengujian unit digunakan *White-Box Testing* dengan teknik *Basic Path Testing*. Pada teknik *Basic Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan. Pengujian validasi menggunakan metode pengujian *Black-Box Testing*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara fungsi sistem dengan daftar kebutuhan. Pada pengujian kompatibilitas dilakukan pengujian dengan menggunakan *Android mobile device* dengan tiga versi Android yang berbeda, yaitu Android versi 2.3.5, versi 4.1.2, dan versi 4.2.2, untuk mengetahui kompatibilitas aplikasi terhadap *Android mobile device* yang akan digunakan. Analisis juga dilakukan untuk mengetahui hasil dari pengujian perangkat lunak sehingga dapat didapatkan kesimpulan dari pengembangan perangkat lunak yang telah dilakukan.

### 3.7 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan perangkat lunak, implementasi perangkat lunak, dan pengujian perangkat lunak telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan perangkat lunak lebih lanjut.





## BAB IV

### PERANCANGAN DAN IMPLEMENTASI

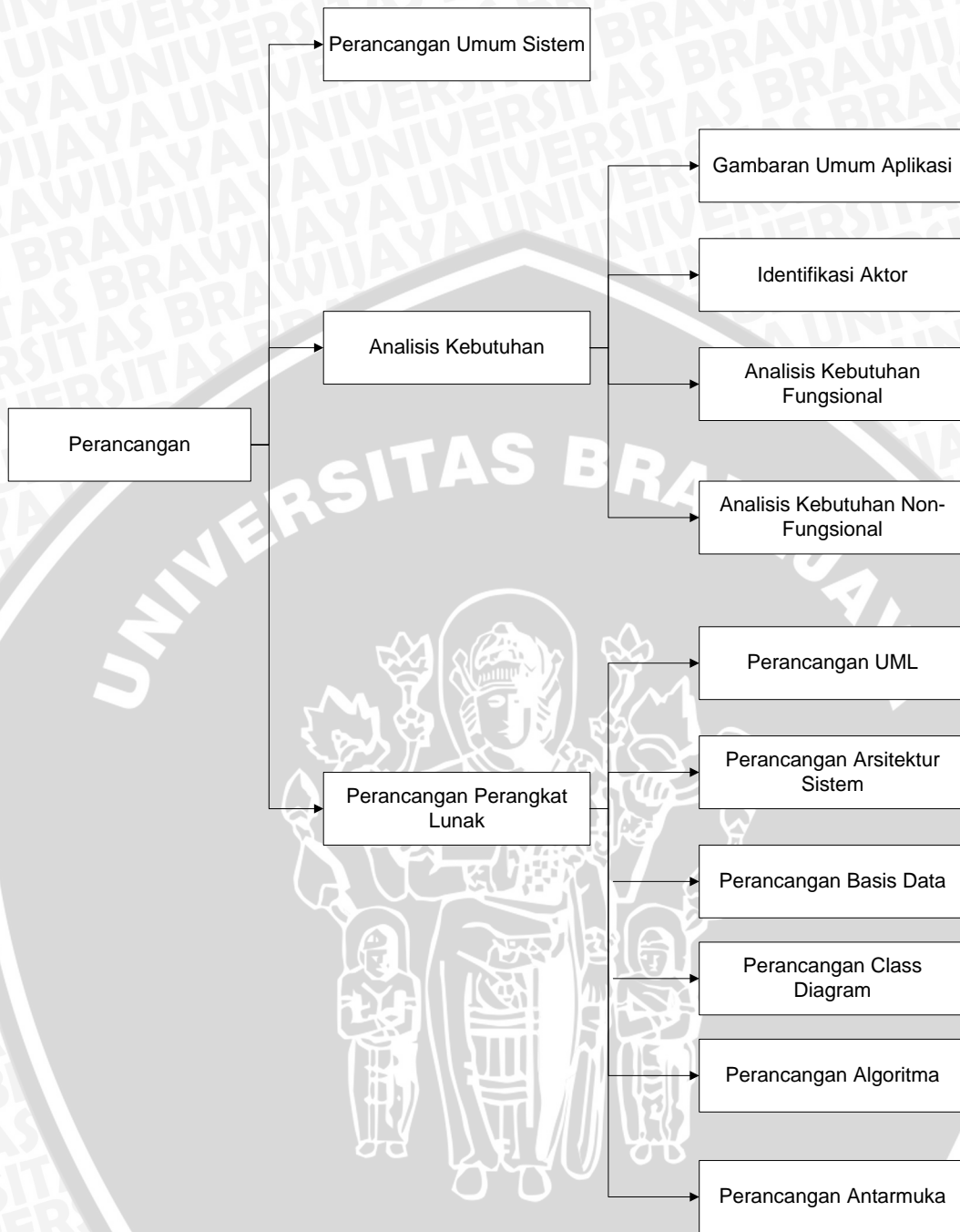
Bab ini membahas mengenai perancangan dan implementasi aplikasi. Perancangan aplikasi dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Perancangan yang dilakukan meliputi empat tahap yaitu perancangan umum sistem, proses analisis kebutuhan, perancangan algoritma dan perancangan perangkat lunak. Sedangkan pembahasan implementasi terdiri atas penjelasan tentang spesifikasi lingkungan implementasi, batasan-batasan dalam implementasi, implementasi format data JSON, implementasi algoritma dan implementasi antarmuka aplikasi.

#### 4.1 Perancangan

Pada proses perancangan aplikasi terdapat tiga tahapan, tahap pertama adalah perancangan umum sistem, tahap kedua adalah analisis kebutuhan yang akan digunakan pada perancangan perangkat lunak, kemudian tahap ketiga adalah perancangan perangkat lunak. Tahap perancangan umum sistem menjelaskan mengenai gambar proses kerja sistem secara umum. Tahap analisis kebutuhan terdiri atas empat langkah yaitu melakukan penjabaran tentang gambaran umum aplikasi, melakukan proses identifikasi aktor yang terlibat dalam aplikasi, membuat daftar kebutuhan pengguna dengan menganalisa kebutuhan fungsional dan kebutuhan non fungsional. Sedangkan pada proses perancangan perangkat lunak memiliki empat langkah, yaitu perancangan UML, perancangan basis data, perancangan algoritma dan perancangan antarmuka.

Mengenai tahap-tahap dari perancangan sistem aplikasi Android akan dijelaskan dalam Gambar 4.1

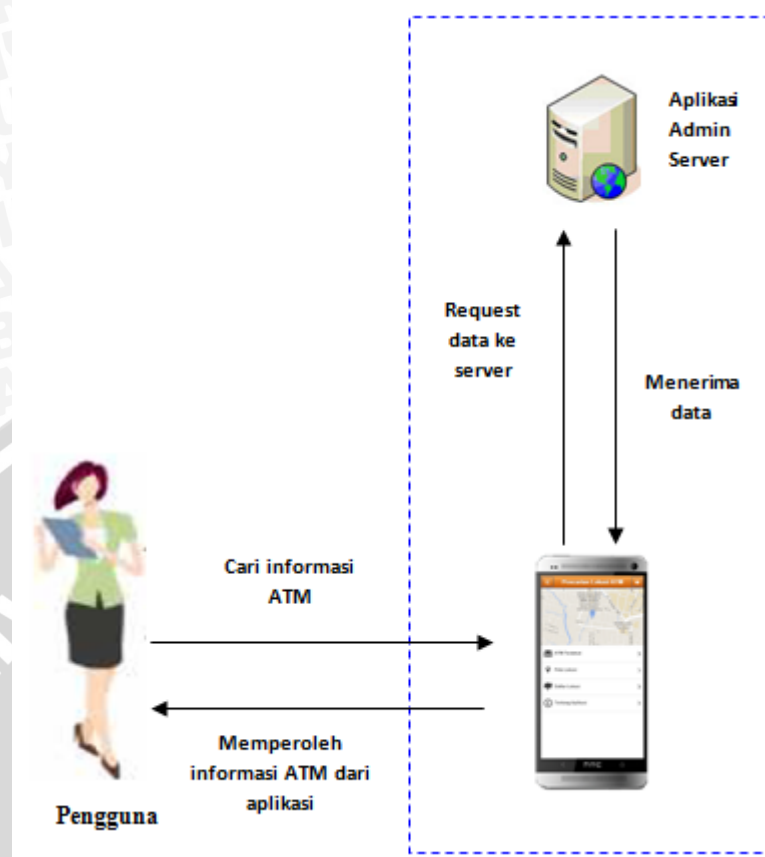




**Gambar 4. 1** Diagram Perancangan

#### 4.2 Perancangan Umum Sistem

Perancangan umum sistem merupakan tahapan awal dari perancangan perangkat lunak. Perancangan sistem dilakukan untuk mempresentasikan arsitektur sistem yang akan dibuat secara umum. Gambar 4.2 berikut menunjukkan perancangan umum sistem.



Gambar 4. 2Perancangan Umum sistem

### 4.3 Analisis Kebutuhan

Proses analisis kebutuhan ini diawali dengan penjabaran analisis sistem, identifikasi aktor, dan analisis kebutuhan sistem yang terdiri atas kebutuhan fungsional yang akan dimodelkan dalam bentuk *use case* diagram, serta kebutuhan non fungsional. Analisis kebutuhan ini bertujuan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

#### 4.3.1 Gambaran Umum Aplikasi

Pembahasan gambaran umum aplikasi Android *mobile* pencarian lokasi ATM terdiri atas dua bagian, yaitu deskripsi umum aplikasi pencarian lokasi ATM dan lingkungan aplikasi pencarian lokasi ATM.

#### 4.3.1.1 Deskripsi Aplikasi

Aplikasi pencarian lokasi ATM ini merupakan aplikasi yang dirancang khusus untuk *platform* Android berbasis *mobile*. Aplikasi ini dikembangkan untuk membantu pengguna dalam mencari posisi pengguna saat ini, kemudian dari posisi sekarang itu, pengguna dapat mencari lokasi ATM terdekat dari posisi pengguna. Aplikasi ini juga dapat membantu pengguna untuk mendapatkan arah tujuan menuju lokasi ATM yang ingin dituju. Aplikasi Android *mobile* pencarian lokasi ATM mengaplikasikan LBS (*Location Based Service*), untuk menentukan posisi pengguna saat ini. Sistem ini menggunakan MySQL sebagai *database* sisi *server* yang dapat menyimpan segala informasi mengenai data ATM bank yang meliputi nama bank, alamat, jenis ATM, *longitude*, dan *latitude*. Sistem ini dibagi menjadi dua bagian yaitu bagian *front-end* dan *back-end*. *Front-end* adalah bagian yang langsung dapat dilihat serta dipergunakan oleh pengguna untuk berinteraksi dengan sistem. Sedangkan *back-end* adalah bagian yang menangani masalah *database management*, dalam aplikasi ini adalah bagian admin. Aplikasi pencarian lokasi ATM ini dikembangkan dengan membuat *online storage* pada *web service* yang terkoneksi dengan *server* sebagai pusat pengambilan data dan menggunakan JSON sebagai cara pengambilan data.

#### 4.3.1.2 Lingkungan Aplikasi

Aplikasi Android *mobile* pencarian lokasi ATM terdekat ini membutuhkan suatu lingkungan yang digunakan sebagai tempat berjalannya aplikasi. Secara keseluruhan aplikasi Android *mobile* ini berbasis *hybrid mobile application*, sehingga membutuhkan sebuah *device* untuk menjalankan aplikasi tersebut. Dengan penggunaan teknologi *mobile smartphone* Android, aplikasi ini dapat dijalankan di semua lingkungan *device* Android.

#### 4.3.2 Identifikasi Aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor-aktor yang akan berinteraksi dengan aplikasi. Pada Tabel 4.1 memperlihatkan aktor-aktor yang terlibat beserta penjelasannya.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna adalah orang yang berada di Kota Malang yang dapat menggunakan aplikasi untuk mencari lokasi ATM yang tersebar di kota Malang.
Admin	Admin adalah orang yang memiliki wewenang penuh dalam sistem untuk meng- <i>input</i> , mengubah, dan menghapus data ATM.

### 4.3.3 Analisis Kebutuhan Fungsional

Daftar kebutuhan terdiri dari kebutuhan fungsional dan non fungsional. Pada daftar kebutuhan fungsional akan dispesifikasikan yaitu spesifikasi kebutuhan fungsional dari pengguna dan admin yang ditunjukkan pada Tabel 4.2 dengan per nomer menggunakan SRS (*Software Requirement Specification*).

Tabel 4. 2 Spesifikasi Kebutuhan Fungsional

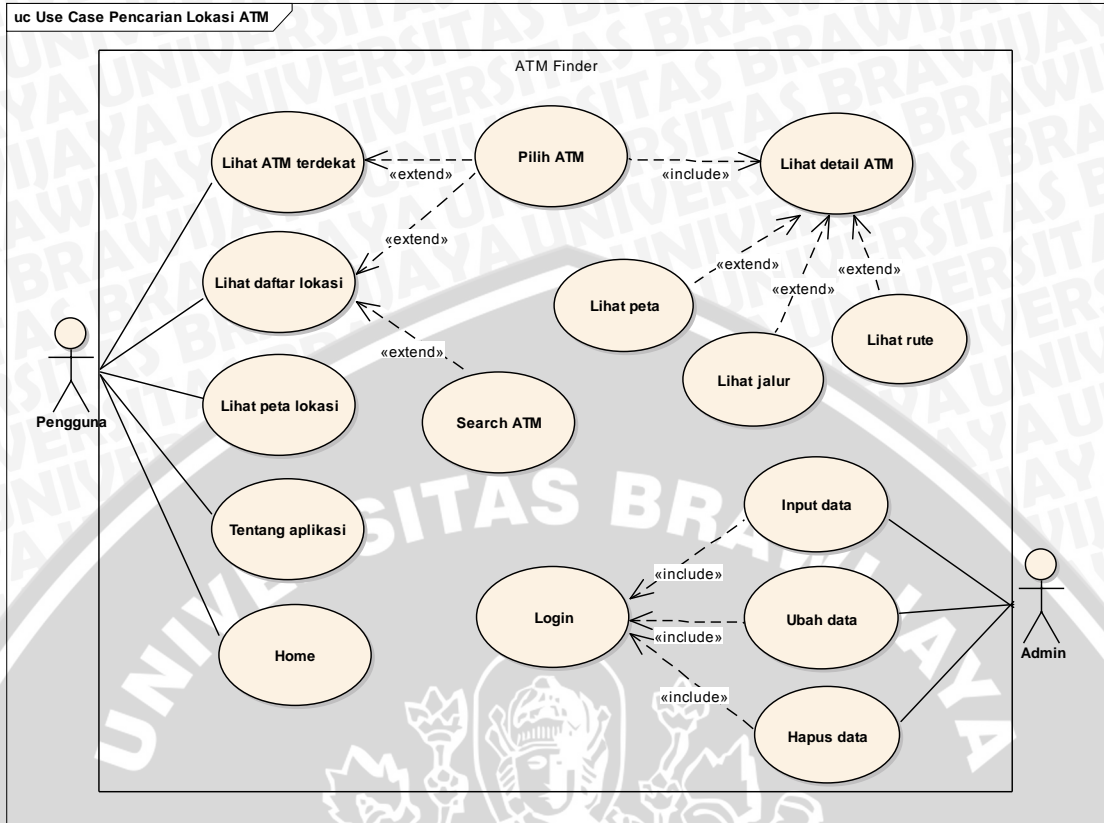
No	Kebutuhan	Use Case
SRS_001_01	Menyimpan data dan informasi mengenai ATM yang ada di kota Malang (nama bank, alamat, <i>longitude</i> , <i>latitude</i> , tipe ATM)	Input data lokasi ATM
SRS_001_02	Mengubah data dan informasi mengenai ATM (nama bank, lokasi, alamat, <i>longitude</i> , <i>latitude</i> , tipe ATM)	Ubah data lokasi ATM
SRS_001_03	Menghapus data dan informasi mengenai ATM (nama bank, lokasi, alamat, <i>longitude</i> , <i>latitude</i> , tipe ATM)	Hapus data lokasi ATM
SRS_002_01	Mampu menampilkan data lokasi ATM yang terdekat dari posisi pengguna	Lihat ATM terdekat
SRS_002_02	Memilih salah satu ATM yang diinginkan oleh pengguna	Pilih ATM
SRS_002_03	Mampu menampilkan detail data berupa informasi dari ATM yang telah dipilih	Lihat detail ATM

SRS_002_04	Mampu menampilkan visualisasi lokasi ATM dalam bentuk peta	Lihat peta
SRS_002_05	Mampu menampilkan jalur yang harus dituju oleh pengguna untuk dapat sampai ke lokasi ATM yang dipilih dan divisualisasikan dalam bentuk peta	Lihat jalur
SRS_002_06	Mampu menampilkan rute yang harus ditempuh oleh pengguna untuk dapat sampai ke lokasi ATM yang dipilih	Lihat rute
SRS_003_01	Mampu menampilkan daftar bank yang ada di kota Malang beserta jumlah ATM yang tersebar	Lihat daftar lokasi
SRS_003_02	Mampu menampilkan fitur <i>search</i> guna mencari ATM yang diinginkan dari menu daftar lokasi	<i>Search</i> ATM
SRS_004_01	Mampu menampilkan visualisasi lokasi ATM yang tersebar di kota Malang dalam bentuk peta, dan ditandai dengan <i>marker</i>	Lihat peta lokasi
SRS_005_01	Mampu menampilkan informasi mengenai pembuat aplikasi	Tentang Aplikasi
SRS_006_01	Mampu menampilkan informasi berupa menu-menu utama yang terdapat pada aplikasi	<i>Home</i>

Selanjutnya daftar kebutuhan fungsional akan lebih dijabarkan menggunakan diagram *use case*.

### 1. Diagram *Use Case*

Pada sistem ini pemodelan kebutuhan fungsional dimodelkan menggunakan diagram *use case*. Diagram *use case* adalah salah satu diagram untuk memodelkan aspek perilaku sistem. Diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. *Use case* merupakan fungsionalitas dari sistem yang diinisialisasi oleh aktor. Diagram *use case* untuk aplikasi ditunjukkan dalam Gambar 4.2 berikut ini :



Gambar 4.3 Diagram Use Case

## 2. Skenario Use Case

Secara lebih mendetail, masing-masing *use case* yang terdapat pada diagram *use case*, dijabarkan dalam skenario *use case*. Di dalam skenario *use case*, akan diberikan uraian nama *use case*, aktor yang berhubungan dengan *use case* tersebut, tujuan dari *use case*, deskripsi global tentang *use case*, kondisi awal yang harus dipenuhi dan kondisi akhir yang diharapkan setelah berjalannya fungsional *use case*. Selain itu juga akan diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang diberikan oleh aktor.

### a. Skenario Use Case Login

Kebutuhan fungsional yang harus disediakan adalah admin dapat melakukan proses *login* dengan memasukkan *username* dan *password* agar dapat masuk ke halaman admin. Kebutuhan tersebut direpresentasikan oleh *use case* input data lokasi ATM oleh admin. Tabel 4.3 merupakan skenario *use case login*.

**Tabel 4. 3** Skenario *Use Case Login*

<b>Nama</b>	<i>Login</i>
<b>Kode SRS</b>	SRS_001_01
<b>Tujuan</b>	Admin dapat masuk ke halaman admin
<b>Aktor</b>	Admin
<b>Kondisi Awal (Pre-Condition)</b>	Admin membuka halaman bagian admin
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan admin
<b>Skenario (Flow of events)</b>	Skenario Dasar ( <i>Basic Flow</i> ) : 1. Admin membuka halaman bagian admin 2. Admin memasukkan <i>username</i> dan <i>password</i>
<b>Kondisi Akhir (Post-Condition)</b>	Jika <i>username</i> dan <i>password</i> benar, maka admin berhasil <i>login</i> ke sistem. Jika salah, maka akan ditampilkan pesan informasi, dan keluar tampilan untuk mengulang memasukkan <i>username</i> dan <i>password</i>

#### b. Skenario Use Case Input Data Lokasi ATM

Kebutuhan fungsional yang harus disediakan adalah dapat menyimpan data dan informasi mengenai ATM. Kebutuhan tersebut direpresentasikan oleh *use case* input data lokasi ATM oleh admin. Tabel 4.4 merupakan skenario *use case* input data lokasi ATM.

**Tabel 4. 4** Skenario *Use Case Input Data Lokasi ATM*

<b>Nama</b>	<i>Input data lokasi ATM</i>
<b>Kode SRS</b>	SRS_001_02
<b>Tujuan</b>	Memasukkan data lokasi ATM
<b>Aktor</b>	Admin
<b>Kondisi Awal (Pre-Condition)</b>	Admin membuka halaman bagian admin
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan admin
<b>Skenario (Flow of events)</b>	Skenario Dasar ( <i>Basic Flow</i> ) : 3. <i>Use case</i> ini akan dimulai ketika admin membuka



	aplikasi pencarian lokasi ATM 4. Admin <i>login</i> ke sistem 5. Admin memasukkan data lokasi ATM ke dalam sistem
<b>Kondisi Akhir</b> ( <i>Post-Condition</i> )	Aplikasi menyimpan data lokasi ATM yang telah di- <i>input</i> oleh admin ke dalam sistem

### c. Skenario *Use case* Ubah Data Lokasi ATM

Kebutuhan fungsional yang harus disediakan adalah dapat mengubah data dan informasi mengenai ATM. Kebutuhan tersebut direpresentasikan oleh *use case* ubah data lokasi ATM oleh admin. Tabel 4.5 merupakan skenario *use case* ubah data lokasi ATM.

**Tabel 4.5** Skenario *Use Case* Ubah Data Lokasi ATM

<b>Nama</b>	Ubah data lokasi ATM
<b>Kode SRS</b>	SRS_001_03
<b>Tujuan</b>	Mengubah data lokasi ATM
<b>Aktor</b>	Admin
<b>Kondisi Awal</b> ( <i>Pre-Condition</i> )	Admin membuka halaman bagian admin
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan admin
<b>Skenario</b> ( <i>Flow of events</i> )	Skenario Dasar ( <i>Basic Flow</i> ) : a. <i>Use case</i> ini akan dimulai ketika admin membuka aplikasi pencarian lokasi ATM b. Admin <i>login</i> ke sistem c. Admin mengubah data lokasi ATM ke dalam sistem
<b>Kondisi Akhir</b> ( <i>Post-Condition</i> )	Aplikasi menyimpan data lokasi ATM yang telah diubah oleh admin ke dalam sistem

### d. Skenario *Use case* Hapus Data Lokasi ATM

Kebutuhan fungsional yang harus disediakan adalah dapat mengubah data dan informasi mengenai ATM. Kebutuhan tersebut direpresentasikan oleh *use case* ubah data lokasi ATM oleh admin. Tabel 4.6 merupakan skenario *use case* ubah

data lokasi ATM.

**Tabel 4. 6** Skenario *Use Case* Hapus Data Lokasi ATM

<b>Nama</b>	Hapus data lokasi ATM
<b>Kode SRS</b>	SRS_001_04
<b>Tujuan</b>	Menghapus data lokasi ATM
<b>Aktor</b>	Admin
<b>Kondisi Awal (Pre-Condition)</b>	Admin membuka halaman bagian admin
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan admin
<b>Skenario (Flow of events)</b>	Skenario Dasar ( <i>Basic Flow</i> ) : a. <i>Use case</i> ini akan dimulai ketika admin membuka aplikasi pencarian lokasi ATM b. Admin <i>login</i> ke sistem c. Admin menghapus data lokasi ATM ke dalam sistem
<b>Kondisi Akhir (Post-Condition)</b>	Aplikasi menyimpan data lokasi ATM yang telah dihapus oleh admin ke dalam sistem

**e. Skenario *Use case* Home**

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan informasi berupa menu-menu utama yang terdapat pada aplikasi. Kebutuhan tersebut direpresentasikan oleh *use case home*. Tabel 4.7 merupakan skenario *use case home*.

**Tabel 4. 7** Skenario *Use Case* Home

<b>Nama</b>	<i>Home</i>
<b>Kode SRS</b>	SRS_006_01
<b>Tujuan</b>	Untuk menampilkan informasi berupa menu-menu utama yang terdapat pada aplikasi
<b>Aktor</b>	Pengguna
<b>Kondisi Awal (Pre-Condition)</b>	Pengguna membuka aplikasi terlebih dahulu sebelum <i>use case</i> dimulai
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan

	pengguna
<b>Skenario</b> ( <i>Flow of events</i> )	Skenario Dasar ( <i>Basic Flow</i> ) : a. <i>Use case</i> ini akan dimulai ketika pengguna membuka aplikasi pencarian lokasi ATM b. Pengguna memilih menu <i>home</i> c. Sistem akan menampilkan menu-menu utama yang ada pada aplikasi
<b>Kondisi Akhir</b> ( <i>Post-Condition</i> )	Pengguna dapat melihat menu-menu utama pada halaman <i>home</i>

#### f. Skenario *Use case* Lihat ATM Terdekat

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan data lokasi ATM yang terdekat dari posisi pengguna. Kebutuhan tersebut direpresentasikan oleh *use case* lihat ATM terdekat. Tabel 4.8 merupakan skenario *use case* lihat ATM terdekat.

**Tabel 4.8** Skenario *Use Case* Lihat ATM Terdekat

<b>Nama</b>	Lihat ATM terdekat
<b>Kode SRS</b>	SRS_002_01
<b>Tujuan</b>	Untuk melihat daftar ATM terdekat dari posisi pengguna
<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b> ( <i>Pre-Condition</i> )	Pengguna membuka dan masuk ke dalam aplikasi terlebih dahulu sebelum <i>use case</i> dimulai, kemudian memilih menu ATM terdekat pada halaman utama ( <i>home</i> )
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario</b> ( <i>Flow of events</i> )	Skenario Dasar ( <i>Basic Flow</i> ) : a. <i>Use case</i> ini akan dimulai ketika pengguna membuka dan masuk ke dalam aplikasi pencarian lokasi ATM dan memilih menu ATM terdekat pada halaman utama ( <i>home</i> ) b. Sistem akan menampilkan pilihan ATM yang terdekat

	dari posisi pengguna yang tersimpan pada <i>database</i>
<b>Kondisi Akhir</b> ( <i>Post-Condition</i> )	Pengguna dapat melihat dan memilih daftar ATM terdekat dari posisi pengguna

#### g. Skenario Use Case Pilih ATM

Kebutuhan fungsional yang harus disediakan adalah sistem dapat menampilkan data ATM dan pengguna dapat memilih ATM yang diinginkan. Kebutuhan tersebut direpresentasikan oleh *use case* pilih ATM. Tabel 4.9 merupakan skenario *use case* pilih ATM.

**Tabel 4.9** Skenario *Use Case* Pilih ATM

<b>Nama</b>	Pilih ATM
<b>Kode SRS</b>	SRS_002_02
<b>Tujuan</b>	Untuk memilih salah satu dari daftar ATM yang ditampilkan
<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b> ( <i>Pre-Condition</i> )	Pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu ATM terdekat
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario</b> ( <i>Flow of events</i> )	Skenario Dasar ( <i>Basic Flow</i> ): <ol style="list-style-type: none"> <li><i>Use case</i> ini akan dimulai ketika pengguna membuka dan masuk ke dalam aplikasi pencarian lokasi ATM dan memilih menu ATM terdekat atau menu daftar lokasi</li> <li>Sistem akan menampilkan daftar ATM</li> <li>Pengguna memilih salah satu ATM yang diinginkan</li> </ol>
<b>Kondisi Akhir</b> ( <i>Post-Condition</i> )	Pengguna dapat memilih salah satu ATM yang diinginkan, guna mendapatkan informasi lebih lanjut mengenai ATM yang telah dipilih

#### h. Skenario *Use case* Lihat Detail ATM

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan detail data berupa informasi dari ATM yang telah dipilih oleh pengguna. Kebutuhan tersebut direpresentasikan oleh *use case* lihat detail ATM. Tabel 4.10 merupakan skenario *use case* lihat detail ATM.

**Tabel 4. 10** Skenario *Use Case* Lihat Detail ATM

<b>Nama</b>	Lihat detail ATM
<b>Kode SRS</b>	SRS_002_03
<b>Tujuan</b>	Untuk melihat detail data mengenai informasi dari masing-masing ATM yang telah dipilih
<b>Aktor</b>	Pengguna
<b>Kondisi Awal (Pre-Condition)</b>	Pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu ATM terdekat dan menu daftar lokasi
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario (Flow of events)</b>	Skenario Dasar ( <i>Basic Flow</i> ) : <ol style="list-style-type: none"> <li><i>Use case</i> ini akan dimulai ketika pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu ATM terdekat atau daftar lokasi</li> <li>Sistem menampilkan daftar ATM dan pengguna memilih ATM yang diinginkan</li> <li>Sistem akan menampilkan detail informasi dari ATM yang telah dipilih oleh pengguna</li> </ol>
<b>Kondisi Akhir (Post-Condition)</b>	Pengguna dapat melihat informasi lebih detail dari ATM yang telah dipilih

#### i. Skenario *Use case* Lihat Peta

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan visualisasi lokasi ATM yang telah dipilih dalam bentuk peta. Kebutuhan tersebut direpresentasikan oleh *use case* lihat peta. Tabel 4.11 merupakan skenario *use case*

lihat peta.

**Tabel 4. 11** Skenario *Use Case* Lihat Peta

<b>Nama</b>	Lihat peta
<b>Kode SRS</b>	SRS_002_04
<b>Tujuan</b>	Untuk menampilkan visualisasi lokasi ATM yang telah dipilih oleh pengguna dalam bentuk peta
<b>Aktor</b>	Pengguna
<b>Kondisi Awal (Pre-Condition)</b>	Pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu ATM terdekat dan menu daftar lokasi
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario (Flow of events)</b>	Skenario Dasar ( <i>Basic Flow</i> ) : a. <i>Use case</i> ini akan dimulai ketika pengguna membuka dan masuk ke dalam aplikasi pencarian lokasi ATM b. Pengguna memilih salah satu ATM yang diinginkan c. Pada tampilan detail informasi ATM, pengguna memilih menu peta
<b>Kondisi Akhir (Post-Condition)</b>	Pengguna dapat melihat lokasi ATM yang telah dipilih pada peta, yang ditandai dengan <i>marker</i>

**j. Skenario *Use case* Lihat Jalur**

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan petunjuk jalur menuju ke lokasi ATM yang telah dipilih. Kebutuhan tersebut direpresentasikan oleh *use case* lihat jalur. Tabel 4.12 merupakan skenario *use case* lihat jalur.

**Tabel 4. 12** Skenario *Use Case* Lihat Jalur

<b>Nama</b>	Lihat jalur
<b>Kode SRS</b>	SRS_002_05
<b>Tujuan</b>	Untuk menampilkan jalur yang harus dituju oleh pengguna

	untuk dapat sampai ke lokasi ATM yang dipilih dan divisualisasikan dalam bentuk peta
<b>Aktor</b>	Pengguna
<b>Kondisi Awal (Pre-Condition)</b>	Pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu ATM terdekat dan menu daftar lokasi
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario (Flow of events)</b>	Skenario Dasar ( <i>Basic Flow</i> ) : a. <i>Use case</i> ini akan dimulai ketika pengguna membuka dan masuk ke dalam aplikasi pencarian lokasi ATM b. Pengguna memilih salah satu ATM yang diinginkan c. Pada tampilan detail informasi ATM, pengguna memilih menu petunjuk jalur
<b>Kondisi Akhir (Post-Condition)</b>	Pengguna dapat melihat petunjuk jalur menuju ATM yang telah dipilih dan divisualisasikan dalam bentuk peta

#### k. Skenario *Use case* Lihat Rute

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan petunjuk rute menuju ke lokasi ATM yang telah dipilih. Kebutuhan tersebut direpresentasikan oleh *use case* lihat rute. Tabel 4.13 merupakan skenario *use case* lihat rute.

**Tabel 4. 13** Skenario *Use Case* Lihat Rute

<b>Nama</b>	Lihat rute
<b>Kode SRS</b>	SRS_002_06
<b>Tujuan</b>	Untuk menampilkan rute yang harus ditempuh oleh pengguna untuk dapat sampai ke lokasi ATM yang dipilih
<b>Aktor</b>	Pengguna
<b>Kondisi Awal (Pre-Condition)</b>	Pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu ATM terdekat dan menu daftar lokasi

<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario</b> <i>(Flow of events)</i>	Skenario Dasar ( <i>Basic Flow</i> ) : c. <i>Use case</i> ini akan dimulai ketika pengguna membuka dan masuk ke dalam aplikasi pencarian lokasi ATM d. Pengguna memilih salah satu ATM yang diinginkan d. Pada tampilan detail informasi ATM, pengguna memilih menu petunjuk rute
<b>Kondisi Akhir</b> <i>(Post-Condition)</i>	Pengguna dapat melihat petunjuk rute menuju ATM yang telah dipilih

### 1. Skenario *Use case* Lihat Daftar Lokasi

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan daftar bank yang ada di kota Malang beserta jumlah ATM yang tersebar. Kebutuhan tersebut direpresentasikan oleh *use case* lihat daftar lokasi. Tabel 4.14 merupakan skenario *use case* lihat daftar lokasi.

**Tabel 4. 14** Skenario *Use Case* Lihat Daftar Lokasi

<b>Nama</b>	Lihat Daftar Lokasi
<b>Kode SRS</b>	SRS_003_01
<b>Tujuan</b>	Untuk melihat daftar bank yang ada di kota Malang serta jumlah ATM yang tersebar di area kota Malang
<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b> <i>(Pre-Condition)</i>	Pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu daftar lokasi dari halaman utama ( <i>home</i> )
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario</b> <i>(Flow of events)</i>	Skenario Dasar ( <i>Basic Flow</i> ) : a. <i>Use case</i> ini akan dimulai ketika pengguna membuka dan masuk ke dalam aplikasi pencarian lokasi ATM b. Pengguna memilih menu daftar lokasi



	c. Sistem akan menampilkan daftar bank yang ada di kota Malang beserta jumlah ATM yang tersebar
<b>Kondisi Akhir</b> ( <i>Post-Condition</i> )	Pengguna dapat memilih dan melihat daftar bank di kota Malang

#### m. Skenario *Use case Search ATM*

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan fitur *search ATM* dari tampilan daftar ATM yang ada di kota Malang. Kebutuhan tersebut direpresentasikan oleh *use case search ATM*. Tabel 4.15 merupakan skenario *use case* lihat daftar lokasi.

**Tabel 4. 15** Skenario *Use Case Search ATM*

<b>Nama</b>	Lihat <i>Search ATM</i>
<b>Kode SRS</b>	SRS_003_02
<b>Tujuan</b>	Untuk melakukan <i>search ATM</i> dari tampilan daftar ATM yang ada di kota Malang
<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b> ( <i>Pre-Condition</i> )	Pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu daftar lokasi pada halaman utama ( <i>home</i> )
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario</b> ( <i>Flow of events</i> )	<p>Skenario Dasar (<i>Basic Flow</i>) :</p> <ol style="list-style-type: none"> <li><i>Use case</i> ini akan dimulai ketika pengguna membuka dan masuk ke dalam aplikasi pencarian lokasi ATM</li> <li>Pengguna memilih menu daftar lokasi pada <i>home</i></li> <li>Sistem akan menampilkan daftar bank yang ada di kota Malang</li> <li>Ketika pengguna memilih salah satu bank, maka akan muncul daftar ATM dari bank tersebut yang tersebar di kota Malang</li> <li>Pengguna menggunakan fitur <i>search</i> untuk menemukan</li> </ol>

	ATM yang diinginkan
<b>Kondisi Akhir</b> ( <i>Post-Condition</i> )	Dengan fitur <i>search</i> , pengguna dapat lebih cepat menemukan ATM yang diinginkan

#### n. Skenario *Use case* Lihat Peta Lokasi

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan visualisasi lokasi ATM yang tersebar di area kota Malang dalam bentuk peta. Kebutuhan tersebut direpresentasikan oleh *use case* lihat peta lokasi. Tabel 4.16 merupakan skenario *use case* lihat peta lokasi.

**Tabel 4. 16** Skenario *Use Case* Lihat Peta Lokasi

<b>Nama</b>	Lihat Peta Lokasi
<b>Kode SRS</b>	SRS_004_01
<b>Tujuan</b>	Untuk menampilkan visualisasi lokasi ATM yang tersebar di kota Malang dalam bentuk peta, dan ditandai dengan <i>marker</i>
<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b> ( <i>Pre-Condition</i> )	Pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu peta lokasi pada halaman utama ( <i>home</i> )
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario</b> ( <i>Flow of events</i> )	Skenario Dasar ( <i>Basic Flow</i> ) : <ol style="list-style-type: none"> <li><i>Use case</i> ini akan dimulai ketika pengguna membuka aplikasi pencarian lokasi ATM</li> <li>Pengguna memilih menu peta lokasi</li> <li>Sistem akan menampilkan visualisasi lokasi ATM dalam bentuk peta dan ditandai dengan <i>marker</i></li> </ol>
<b>Kondisi Akhir</b> ( <i>Post-Condition</i> )	Pengguna dapat melihat lokasi ATM yang tersebar di area kota Malang pada peta yang ditandai dengan <i>marker</i>

#### o. Skenario *Use case* Tentang Aplikasi

Kebutuhan fungsional yang harus disediakan adalah dapat menampilkan informasi dari pembuat aplikasi pencarian lokasi ATM ini. Kebutuhan tersebut direpresentasikan oleh *use case* tentang aplikasi. Tabel 4.17 merupakan skenario *use case* tentang aplikasi.

**Tabel 4. 17** Skenario *Use Case* Tentang Aplikasi

<b>Nama</b>	Tentang Aplikasi
<b>Kode SRS</b>	SRS_005_01
<b>Tujuan</b>	Untuk menampilkan informasi dari pembuat aplikasi pencarian lokasi ATM ini
<b>Aktor</b>	Pengguna
<b>Kondisi Awal (Pre-Condition)</b>	Pengguna membuka dan masuk ke dalam aplikasi, kemudian memilih menu tentang plikasi pada halaman utama ( <i>home</i> )
<b>Karakteristik</b>	Tindakan dan eksekusi tergantung dari permintaan pengguna
<b>Skenario (Flow of events)</b>	Skenario Dasar ( <i>Basic Flow</i> ) : <ol style="list-style-type: none"> <li><i>Use case</i> ini akan dimulai ketika pengguna membuka dan masuk ke dalam aplikasi pencarian lokasi ATM</li> <li>Pengguna memilih menu tentang aplikasi pada <i>home</i></li> <li>Sistem akan menampilkan informasi mengenai pembuat aplikasi</li> </ol>
<b>Kondisi Akhir (Post-Condition)</b>	Pengguna dapat melihat informasi mengenai pembuat aplikasi

#### 4.3.4 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah analisis untuk mengetahui spesifikasi yang dibutuhkan oleh sistem. Ada beberapa parameter dan deskripsi kebutuhan yang akan digunakan dalam pengembangan, yaitu *Compatibility* dan *Usability*.

**Tabel 4. 18** Analisis Kebutuhan Non Fungsional

Paramater	Deskripsi Kebutuhan
<i>Compatibility</i>	Aplikasi harus dapat dijalankan di berbagai sistem operasi Android Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, dan KitKat
<i>Usability</i>	Tampilan antarmuka pada aplikasi dirancang seperti tampilan pada <i>mobile</i> . Dengan demikian pengguna aplikasi tidak mengalami kesulitan dalam menggunakan sistem. Selain itu bahasa yang digunakan adalah bahasa Indonesia, sehingga pengguna dapat lebih mudah memahami setiap fitur-fitur yang terdapat pada sistem

#### 4.4 Perancangan Perangkat Lunak

Proses perancangan perangkat lunak memiliki empat langkah, yaitu perancangan UML (*Unified Modelling Language*), perancangan basis data, perancangan algoritma, dan perancangan antarmuka. Perancangan aplikasi pada tugas akhir ini menggunakan pendekatan desain berorientasi objek yang direpresentasikan dengan menggunakan UML.

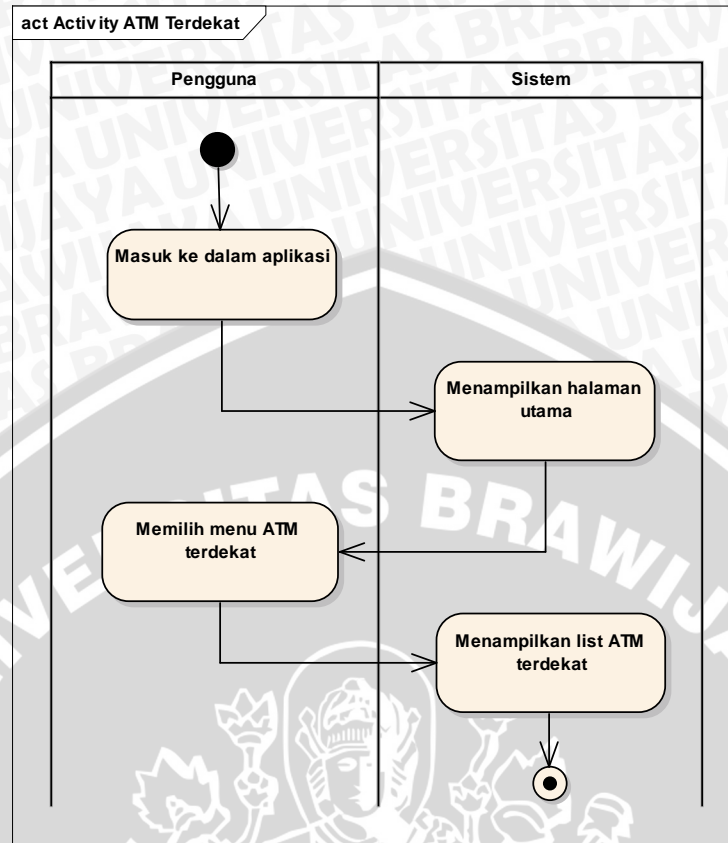
##### 4.4.1 Perancangan UML (*Unified Modelling Language*)

###### 4.4.1.1 Perancangan *Activity Diagram*

Diagram aktifitas (*Activity Diagram*) adalah diagram untuk memodelkan aktivitas antara pengguna dan sistem yang berjalan berdasarkan pada skenario *use case*.

##### 1. *Activity Diagram* untuk Lihat ATM Terdekat

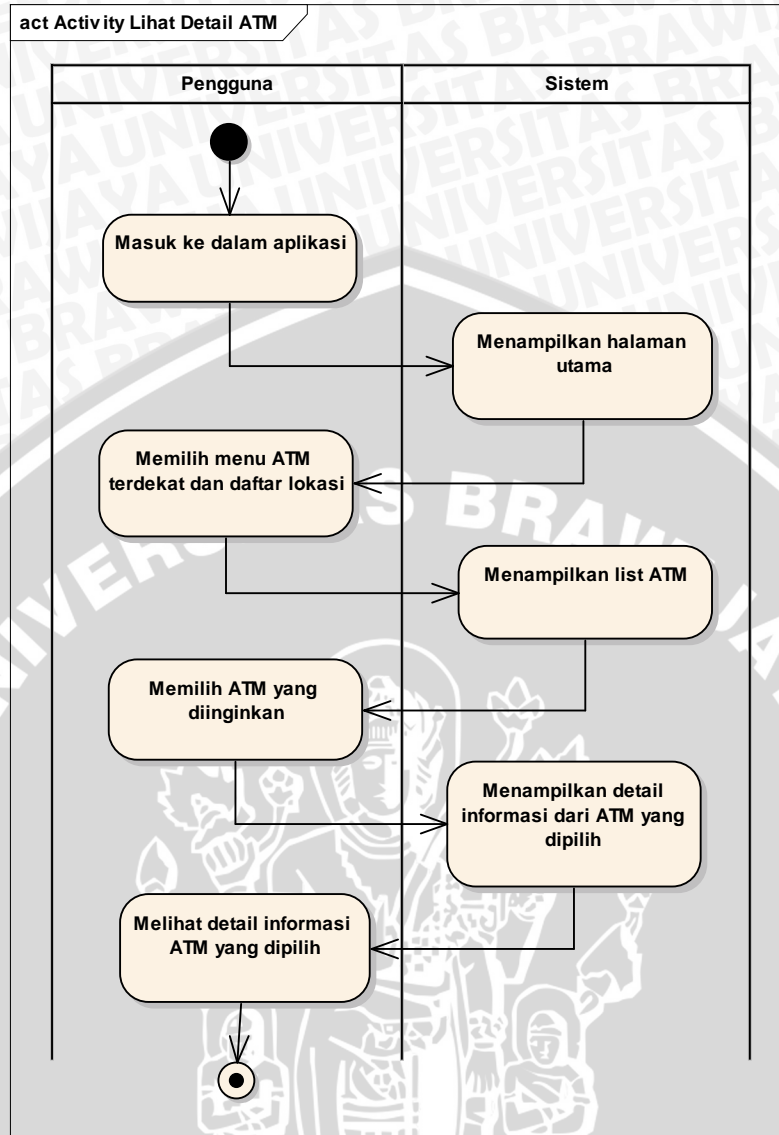
Dalam Gambar 4.4 ditunjukkan aktivitas yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *use case* lihat ATM terdekat yang ditunjukkan pada Tabel 4.8. Pengguna masuk aplikasi dan memilih menu ATM terdekat pada halaman utama (*home*), selanjutnya sistem menampilkan daftar ATM yang terdekat dari posisi pengguna.



Gambar 4. 4Activity Diagram untuk Lihat ATM Terdekat

## 2. Activity Diagram Lihat Detail ATM

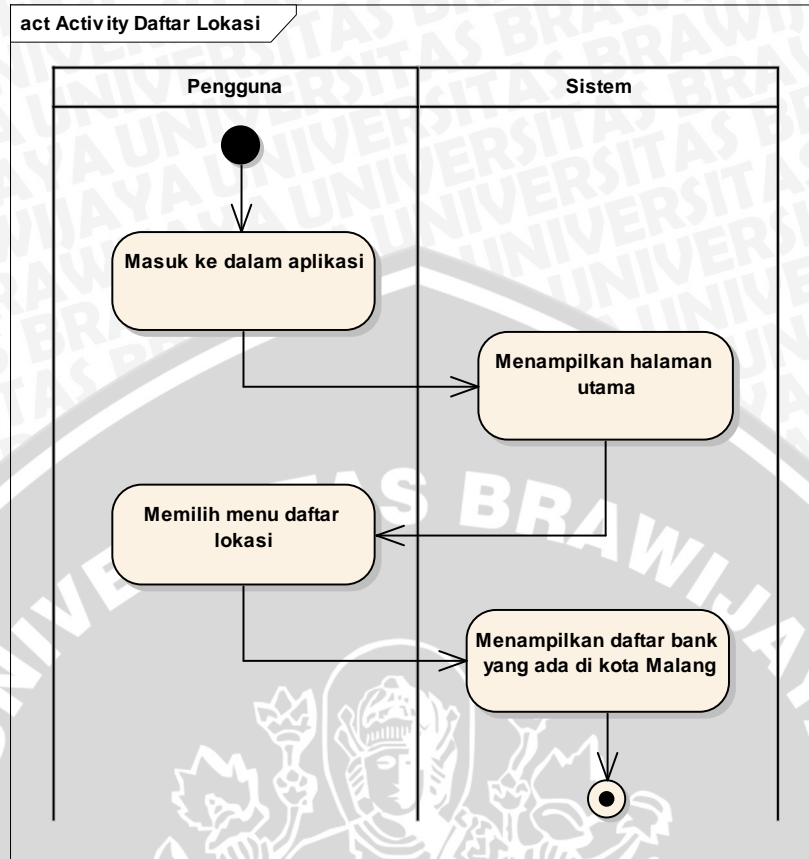
Pada Gambar 4.5 akan ditunjukkan aktivitas yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *use case* lihat detail ATM yang ditunjukkan pada Tabel 4.10. Pengguna masuk aplikasi dan memilih salah satu menu pencarian ATM pada menu utama. Pengguna dapat memilih menu ATM terdekat atau menu daftar lokasi. Kemudian sistem menampilkan daftar ATM dan pengguna memilih salah satu ATM yang diinginkan. Selanjutnya sistem menampilkan detail informasi dari ATM yang telah dipilih oleh pengguna.



Gambar 4. 5 Activity Diagram untuk Lihat Detail ATM

### 3. Activity Diagram Daftar Lokasi

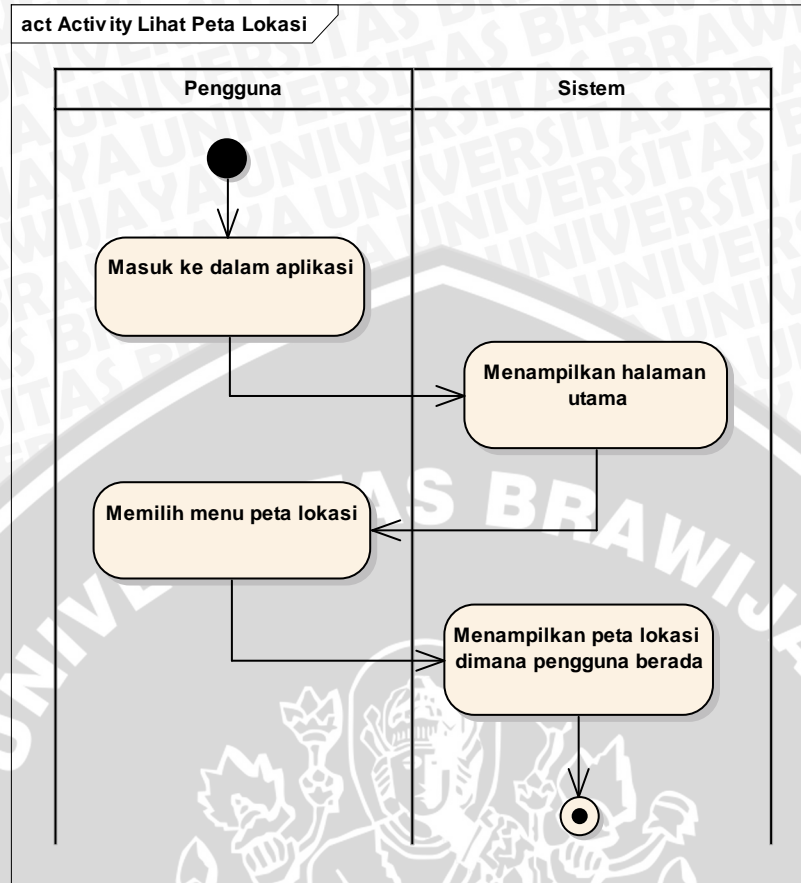
Pada Gambar 4.6 akan ditunjukkan aktivitas yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *use case* daftar lokasi yang ditunjukkan pada Tabel 4.14. Pengguna masuk aplikasi dan memilih menu daftar lokasi pada menu utama. Selanjutnya sistem menampilkan daftar bank yang ada di kota Malang.



Gambar 4. 6Activity Diagram untuk Daftar Lokasi

4. Activity Diagram Lihat Peta Lokasi

Pada Gambar 4.7 akan ditunjukkan aktivitas yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *use case* lihat peta lokasi yang ditunjukkan pada Tabel 4.16. Pengguna masuk aplikasi dan memilih menu peta lokasi pada menu utama. Kemudian sistem menampilkan lokasi ATM yang tersebar di area kota Malang pada peta dan ditandai dengan *marker*.

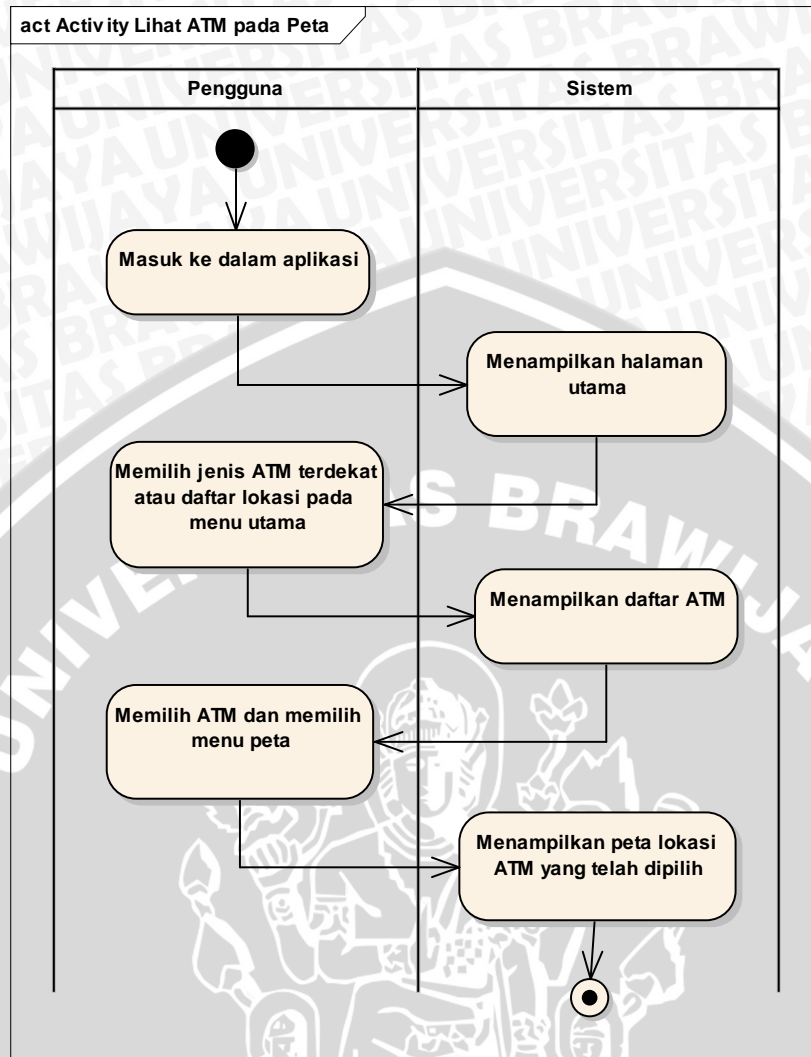


Gambar 4. 7Activity Diagram untuk Lihat Peta Lokasi

5. Activity Diagram Lihat ATM pada Peta

Pada Gambar 4.8 akan ditunjukkan aktivitas yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *use case* lihat peta yang ditunjukkan pada Tabel 4.11. Pengguna masuk aplikasi dan memilih menu ATM terdekat dan daftar lokasi pada halaman utama, kemudian sistem menampilkan daftar ATM. Selanjutnya pengguna memilih salah satu ATM dan pilih menu peta, dan sistem menampilkan peta lokasi ATM yang telah dipilih oleh pengguna yang ditandai dengan *marker*.

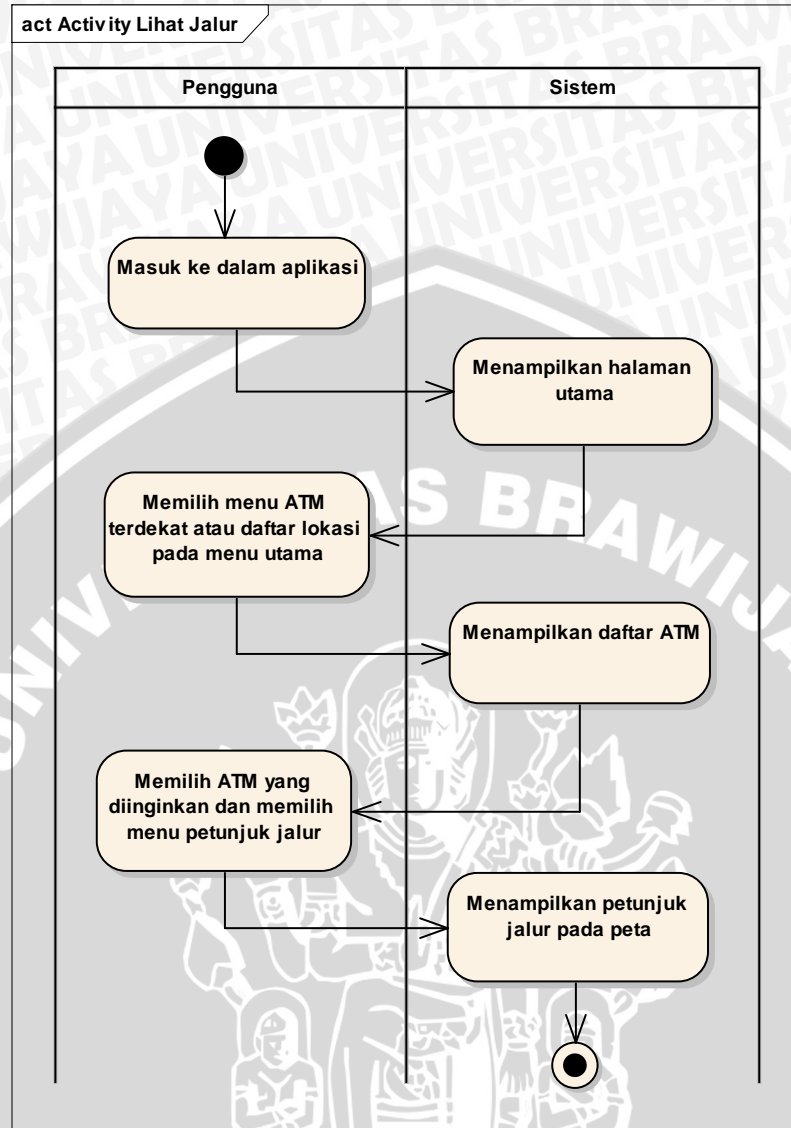




**Gambar 4. 8** Activity Diagram untuk Lihat ATM pada Peta

## 6. Activity Diagram Lihat Jalur

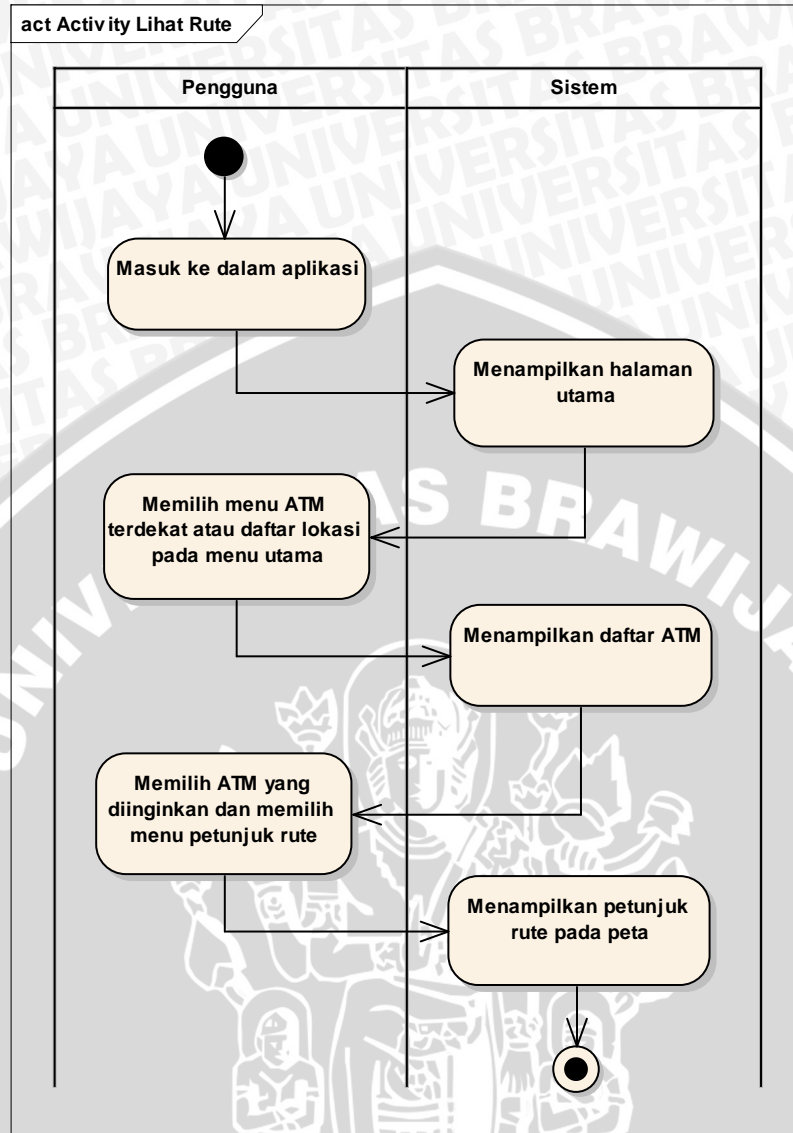
Pada Gambar 4.9 akan ditunjukkan aktivitas yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *use case* lihat jalur yang ditunjukkan pada Tabel 4.12. Pengguna masuk aplikasi dan memilih menu ATM terdekat dan daftar lokasi pada halaman utama, kemudian sistem menampilkan daftar ATM. Selanjutnya pengguna memilih salah satu ATM dan pilih menu petunjuk jalur.



Gambar 4. 9 Activity Diagram untuk Lihat Jalur

### 7. Activity Diagram Lihat Rute

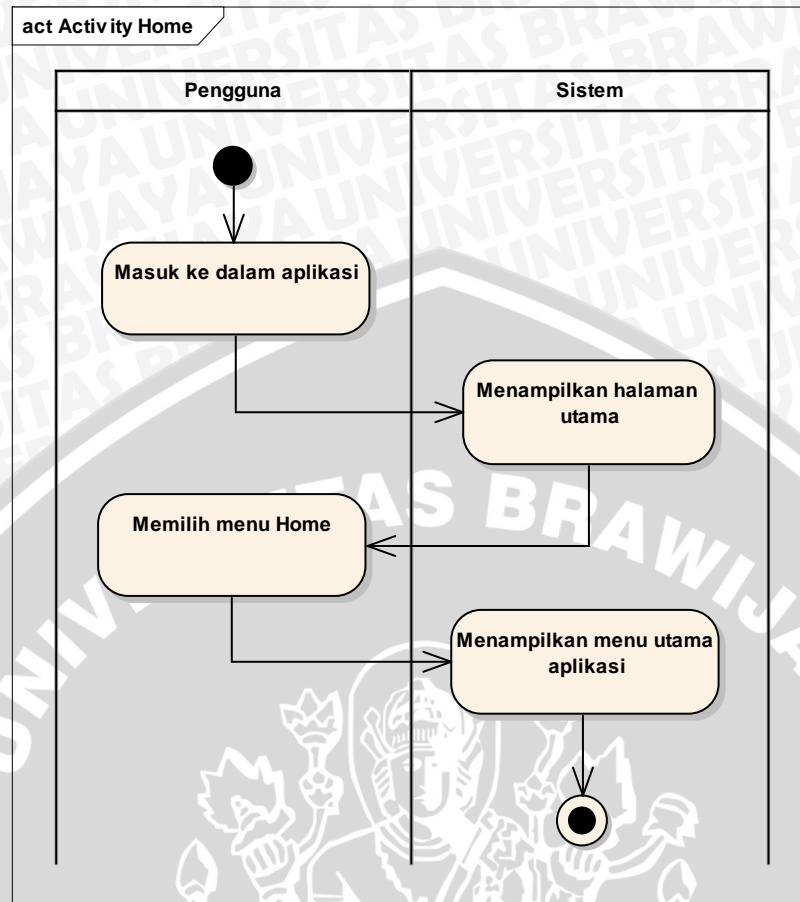
Pada Gambar 4.10 akan ditunjukkan aktivitas yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *use case* lihat rute yang ditunjukkan pada Tabel 4.13. Pengguna masuk aplikasi dan memilih menu ATM terdekat dan daftar lokasi pada halaman utama, kemudian sistem menampilkan daftar ATM. Selanjutnya pengguna memilih salah satu ATM dan pilih menu petunjuk rute.



Gambar 4. 10 Activity Diagram untuk Lihat Rute

8. Activity Diagram Home

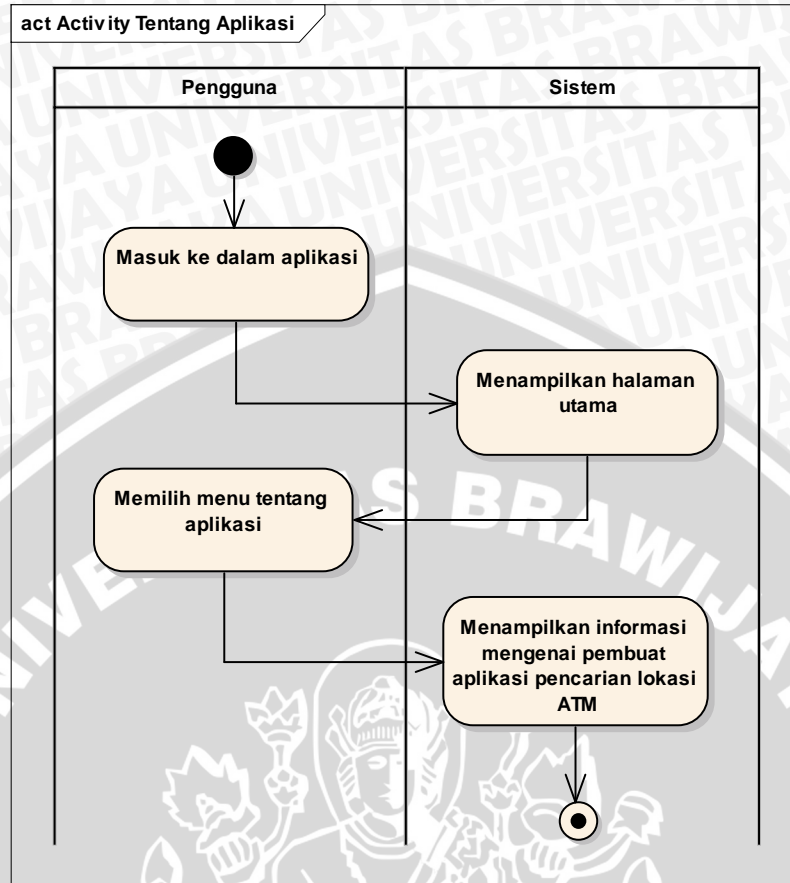
Pada Gambar 4.11 akan ditunjukkan aktivitas yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *use case home* yang ditunjukkan pada Tabel 4.7. Pengguna masuk aplikasi dan memilih menu *home*, selanjutnya sistem menampilkan menu utama aplikasi.



**Gambar 4. 11**Activity Diagram untuk Home

### 9. *Activity Diagram* Tentang Aplikasi

Dalam Gambar 4.12 ditunjukkan aktivitas yang dilakukan oleh pengguna dan sistem. Deskripsi *activity diagram* sesuai dengan skenario pada *use case* tentang aplikasi yang ditunjukkan pada Tabel 4.17. Pengguna masuk aplikasi dan memilih menu tentang aplikasi, selanjutnya sistem menampilkan informasi mengenai pembuat aplikasi.



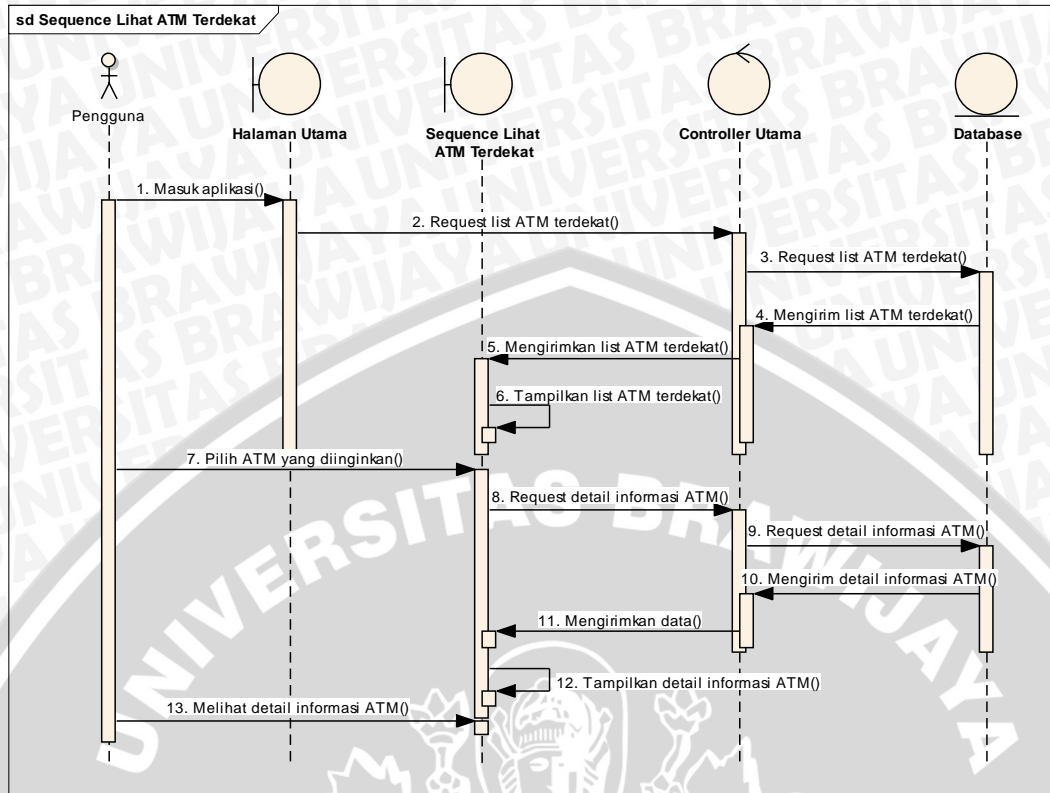
Gambar 4. 12 Activity Diagram Tentang Aplikasi

#### 4.4.1.2 Perancangan Sequence Diagram

Sequence diagram menggambarkan perilaku sistem secara dinamis dan memperlihatkan interaksi dari objek-objek yang terlibat di dalam aplikasi.

##### 1. Diagram Sequence Pengguna: Lihat ATM Terdekat

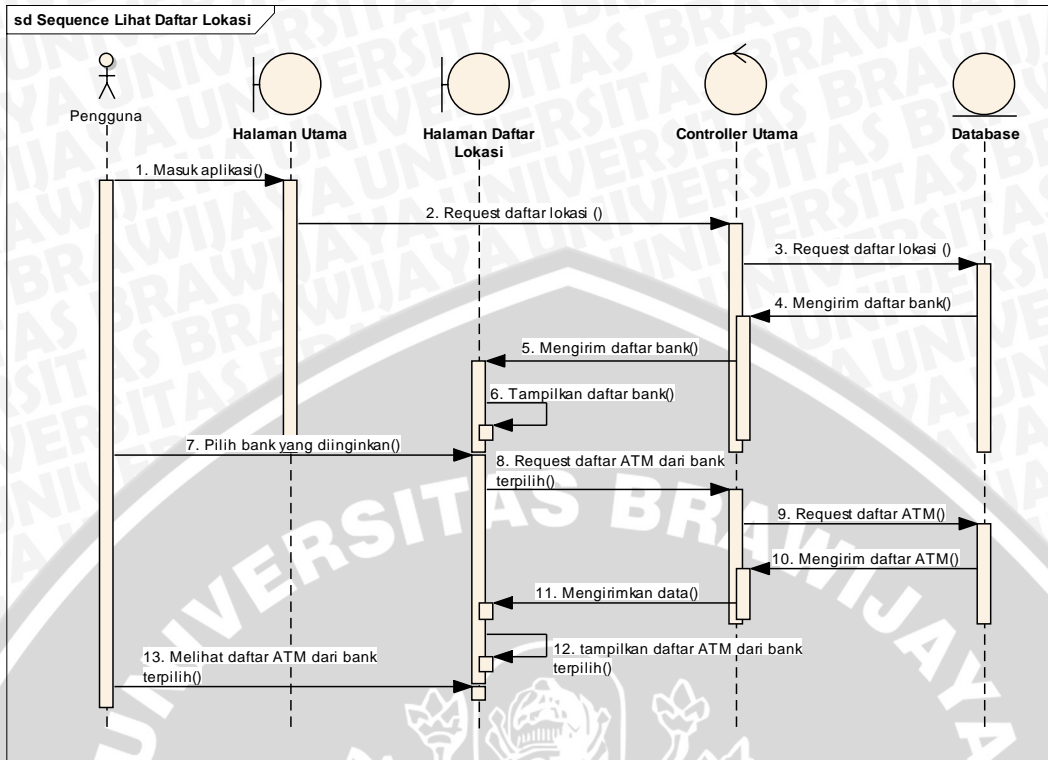
Diagram *sequence* pengguna lihat ATM terdekat menggambarkan perilaku sistem dalam melakukan proses melihat daftar ATM terdekat. *Sequence* dimulai ketika pengguna masuk pada halaman utama aplikasi, *controller* utama akan menampilkan *list* ATM terdekat. Selanjutnya digambarkan alur sistem dalam melakukan proses pilih salah satu ATM dari halaman ATM terdekat.



**Gambar 4. 13** *Sequence Diagram* Lihat ATM Terdekat

## 2. Diagram *Sequence* Pengguna: Lihat Daftar Lokasi

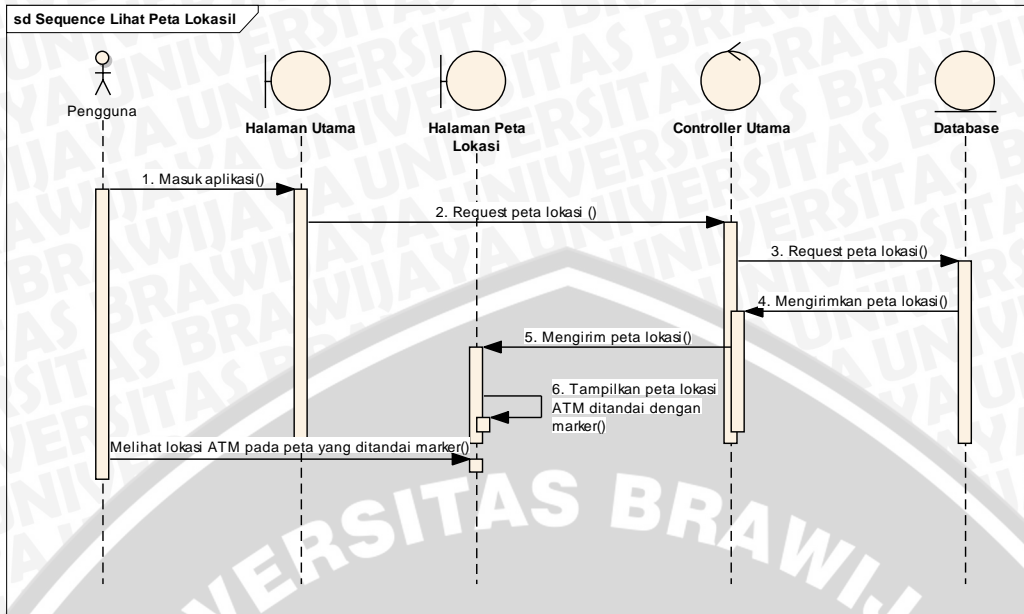
Diagram *sequence* pengguna lihat daftar lokasi menggambarkan perilaku sistem dalam melakukan proses melihat list ATM daftar lokasi. *Sequence* dimulai ketika pengguna masuk pada halaman utama aplikasi, *controller* utama akan menampilkan daftar lokasi ATM. Selanjutnya digambarkan alur sistem dalam melakukan proses pilih salah satu ATM dari halaman daftar lokasi.



Gambar 4. 14 Sequence Diagram Lihat Daftar Lokasi

### 3. Diagram Sequence Pengguna: Lihat Peta Lokasi

Diagram *sequence* pengguna lihat peta lokasi menggambarkan perilaku sistem dalam melakukan proses melihat peta lokasi ATM. *Sequence* dimulai ketika pengguna masuk pada halaman utama aplikasi, *controller* utama akan menampilkan peta lokasi ATM. Selanjutnya digambarkan alur sistem dalam melakukan proses pilih salah satu ATM dari peta lokasi.

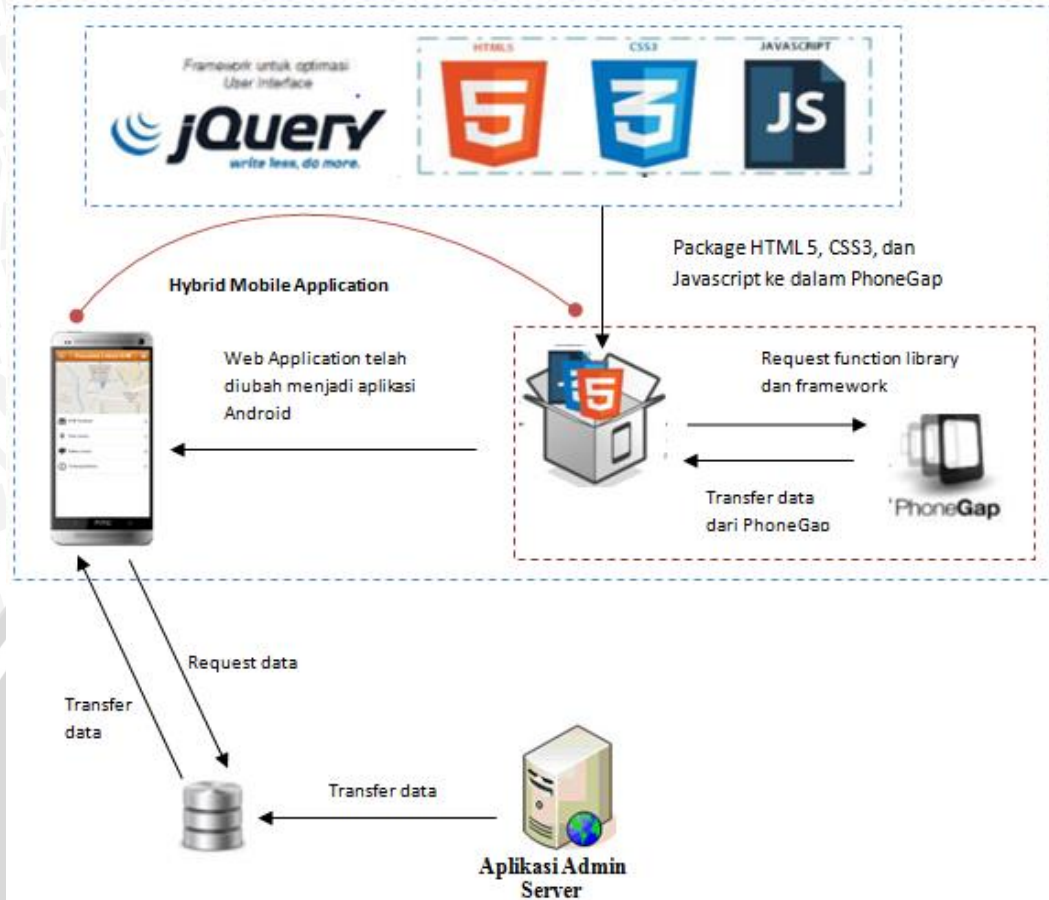


Gambar 4. 15 Sequence Diagram Peta Lokasi

#### 4.4.2 Perancangan Arsitektur Sistem

Perancangan perangkat lunak aplikasi Android *mobile* pencarian lokasi ATM dibangun dengan menggunakan konsep *hybrid mobile application* pada perangkat Android. Pada implementasi konsep *hybrid mobile application* ini digunakan PhoneGap sebagai *framework* yang dapat mengubah aplikasi web menjadi aplikasi seperti *native*, sehingga dapat dijadikan suatu aplikasi berbasis Android. Konsep *hybrid* sendiri dibangun dengan menggunakan bahasa pemrograman HTML 5, CSS3 serta JavaScript dan dimaksimalkan pada *layout interface* menggunakan jQuery. Teknik penyimpanan data menggunakan *online storage* untuk memudahkan dalam *maintenance* apabila sewaktu-waktu diperlukan dalam *update* data. Rancangan arsitektur hubungan sistem akan ditunjukkan dalam Gambar 4.16.

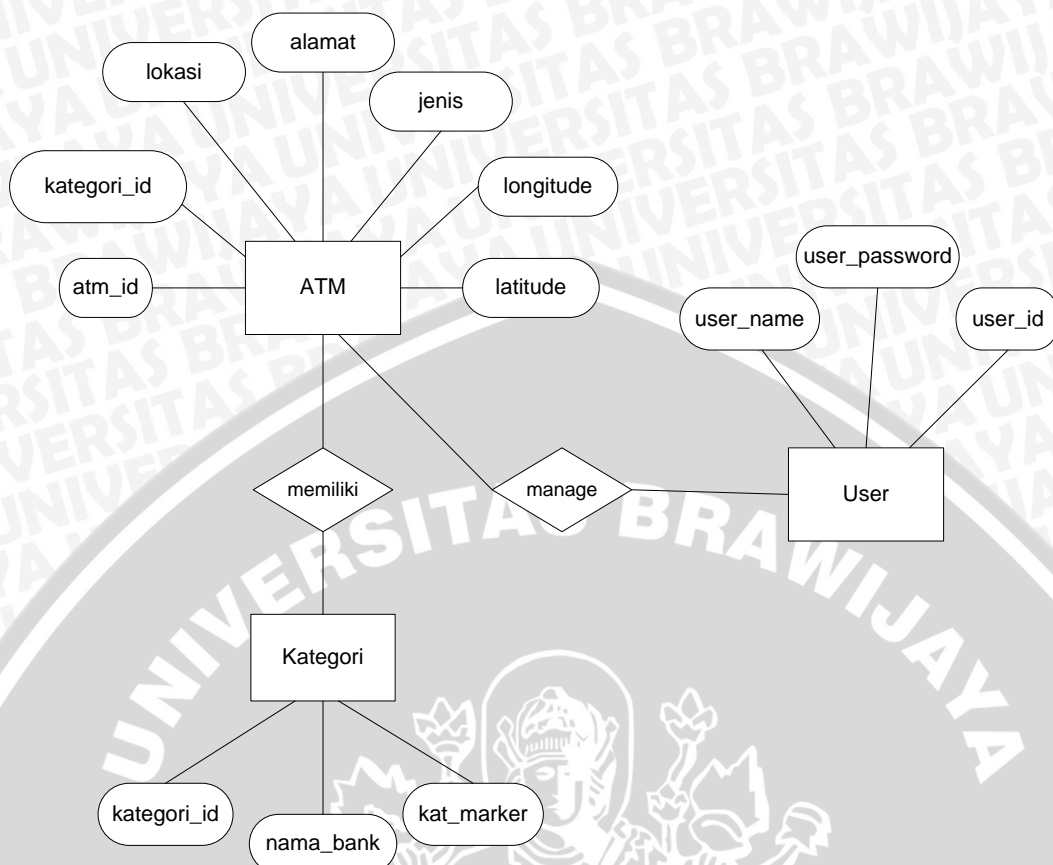




Gambar 4. 16 Perancangan Arsitektur Sistem Aplikasi

#### 4.4.3 Perancangan Basis Data

Basis data berfungsi sebagai tempat menyimpan data. Pada tugas akhir ini perancangan basis data direpresentasikan dalam bentuk ERD (*Entity Relationship Diagram*). ERD menunjukkan hubungan yang terjadi di antara objek (entitas) yang terlibat dalam satu *database*. ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan beberapa atribut yang mempresentasikan seluruh fakta yang ditinjau dari keadaan yang nyata. Pada perancangan basis data sistem ini terdapat tiga buah tabel yaitu tabel atm, tabel kategori bank, dan tabel admin. ERD Sistem ini dapat dilihat dalam Gambar 4.17.



**Gambar 4. 17** Entity Relational Diagram Sistem Pencarian Lokasi ATM Terdekat

Berikut ini merupakan struktur tabel serta keterangan masing-masing tabel dan *field* yang ada pada *database*. Entitas ATM mempresentasikan tabel ATM yang berisi data-data dari ATM yang ada di kota Malang. Struktur tabel ATM ditunjukkan pada Tabel 4.19.

**Tabel 4. 19** Struktur Tabel ATM

No	Nama Field	Type Data	Keterangan
1	Atm_id (PK)	Integer	Kode dari ATM
2	Kategori_id	Tinyint	Kategori dari ATM
3	Lokasi	Text	Lokasi dari ATM
4	Alamat	Text	Alamat dari ATM
5	Keterangan	Text	Jenis dari ATM
6	Latitude	Decimal	Garis lintang
7	Longitude	Decimal	Garis bujur

Entitas kategori bank mempresentasikan tabel kategori yang berisi data-data dari kategori bank. Struktur tabel kategori ditunjukkan pada Tabel 4.20.

**Tabel 4. 20** Struktur Tabel Kategori

No	Nama Field	Tipe Data	Keterangan
1	Kategori_id (PK)	Tinyint	Kode dari kategori bank
2	Nama_bank	Varchar	Nama Abank
4	Kat_marker	Varchar	Kategori marker dari ATM

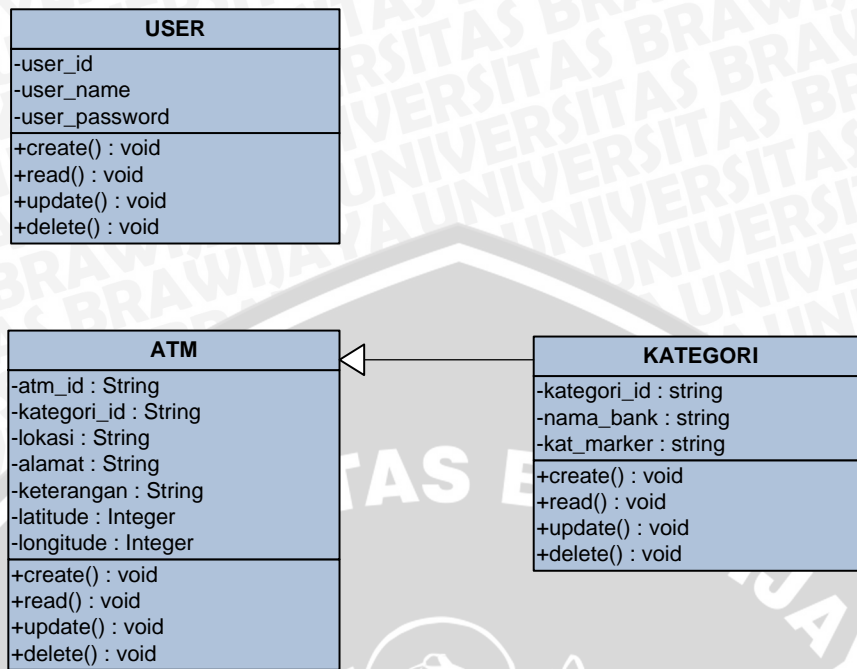
Entitas *user* mempresentasikan tabel *user* yang berisi data-data dari *user*. Struktur tabel admin ditunjukkan pada Tabel 4.21.

**Tabel 4. 21** Struktur Tabel *User*

No	Nama Field	Tipe Data	Keterangan
1	User_id (PK)	Integer	Kode dari user
2	User_name	Varchar	Nama dari user
3	User_password	Varchar	Password dari user

#### 4.4.4 Perancangan *Class Diagram*

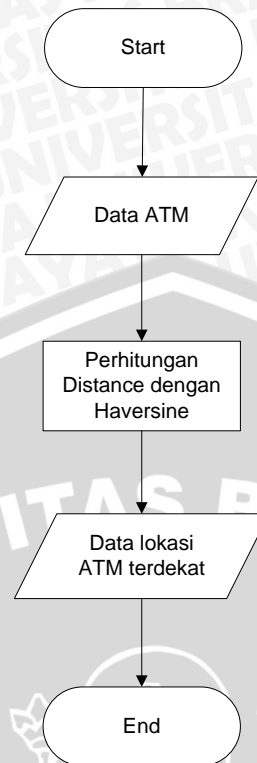
*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain yang membentuk sebuah perangkat lunak. *Class* bisa didapatkan dengan menganalisis secara detail terhadap *use case* yang dimodelkan. Gambar 4.18 menunjukkan diagram *class* dari aplikasi pencarian lokasi ATM terdekat di kota Malang. Pada perancangan proses ini terdapat tiga elemen *class* pada sisi aplikasi admin meliputi *class* ATM, *class* kategori\_bank, dan *class* user.



**Gambar 4. 18** Class Diagram Aplikasi Pencarian Lokasi ATM Terdekat

#### 4.4.5 Perancangan Algoritma Pencarian ATM Terdekat dengan Haversine

Aplikasi pencarian lokasi ATM terdekat di kota Malang menggunakan algoritma Haversine untuk pencarian lokasi ATM terdekat dari posisi pengguna, melalui *latitude* (garis lintang) dan *longitude* (garis bujur). Algoritma Haversine digunakan untuk menghitung jarak antara dua titik GPS. Dalam hal ini adalah titik GPS pengguna dan titik GPS ATM yang sudah tersimpan di dalam *database* sistem, sehingga dapat menjadi kunci utama dalam perbandingan jarak pada penentuan jarak terdekat. Berikut adalah *flow chart* dari pencarian lokasi ATM terdekat menggunakan perhitungan Haversine.



**Gambar 4.19** Flow Chart Pencarian ATM Terdekat Menggunakan Haversine

Adapun langkah-langkah pada proses pencarian ATM terdekat menggunakan algoritma Haversine adalah :

1. Data masukan berupa data lokasi ATM, berupa *latitude* dan *longitude*
2. Dilakukan perhitungan  $d$  (jarak yang dicari) antara pengguna dengan lokasi ATM terpilih berdasarkan rumus pada Persamaan 2.4
3. Hasil akhir berupa daftar ATM terdekat dari posisi pengguna

Berikut adalah perhitungan manual dalam pencarian jarak menggunakan rumus Haversine yang diubah ke dalam sintak SQL, sesuai fungsi pada persamaan... untuk mempermudah perhitungan jarak terdekat antara lokasi pengguna dengan lokasi ATM yang diinginkan, yang berada pada *database* sistem.

Misalkan akan menghitung jarak terdekat antara ATM BCA Kiosk Blimbing dengan posisi pengguna di Universitas Brawijaya, maka :

**Tabel 4. 22** Koordinat Antara UB dengan ATM BCA Kiosk Blimbing

Koordinat	Pengguna (UB)	ATM BCA Kiosk Blimbing
Latitude	-7.955112	-7.932100
Longitude	112.615957	112.647267

**Penerapan dalam Rumus Haversine dengan satuan jarak km :**

$$Distance = (((\cos(\sin((-7.932100 * \pi()/180)) * \sin((-7.955112 * \pi()/180)) + \cos((-7.932100 * \pi()/180)) * \cos((-7.955112 * \pi()/180)) * \cos(((112.647267 - 112.615957) * \pi()/180)))) * 180 / \pi()) * 60 * 1.1515 * 1.609344)$$

$$Distance = 4.293624615 \text{ km}$$

Jarak antara pengguna di Universitas Brawijaya dengan ATM BCA Kiosk

Blimbing dalam satuan km adalah 4.293624615 km

**Penerapan dalam Rumus Haversine dengan satuan jarak mil :**

$$Distance = (((\cos(\sin((-7.932100 * \pi()/180)) * \sin((-7.955112 * \pi()/180)) + \cos((-7.932100 * \pi()/180)) * \cos((-7.955112 * \pi()/180)) * \cos(((112.647267 - 112.615957) * \pi()/180)))) * 180 / \pi()) * 60 * 1.1515)$$

$$Distance = 2.667934646 \text{ mil}$$

Jarak antara pengguna di Universitas Brawijaya dengan ATM BCA Kiosk

Blimbing dalam satuan mil adalah 2.667934646 mil

Kemudian akan menghitung jarak terdekat antara ATM BCA Kiosk Blimbing dengan posisi pengguna yang berbeda, misalkan pengguna berada di Batu Secret Zoo, maka :

**Tabel 4. 23** Koordinat Antara Batu Secret Zoo dengan ATM BCA Kiosk Blimbing

Koordinat	Pengguna (Batu Secret Zoo)	ATM BCA Kiosk Blimbing
Latitude	-7.890807	-7.932100
Longitude	112.532116	112.647267

**Penerapan dalam Rumus Haversine dengan satuan jarak km :**

$$Distance = (((\cos(\sin((-7.932100 * \pi()/180)) * \sin((-7.890807 * \pi()/180)) + \cos((-7.932100 * \pi()/180)) * \cos((-7.890807 * \pi()/180)) * \cos(((112.647267 - 112.532116) * \pi()/180)))) * 180 / \pi()) * 60 * 1.1515 * 1.609344)$$

$$Distance = 13.48727814 \text{ km}$$

Jarak antara pengguna di Batu Secret Zoo dengan ATM BCA Kiosk Blimbing dalam satuan km adalah 13.48727814 km

**Penerapan dalam Rumus Haversine dengan satuan jarak mil :**

$$Distance = (((\cos(\sin((-7.932100 * \pi()/180)) * \sin((-7.890807 * \pi()/180)) + \cos((-7.932100 * \pi()/180)) * \cos((-7.890807 * \pi()/180)) * \cos(((112.647267 - 112.532116) * \pi()/180)))) * 180 / \pi()) * 60 * 1.1515)$$

$$Distance = 8.380606096 \text{ mil}$$

Jarak antara pengguna di Batu Secret Zoo dengan ATM BCA Kiosk Blimbing dalam satuan mil adalah 8.380606096 mil

#### 4.4.6 Perancangan Antarmuka

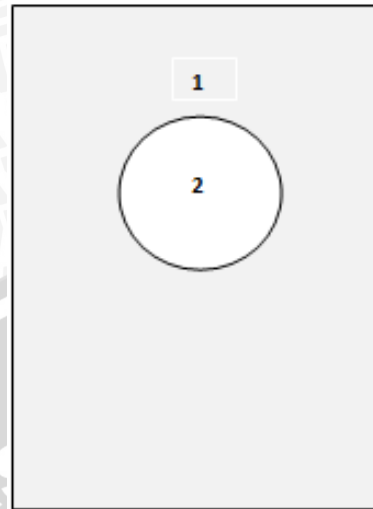
Perancangan antarmuka dilakukan untuk menentukan kondisi akhir dari perangkat lunak yang akan dibangun. Pada perancangan antarmuka ini akan dibahas bagaimana perancangan tampilan.

##### 4.4.6.1 Perancangan Tampilan

Perancangan tampilan digunakan untuk menggambarkan aplikasi yang sebenarnya. Beberapa model tampilan implementasi dalam tuhas akhir ini adalah sebagai berikut :

##### 1. Halaman *Splash Screen*

Halaman *splash screen* merupakan tampilan awal ketika pengguna membuka aplikasi. *Splash screen* memiliki durasi selama beberapa detik atau sampai pengguna menekan layar yang kemudian akan langsung masuk ke halaman Home. *Spalsh screen* pada aplikasi pencarian ATM ini berisi logo aplikasi dan nama aplikasi.



**Gambar 4.20** Antarmuka Halaman *Splash Screen*

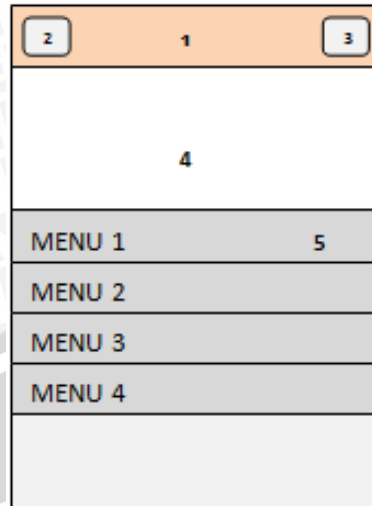
Keterangan :

1. Nama aplikasi
2. Logo aplikasi

## 2. **Halaman Menu Utama (*Home*)**

Halaman home adalah halaman awal saat kita membuka aplikasi setelah keluar tampilan halaman *splash screen*. Pada halaman utama terdiri dari beberapa bagian, yaitu *header* yang berisi logo aplikasi, kemudian beberapa tombol yang berisi menu-menu utama. Menu-menu utama terdiri dari menu ATM Terdekat, menu Daftar Lokasi, menu Peta Lokasi, dan menu Tentang Aplikasi.





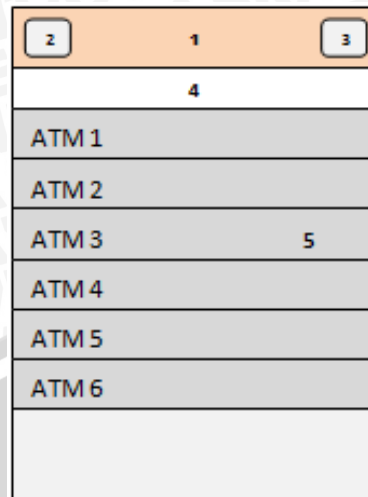
**Gambar 4.21** Antarmuka Halaman Utama (*Home*)

Keterangan :

1. Header aplikasi berisi nama aplikasi
2. *Sidebar* kiri berisi lintasan untuk menu-menu yang ada di halaman utama (*Home*)
3. *Sidebar* kanan berisi pengaturan untuk satuan jarak yang akan digunakan oleh aplikasi, terdapat dua pilihan yaitu dalam satuan mil dan km
4. Menampilkan lokasi pengguna saat memakai aplikasi, yang divisualisasikan dalam bentuk peta
5. Daftar menu-menu utama, diantaranya adalah menu ATM Terdekat, menu Daftar Lokasi, menu Peta Lokasi, dan menu Tentang Aplikasi

### 3. Halaman Menu ATM Terdekat

Halaman menu ATM terdekat akan muncul ketika pengguna memilih menu ATM terdekat pada halaman utama. Di dalam halaman ATM terdekat terdapat kolom yang menampilkan alamat dari posisi pengguna, beberapa nama ATM yang terdekat dengan lokasi pengguna. Gambar 4.20 adalah gambar rancangan antarmuka halaman menu ATM terdekat.

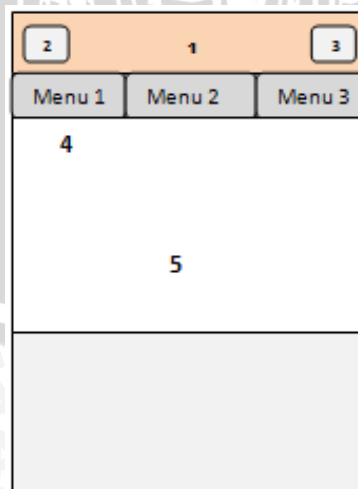


**Gambar 4.22** Antarmuka Halaman ATM Terdekat

Keterangan :

1. *Header* aplikasi, berisi nama aplikasi dan menu
2. *Sidebar* kiri berisi lintasan untuk menu-menu yang ada di *Home*
3. *Sidebar* kanan berisi pengaturan untuk satuan jarak yang akan digunakan oleh aplikasi, terdapat dua pilihan yaitu dalam satuan mil dan km
4. Menampilkan alamat lengkap dari posisi pengguna berada
5. Daftar nama ATM yang terdekat dengan lokasi pengguna

Apabila salah satu tombol untuk daftar nama ATM dipilih, maka akan muncul tampilan antarmuka seperti di bawah ini :



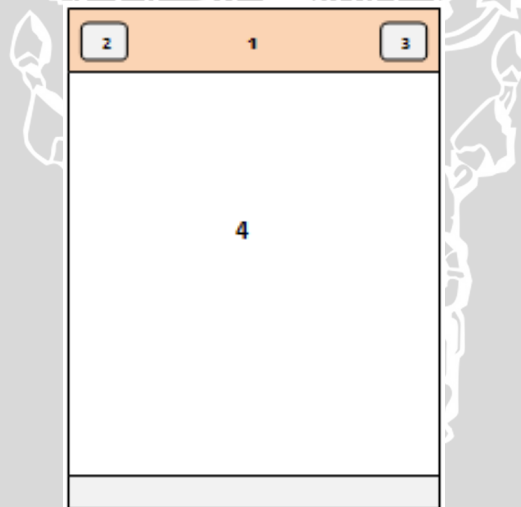
**Gambar 4.23** Antarmuka Halaman Detail ATM dari ATM Terdekat

Keterangan :

1. *Header* aplikasi berisi nama aplikasi
2. *Sidebar* kiri berisi lintasan untuk menu-menu yang ada di *Home*
3. *Sidebar* kanan berisi pengaturan untuk satuan jarak yang akan digunakan oleh aplikasi, terdapat dua pilihan yaitu dalam satuan mil dan km
4. Pilihan navigasi berupa menu peta, petunjuk jalur dan petunjuk *route*
5. Berisi keterangan atau informasi dari ATM yang telah dipilih oleh pengguna. Pada kolom ini akan ditampilkan kategori ATM, nama ATM, alamat ATM dan keterangan ATM (tunai atau non tunai)

#### 4. Halaman Peta Lokasi

Halaman peta lokasi akan muncul ketika pengguna memilih menu peta lokasi pada halaman utama. Di dalam halaman peta lokasi terdapat kolom header dan peta lokasi yang menunjukkan posisi pengguna saat itu. Gambar 4.20 adalah gambar rancangan antarmuka halaman menu peta lokasi.



**Gambar 4.24** Antarmuka Halaman Peta Lokasi

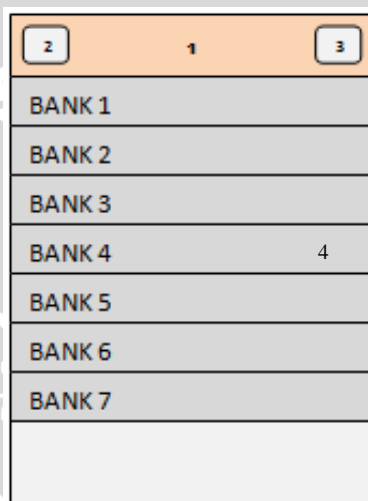
Keterangan:

1. *Header* aplikasi berisi nama aplikasi
2. *Sidebar* kiri berisi lintasan untuk menu-menu yang ada di *Home*
3. *Sidebar* kanan berisi pengaturan untuk satuan jarak yang akan digunakan oleh aplikasi, terdapat dua pilihan yaitu dalam satuan mil dan km

4. Berisi lokasi beberapa ATM yang tersebar di kota Malang, yang divisualisasikan dalam bentuk peta dan ditandai dengan *marker*

#### 5. Halaman Daftar Lokasi

Halaman daftar lokasi akan muncul ketika pengguna memilih menu daftar lokasi pada halaman utama. Di dalam halaman daftar lokasi terdapat *header* aplikasi, beberapa daftar nama bank yang ada di kota Malang. Gambar 4.23 adalah gambar rancangan antarmuka halaman menu daftar lokasi.

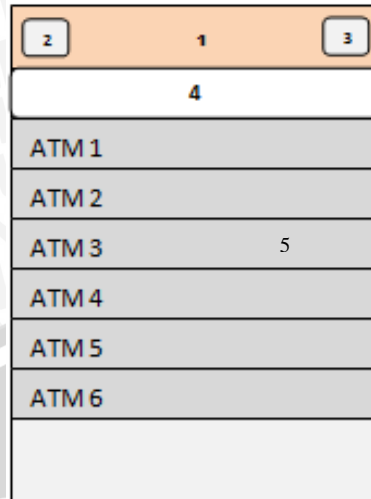


**Gambar 4.25** Antarmuka Halaman Daftar Lokasi

Keterangan :

1. *Header* aplikasi, berisi nama aplikasi
2. *Sidebar* kiri berisi lintasan untuk menu-menu yang ada di *Home*
3. *Sidebar* kanan berisi pengaturan untuk satuan jarak yang akan digunakan oleh aplikasi, terdapat dua pilihan yaitu dalam satuan mil dan km
4. Daftar beberapa nama bank yang ada di kota Malang

Apabila salah satu tombol untuk daftar nama bank dipilih, maka akan muncul tampilan antarmuka seperti di bawah ini :

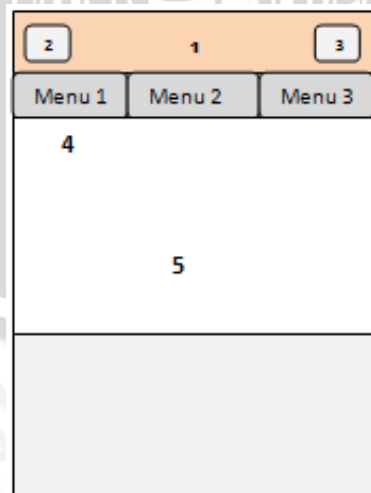


**Gambar 4.26** Antarmuka Halaman Daftar Lokasi ATM Dari Bank yang Terpilih

Keterangan :

1. *Header* aplikasi, berisi nama aplikasi
2. *Sidebar* kiri berisi lintasan untuk menu-menu yang ada di *Home*
3. *Sidebar* kanan berisi pengaturan untuk satuan jarak yang akan digunakan oleh aplikasi, terdapat dua pilihan yaitu dalam satuan mil dan km
4. Berisi fitur *search*
5. Daftar ATM yang ada di kota Malang

Dan apabila salah satu tombol ATM dipilih, maka akan muncul tampilan antarmuka seperti di bawah ini :



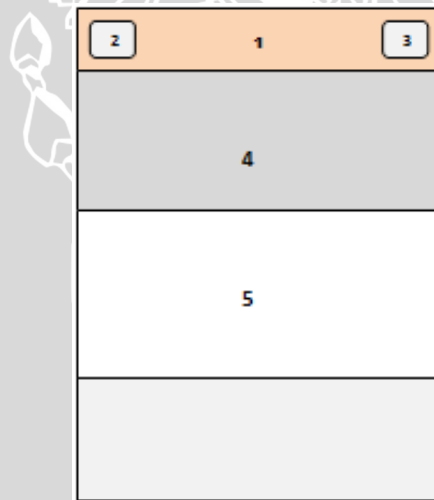
**Gambar 4.27** Antarmuka Halaman Detail ATM dari Daftar Lokasi

Keterangan :

1. *Header* aplikasi berisi nama aplikasi
2. *Sidebar* kiri berisi lintasan untuk menu-menu yang ada di *Home*
3. *Sidebar* kanan berisi pengaturan untuk satuan jarak yang akan digunakan oleh aplikasi, terdapat dua pilihan yaitu dalam satuan mil dan km
4. Pilihan navigasi berupa menu peta, petunjuk jalur dan petunjuk *route*
5. Berisi keterangan atau informasi dari ATM yang telah dipilih oleh pengguna. Pada kolom ini akan ditampilkan kategori ATM, nama ATM, alamat ATM dan keterangan ATM (tunai atau non tunai)

#### 6. Halaman Tentang Aplikasi

Halaman tentang aplikasi akan muncul ketika pengguna memilih menu tentang aplikasi pada halaman utama. Di dalam halaman tentang aplikasi terdapat kolom *header* dan ulasan tentang pembuat aplikasi. Gambar 4.26 adalah gambar rancangan antarmuka halaman menu tentang aplikasi.



**Gambar 4.28** Antarmuka Halaman Tentang Aplikasi

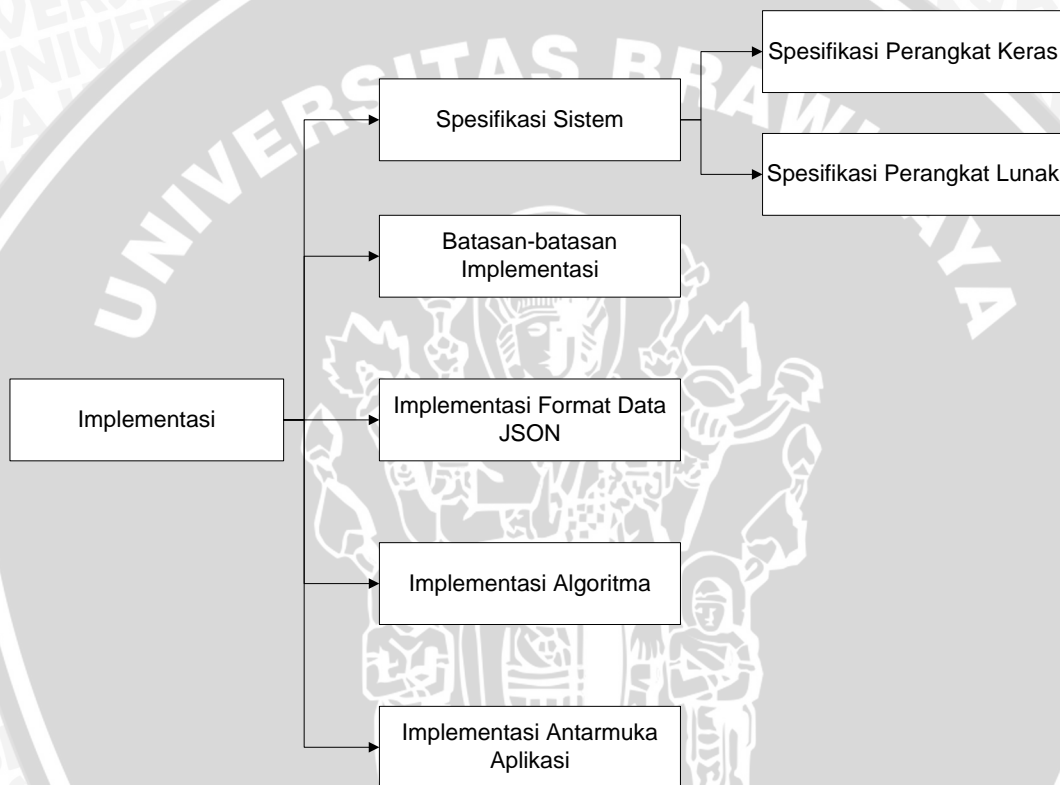
Keterangan:

1. *Header* aplikasi berisi nama aplikasi
2. *Sidebar* kiri berisi lintasan untuk menu-menu yang ada di *Home*
3. *Sidebar* kanan berisi pengaturan untuk satuan jarak yang akan digunakan oleh aplikasi, terdapat dua pilihan yaitu dalam satuan mil dan km
4. Logo dari aplikasi

5. Berisi ulasan tentang pembuat aplikasi

#### 4.5 Implementasi

Pada bab ini akan dibahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan perancangan perangkat lunak. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi format data JSON, implementasi algoritma, dan implementasi antarmuka perangkat lunak.



**Gambar 4.29** Diagram Pohon Implementasi

##### 4.5.1 Spesifikasi Sistem

Hasil dari analisis kebutuhan dan perancangan lunak yang telah dijelaskan pada tahap analisis kebutuhan dan tahap perancangan sistem menjadi dasar untuk dilakukan implementasi menjadi sebuah aplikasi Android yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

#### 4.5.1.1 Spesifikasi Perangkat Keras

Dalam pengembangan aplikasi pencarian lokasi ATM terdekat ini menggunakan sebuah komputer dengan spesifikasi *processor*, *memory*, dan *display* yang akan dijelaskan pada Tabel 4.24 berikut.

**Tabel 4. 24** Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
<i>System Model</i>	Sony VPCCA35FG
<i>Processor</i>	Intel(R) Core(TM) i5-2430M CPU @ 2,40 GHz(4CPUs)
<i>Memory</i>	4096 MB RAM
<i>Display</i>	Intel(R) HD Graphics Family

Dalam melakukan proses instalasi dan pengujian, perangkat yang digunakan adalah perangkat bergerak *smartphone* Android dengan spesifikasi perangkat keras yang ditunjukkan pada Tabel 4.25 berikut.

**Tabel 4. 25** Spesifikasi Perangkat Keras *Smartphone* Android

Nama Komponen	Spesifikasi
<i>System model</i>	Samsung Galaxy Note 8.0 GT –N5100
<i>Processor</i>	1,6 GHz Quad-Core
<i>Memory</i>	16 GB
<i>Display</i>	203,1 mm (8.0'') WXGA 1280x800 Super Clear Type LCD

#### 4.5.1.2 Spesifikasi Perangkat Lunak

Dalam pengembangan aplikasi pencarian lokasi ATM terdekat ini menggunakan sebuah komputer dengan spesifikasi perangkat lunak yang akan dijelaskan pada Tabel 4.26 berikut.

**Tabel 4. 26** Spesifikasi Perangkat Lunak Komputer

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 7 Home Premium 64-bit



Bahasa Pemrograman	HTML, CSS, JavaScript, Java dan MySQL
IDE ( <i>Integrated Development Environment</i> )	Eclipse Juno dengan ADT ( <i>Android Development Tools</i> ) plugin

Dalam melakukan proses instalasi dan pengujian, perangkat yang digunakan adalah perangkat bergerak *smartphone* Android dengan spesifikasi perangkat lunak yang ditunjukkan pada Tabel 4.27 berikut.

**Tabel 4. 27** Spesifikasi Perangkat Lunak *Smartphone* Android

Nama Komponen	Spesifikasi
<i>Platform</i>	Android versi 4.2.2 Android Browser HSDPA 21 Mbps / HSUPA 5,76 Mbps

#### 4.5.2 Batasan Implementasi

Pada implementasi perangkat lunak aplikasi Android *mobile* terdapat batasan-batasan dalam proses yaitu sebagai berikut :

1. Aplikasi pencarian lokasi ATM terdekat di kota Malang, dirancang untuk dijalankan pada *smartphone* Android *mobile* menggunakan konsep *hybrid*.
2. Aplikasi dijalankan dengan menggunakan koneksi internet karena menggunakan konsep *online storage*.
3. Aplikasi menggunakan *framework* PhoneGap dalam pembuatan aplikasi *mobile hybrid*.
4. Untuk penyimpanan data digunakan *online storage* dengan *database* MySQL sebagai penyimpanan data pada sisi *server*.
5. Komunikasi data antara aplikasi dengan *online storage* diimplementasikan dengan menggunakan JSON
6. Pada fitur pencarian lokasi ATM terdekat menggunakan algoritma Haversine untuk menghitung jarak terdekat.
7. Pembuatan *layout* pada *user interface* aplikasi menggunakan bahasa pemrograman HTML, CSS, dan JavaScript.

### 4.5.3 Implementasi Format Data JSON

Pada program aplikasi pencarian lokasi ATM terdapat dua *class* yang digunakan untuk proses pertukaran data, yaitu *class* service yang terletak pada *file* service.php dan *class* service\_model yang terletak pada *file* service\_model.php. Misalkan akan mengakses service get\_list dengan id=1, maka :

$$\underbrace{\text{Atmfinder.url.ph/index.php/service/get\_list?id=1}}_{\text{url admin}}$$

Format url di atas akan memanggil kelas service pada *file* service.php, dengan kode program sebagai berikut.

**Tabel 4. 28** Kode Program Service Get\_List

```

01 public function get_list()
02     {
03         $filter = $this->input->get('q');
04         $category_id = $this->input->get('id');
05
06         $this->load->model('service_model');
07         $query = $this->service_model-
08 >get_list($category_id, $filter);
09
10         header("Access-Control-Allow-Origin: *");
11         header('Access-Control-Allow-Methods: GET,
12 POST');
13         echo json_encode($query);
14     }

```

Kemudian service akan mengirimkan *query* ke function get\_list pada file service\_model dengan kode program sebagai berikut :

**Tabel 4. 29** Kode Program Function Get\_List

```

01 function get_list($category_id, $filter)

```

```

02     {
03         if($category_id != 'undefined'){
04             $this->db->where('kategori',
05 $category_id);
06         }
07         if($filter != 'undefined'){
08             $this->db->like('lokasi', $filter);
09         }
10         $query = $this->db->get('atm');
11         $result = $query->result_array();
12         return $result;
13     }

```

Kemudian hasilnya, *query* akan dikirimkan ke service *get\_list*, dan selanjutnya akan di *encode* sebagai data JSON yang kemudian dikirim ke *client*. Berikut adalah tampilan struktur data Service *Get\_list*

**Tabel 4. 30** Struktur Data *Service Get\_List*

```

01 [{"atm_id":"1","kategori":"1","lokasi":"ATM  BCA  Kiosk
02 Blimbing","alamat":"Jl.  A.  Yani  183  -  Malang  -
03 65125","keterangan":"-","latitude":"-
04 7.932100","longitude":"112.647267"},{"atm_id":"2","kateg
05 ori":"1","lokasi":"ATM  BCA  Kiosk
06 Borobudur","alamat":"Jalan  Candi  Waringin  Lawang,
07 Lowokwaru,  Malang  City,  East  Java,
08 Indonesia","keterangan":"-","latitude":"-
09 7.940569","longitude":"112.638975"},{"atm_id":"3","kateg
10 ori":"1","lokasi":"ATM  BCA  Kiosk
11 Sawojajar","alamat":"Jl.  Danau  Toba  Blok
12 B6","keterangan":"-","latitude":"-
13 7.978870","longitude":"112.660436"},{"atm_id":"4","kateg
14 ori":"1","lokasi":"ATM  BCA  Kiosk
15 Gondanglegi","alamat":"Jl.  Gajah  Mada  36  A  -
16 Gondanglegi","keterangan":"-","latitude":"-
17 7.849778","longitude":"112.524612"},{"atm_id":"5","kateg

```

18	ori": "1", "lokasi": "ATM	BCCA	Kiosk
19	Galunggung", "alamat": "Jl.	Galunggung	No. 78,
20	Malang", "keterangan": "-","latitude": "-		
21	7.968458", "longitude": "112.613379"}]		

Tabel 4.30 di atas menampilkan struktur data *service get\_list* dengan *id=1*, maka data yang akan muncul adalah data ATM dari bank BCA.

#### 4.5.4 Implementasi Algoritma

Aplikasi pencarian lokasi ATM terdekat memiliki beberapa proses atau method yang terdapat pada beberapa class. Pada penulisan tugas akhir ini hanya dicantumkan algoritma dari perhitungan jarak terdekat dari dua titik lokasi menggunakan algoritma Haversine dan algoritma tambah data ATM. Algoritma proses yang dicantumkan adalah pada proses *get ATM nearby* atau pencarian ATM terdekat.

##### 4.5.4.1 Implementasi Algoritma Haversine

Algoritma Haversine dijalankan saat pengguna memilih menu ATM terdekat. Kemudian implementasi algoritma Haversine pada kode program akan menghasilkan jarak terdekat antara lokasi pengguna dengan lokasi ATM. Implementasi algoritma Haversine untuk mendapatkan jarak antara posisi pengguna dengan posisi ATM terdekat akan ditunjukkan pada kode program pada Tabel 4.30 berikut.

**Tabel 4. 31** Implementasi Algoritma pada Menu ATM Terdekat

1	function get_nearby(\$lat, \$long, \$option)
2	{
3	
4	if(\$option == "km")
5	\$this->db->select("*,
6	ROUND(((acos(sin((".\$lat." * pi()/180)) *
7	sin((latitude*pi()/180))+cos((".\$lat." * pi()/180)) *
8	cos((latitude*pi()/180)) * cos((".\$long." - longitude)

```

9  *pi()/180))))*180/pi()*60*1.1515*1.609344) as
10 distance");
11     Else
12         $this->db->select("*,
13 ROUND(((acos(sin((".$lat." * pi()/180)) *
14 sin((latitude*pi()/180))+cos((".$lat." * pi()/180)) *
15 cos((latitude*pi()/180)) * cos((".$long."- longitude)*
16 pi()/180))))*180/pi()*60*1.1515) as distance");
17
18     $this->db->from('atm');
19     $this->db->order_by('distance', 'asc');
20     $query = $this->db->get();
21     $result = $query->result_array();
22
23     return $result;
24 }

```

#### 4.5.4.2 Implementasi Algoritma Tambah Data ATM

Penambahan data ATM dapat dilakukan setelah admin masuk ke halaman admin. Setelah admin dapat masuk ke halaman admin, admin dapat memilih menu *location list* dan memilih *button insert*. Implementasi algoritma untuk menambahkan data ATM dijelaskan pada Gambar 4.30.

NAMA ALGORITMA Menambah ATM

DEKLARASI

\$db;

\$user;

\$data = \$\_POST;

**Masukan :**

1. Memasukkan nilai variabel atm\_id
2. Memasukkan nilai variabel kategori
3. Memasukkan nilai variabel lokasi
4. Memasukkan nilai variabel alamat
5. Memasukkan nilai variabel keterangan
6. Memasukkan nilai variabel latitude

7. Memasukkan nilai variabel longitude

**Proses :**

1. Memanggil method save pada halaman tambah yang berisi:
  - a. Variabel \$data → \$\_POST
  - b. Variabel \$aksi → memanggil method aksi() pada model operasi yang berisi:
    - i. Variabel \$input → query insert data
    - ii. Variabel query → eksekusi query
    - iii. Seleksi kondisi jika kembalian query variable result → query
    - iv. Mengembalikan nilai result
  - c. Jika tidak berhasil menambahkan data maka:
    - i. Menuju halaman tambah ATM
  - d. Menuju halaman tambah ATM

**Keluaran :**

Sistem menampilkan data di halaman Kategori ATM

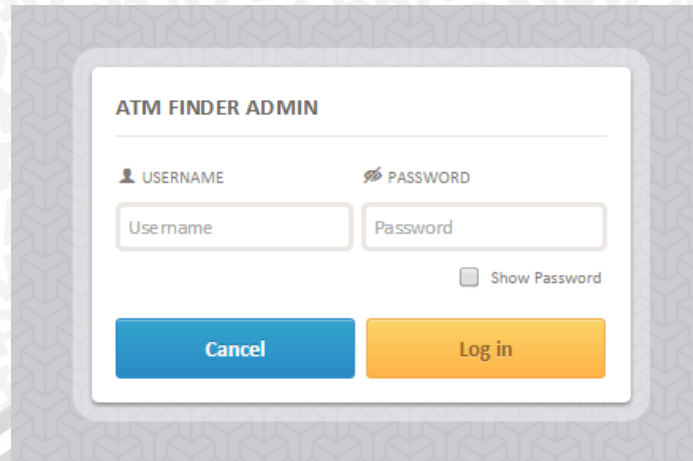
**Gambar 4. 30** Algoritma Menambah Data ATM

#### 4.5.5 Implementasi Antarmuka Aplikasi

Pada implementasi antarmuka aplikasi akan ditampilkan hasil implementasi antarmuka bagian *back-end (server)* dan *front-end* aplikasi *mobile* pencarian lokasi ATM terdekat di kota Malang.

##### 4.5.5.1 Implementasi Back-End (Server)

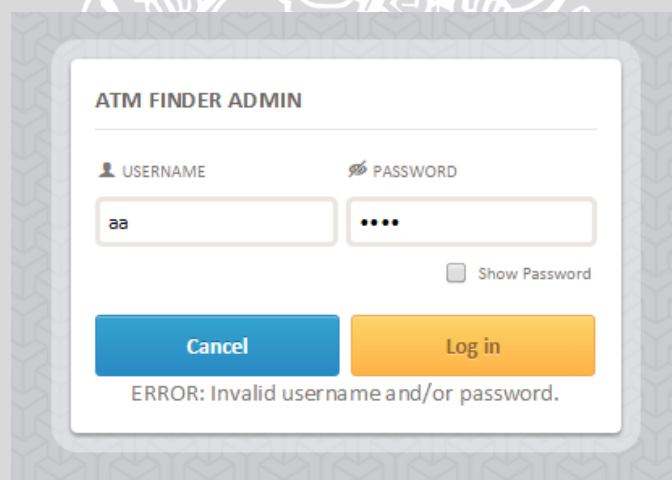
Bagian *back-end* merupakan bagian yang digunakan untuk mengolah *database* yang nantinya akan digunakan sistem. Bagian ini dikelola oleh admin sebagai *user* tunggalnya. Agar admin memiliki akses, maka admin perlu melakukan proses *login* terlebih dahulu. Berikut adalah tampilan *login* dari halaman admin :



The image shows a login form titled "ATM FINDER ADMIN". It has two input fields: "USERNAME" with a user icon and "PASSWORD" with a key icon. The username field contains the text "Use name" and the password field contains "Password". There is a "Show Password" checkbox below the password field. At the bottom, there are two buttons: a blue "Cancel" button and an orange "Log in" button.

**Gambar 4.31** Halaman Login Admin

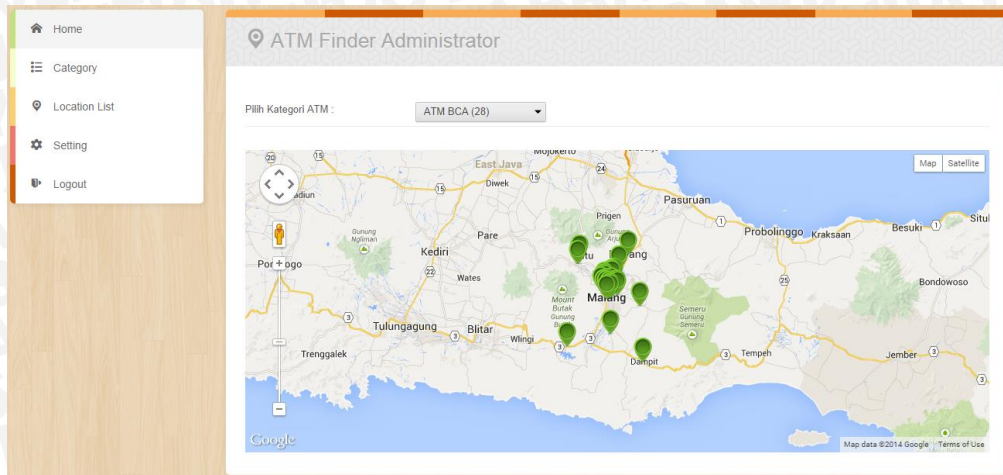
Jika terjadi kesalahan *input* pada data *username* dan *password*, maka sistem akan merespons dengan menampilkan konfirmasi bahwa *username* dan *password* salah. Gambar 4.30 akan menggambarkan pesan *error* saat memasukkan *username* dan *password* yang salah.



The image shows the same login form as in Gambar 4.31, but with an error message displayed at the bottom: "ERROR: Invalid username and/or password." The username field now contains "aa" and the password field contains four dots. The "Show Password" checkbox is still present. The "Cancel" and "Log in" buttons are also visible.

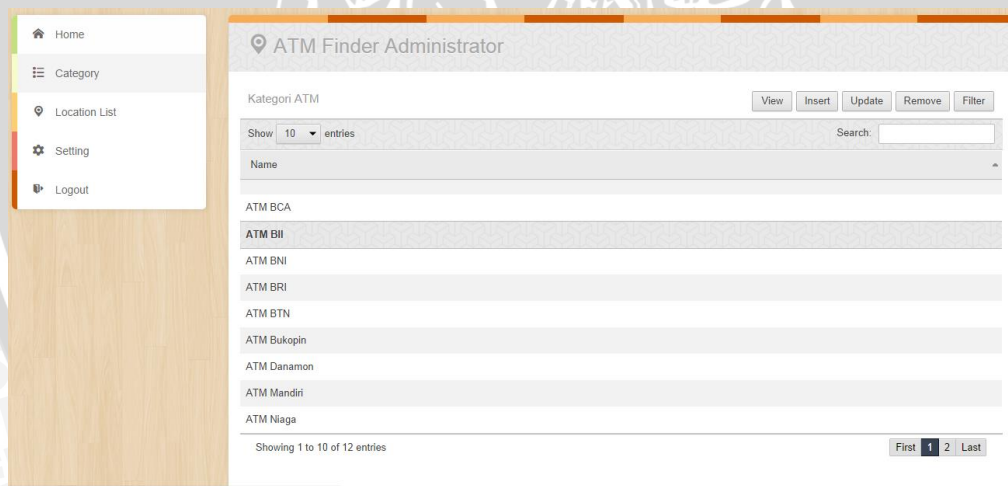
**Gambar 4.32** Halaman Login Admin Bila Terjadi Kesalahan

Setelah admin memasukkan *username* dan *password* yang benar, selanjutnya admin akan dihadapkan pada halaman *home*. Berikut adalah tampilan halaman *home*.



**Gambar 4.33** Halaman *Home Admin*

Pada halaman admin terdapat dua menu utama yang digunakan untuk melakukan pengolahan data ATM, yaitu menu *category* dan menu *location list*. Apabila admin memilih menu *category*, maka akan ditampilkan gambar seperti di bawah ini.



**Gambar 4.34** Halaman Admin Menu *Category*

Pada halaman menu *category*, akan ditampilkan daftar kategori ATM dari bank yang ada di kota Malang. Pada menu ini admin dapat melakukan pengolahan data berupa *view*, *insert*, *update*, *remove*, dan *filter*. Selanjutnya apabila admin memilih menu *location list*, maka akan ditampilkan gambar seperti di bawah ini.



Name	Category	Keterangan	Address
Alfamidi Bandulan	ATM Bil	-	Jalan Bandulan No.12, Sukun, Malang City, East Java 65146, Indonesia
ATM Bank Permata Carrefour Malang	ATM Permata Bank	-	Jl. A. Yani 2A, East Java 65163
ATM Bank Permata Plaza Matos	ATM Permata Bank	-	Plaza Matos Kav GS 5 No. 8 Jl Veteran No.2, East Java 65145
ATM BCA Indomaret Esberg	ATM BCA	-	Jl. Esberg 8A, karang Widoro Malang 65151
ATM BCA Kiosk Basuki Rachmat 2	ATM BCA	-	Jl. Jend. Basuki Rachmat 70-74 - Malang- 65111
ATM BCA Kiosk Batu 1	ATM BCA	-	Jl. Panglima Sudirman 1 - Batu - 65311
ATM BCA Kiosk Blimbing	ATM BCA	-	Jl. A. Yani 183 - Malang - 65125
ATM BCA Kiosk Borobudur	ATM BCA	-	Jalan Candi Waringin Lawang, Lowokwaru, Malang City, East Java, Indonesia
ATM BCA Kiosk Dampit	ATM BCA	-	Jl. Semeru Selatan 32 - Dampit - 65181
ATM BCA Kiosk Dlang Plaza	ATM BCA	-	Jl. Raya Langsep No. 2

**Gambar 4.35** Halaman Admin Menu *Location List*

Pada halaman menu *location list*, akan ditampilkan daftar ATM yang ada di kota Malang. Pada menu ini admin juga dapat melakukan pengolahan data berupa *view*, *insert*, *update*, *remove*, dan *filter*. Apabila admin melakukan proses *update* terhadap salah satu ATM, maka akan ditampilkan gambar seperti di bawah ini.

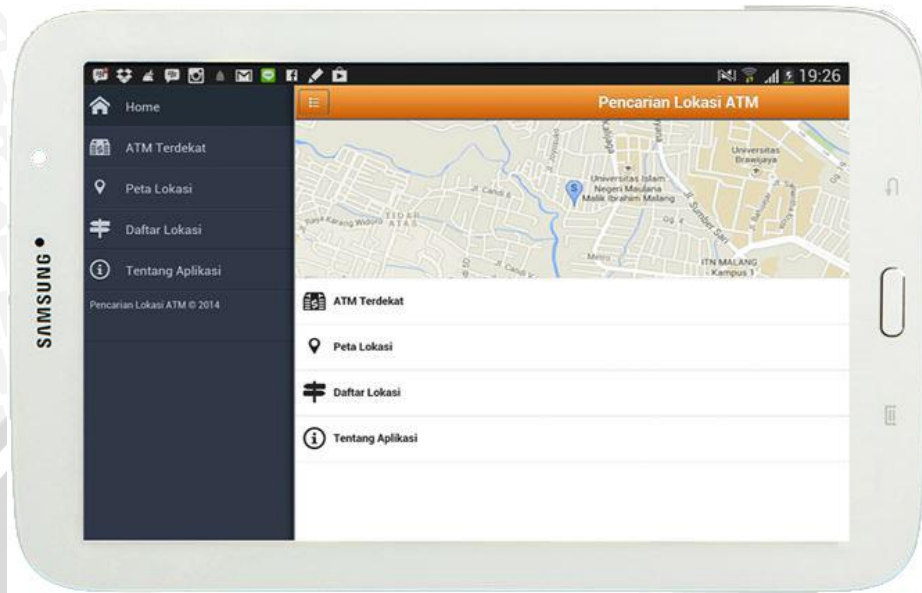
**Gambar 4.36** Halaman *Update Data ATM* pada Menu *Location List*

#### 4.5.5.2 Implementasi *Front-End*

Bagian *front-end* adalah bagian aplikasi yang dijalankan oleh pengguna. Untuk dapat mengaksesnya pengguna tidak perlu melakukan proses *login*, dan cukup meng-*install* aplikasinya pada *smartphone* Android *mobile*. Berikut adalah implementasi dari bagian *front-end*.

##### 1. Halaman *Home*

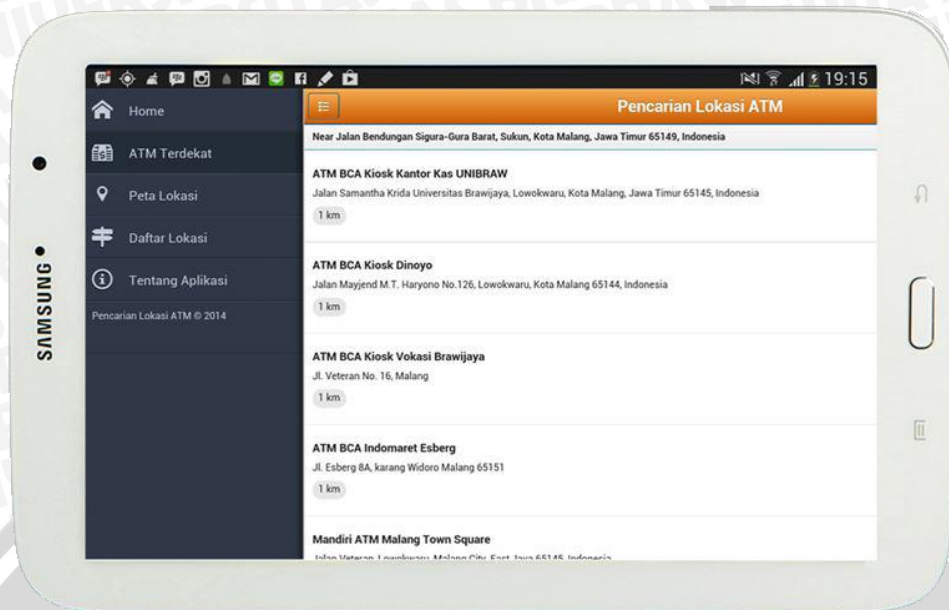
Halaman *home* ditampilkan pertama kali saat pengguna membuka aplikasi.  
Halaman *home* berisi daftar menu-menu utama pada aplikasi.



**Gambar 4.37** Tampilan Halaman *Home*

## 2. Halaman ATM Terdekat

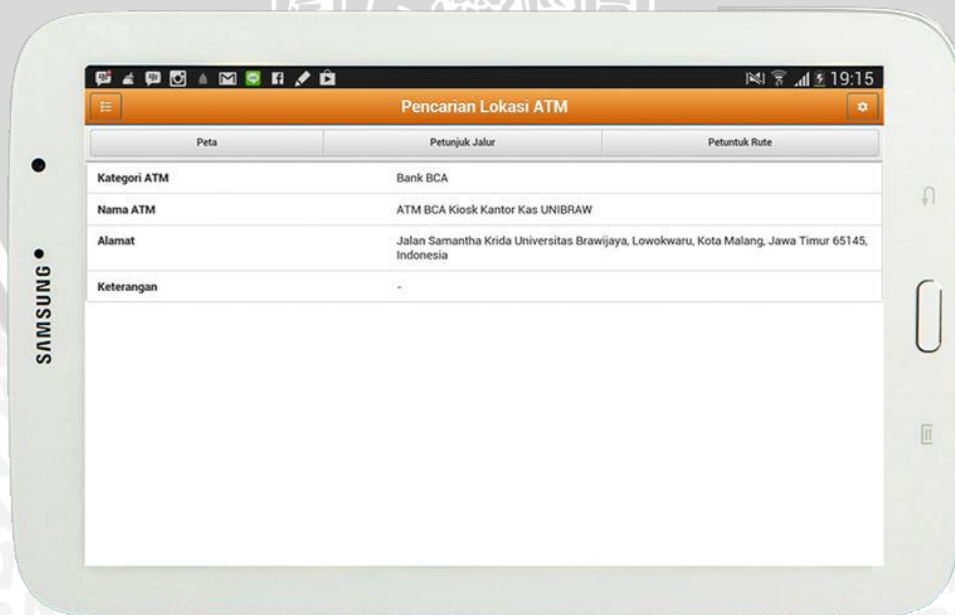
Halaman ATM terdekat ditampilkan saat pengguna memilih menu ATM Terdekat pada halaman *home*. Pada halaman ini ditampilkan daftar ATM yang terdekat dari lokasi pengguna.



Gambar 4. 38 Tampilan Halaman ATM Terdekat

### 3. Halaman Detail ATM

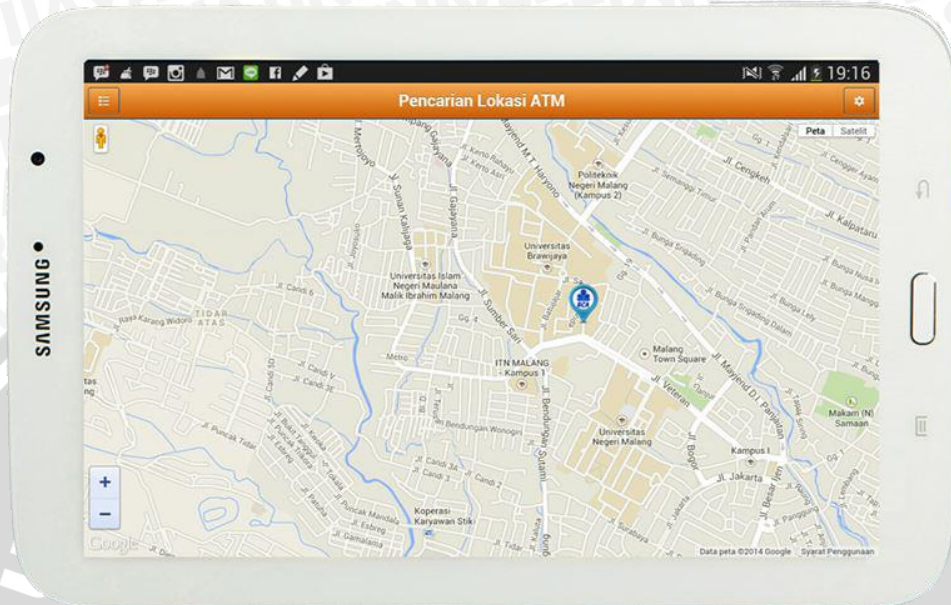
Halaman detail ATM ditampilkan saat pengguna memilih menu ATM terdekat, kemudian aplikasi menampilkan daftar ATM terdekat dan pengguna memilih ATM yang diinginkan. Maka akan muncul halaman detail ATM.



Gambar 4.39 Tampilan Halaman Detail ATM

#### 4. Halaman Lihat Peta

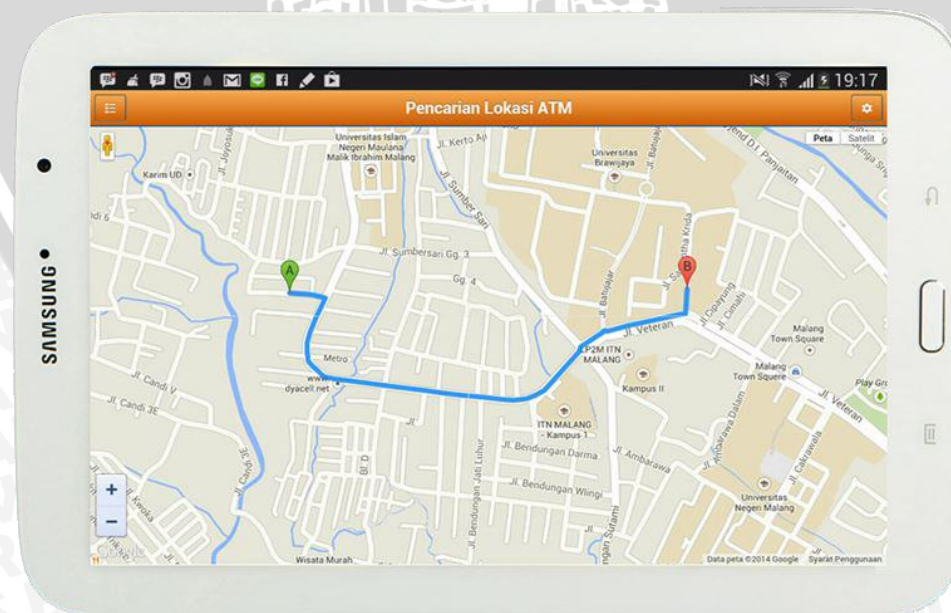
Halaman lihat peta ditampilkan saat aplikasi menampilkan halaman detail ATM dan pengguna memilih menu peta.



Gambar 4.40 Tampilan Halaman Lihat Peta

#### 5. Halaman Lihat Jalur

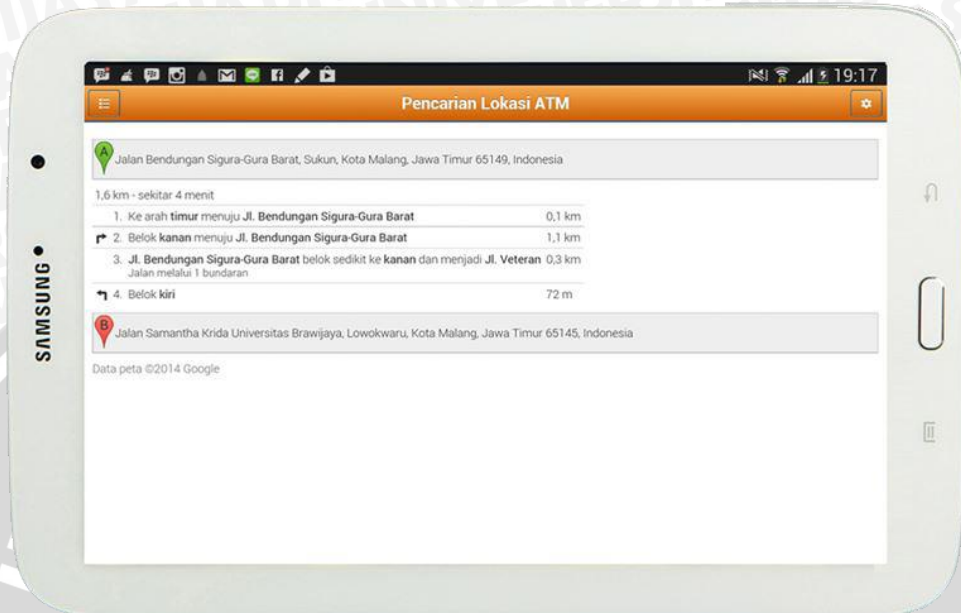
Halaman lihat jalur ditampilkan saat aplikasi menampilkan halaman detail ATM dan pengguna memilih menu jalur.



Gambar 4.41 Tampilan Halaman Lihat Jalur

## 6. Halaman Lihat Rute

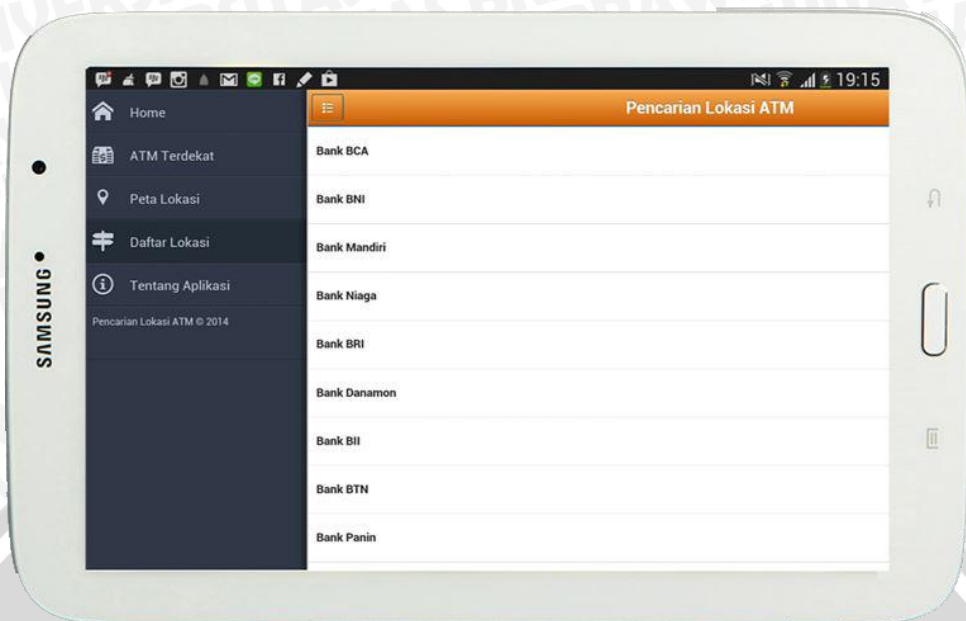
Halaman lihat rute ditampilkan saat aplikasi menampilkan halaman detail ATM dan pengguna memilih menu rute.



**Gambar 4.42** Tampilan Halaman Lihat Rute

## 7. Halaman Daftar Lokasi

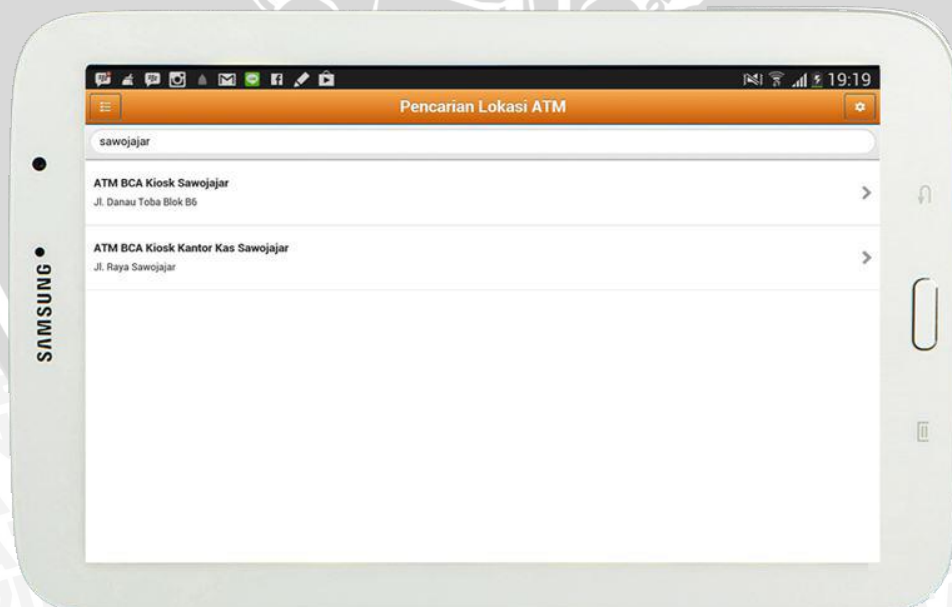
Halaman daftar lokasi ditampilkan saat pengguna memilih menu Daftar Lokasi pada halaman *home*. Pada halaman ini ditampilkan daftar bank-bank yang ada di kota Malang, disertai jumlah ATM dari bank tersebut, yang tersebar di area kota Malang.



**Gambar 4.43** Tampilan Halaman Daftar Lokasi

## 8. Halaman *Search* ATM

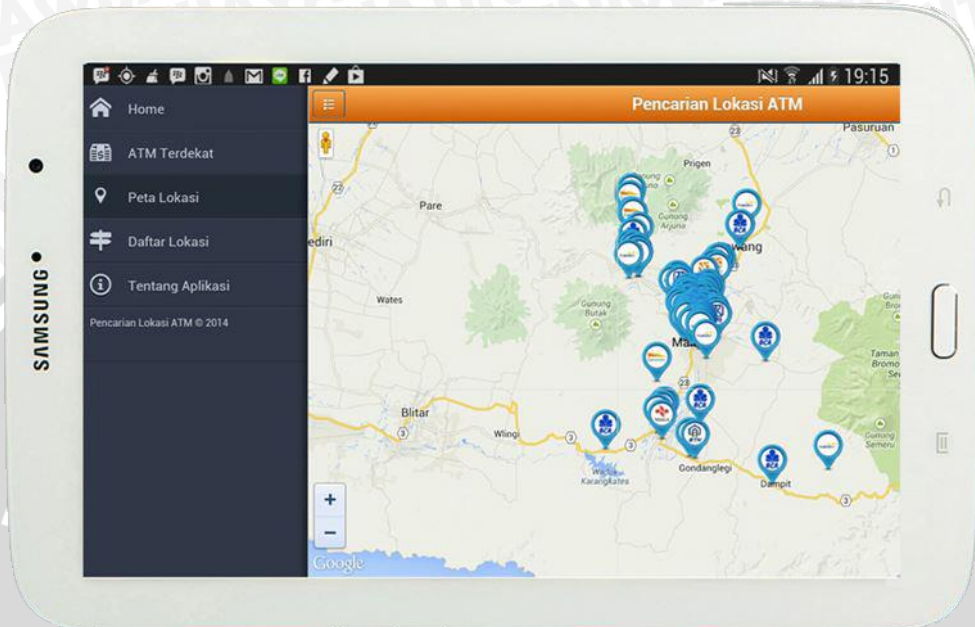
Halaman *search* ATM ditampilkan saat pengguna memilih menu daftar lokasi dan aplikasi menampilkan daftar keseluruhan ATM dari bank terpilih. Kemudian pengguna dapat *search* ATM yang diinginkan pada kolom *search*.



**Gambar 4.44** Tampilan Halaman *Search* ATM

## 9. Halaman Peta Lokasi

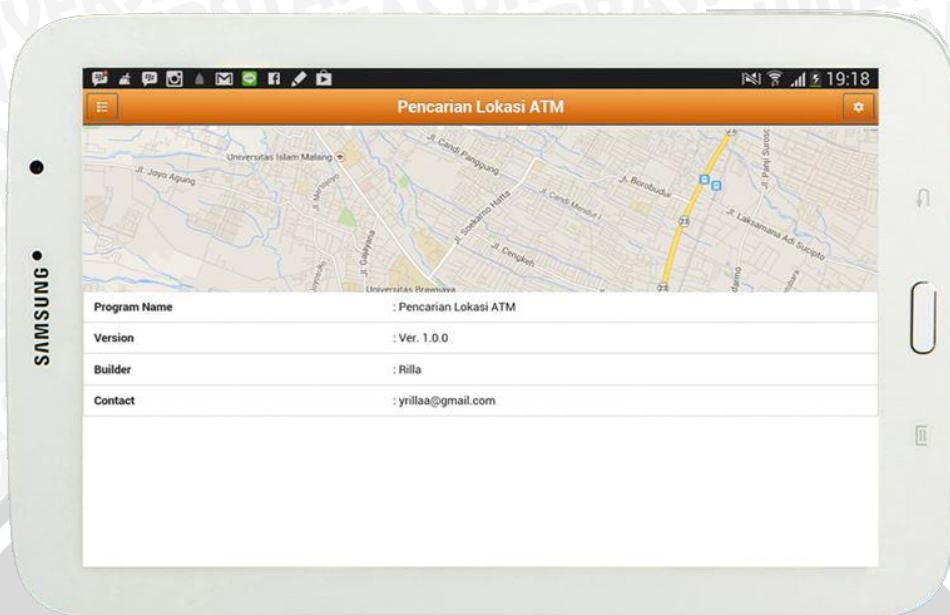
Halaman peta lokasi ditampilkan saat pengguna memilih menu Peta Lokasi pada halaman utama (*home*). Pada halaman ini ditampilkan visualisasi lokasi ATM yang tersebar di area kota Malang dalam bentuk peta dan ditandai dengan *marker*.



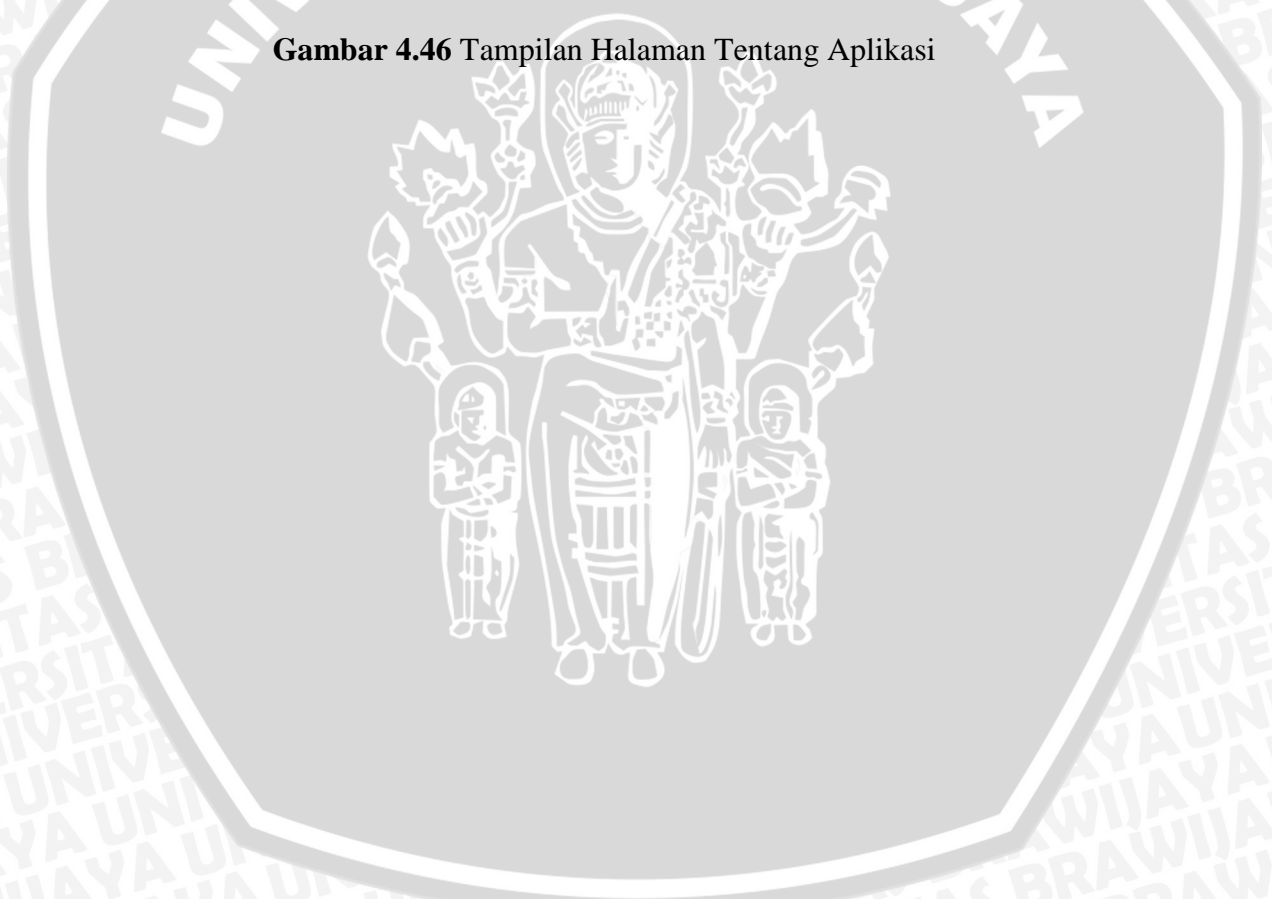
Gambar 4.45 Tampilan Halaman Peta Lokasi

## 10. Halaman Tentang Aplikasi

Halaman tentang aplikasi ditampilkan saat pengguna memilih menu Tentang Aplikasi pada halaman utama (*home*). Pada halaman ini ditampilkan informasi mengenai pembuat aplikasi



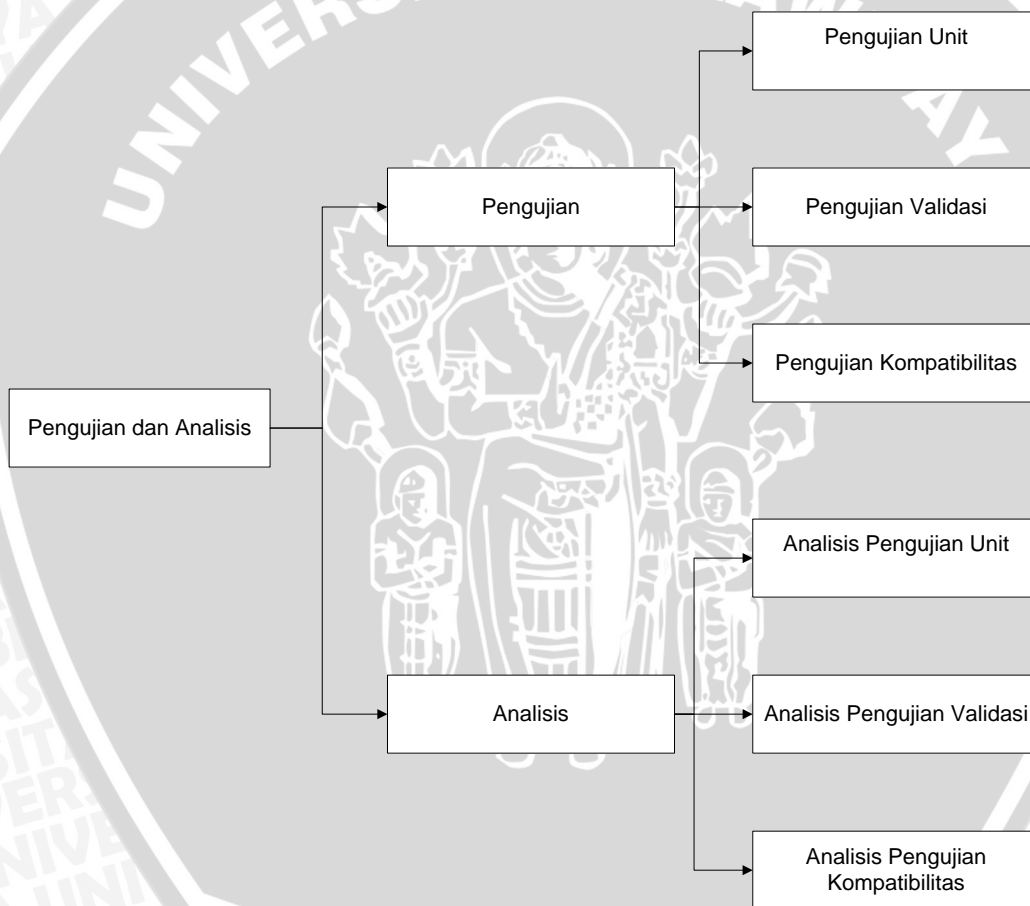
**Gambar 4.46** Tampilan Halaman Tentang Aplikasi





## BAB V PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis terhadap aplikasi mobile yang telah dibangun. Proses pengujian dilakukan melalui tiga tahapan (strategi) yaitu pengujian unit, pengujian validasi dan pengujian kompatibilitas. Pada pengujian unit digunakan teknik pengujian *White-Box (White-Box Testing)* Pada pengujian validasi dan kompatibilitas digunakan teknik pengujian *Black-Box (Black-Box Testing)*.



**Gambar 5.1** Diagram Pohon Pengujian dan Analisis

### 5.1 Pengujian

Proses pengujian dilakukan melalui tiga tahapan (strategi) yaitu pengujian unit, pengujian validasi, dan pengujian kompatibilitas.

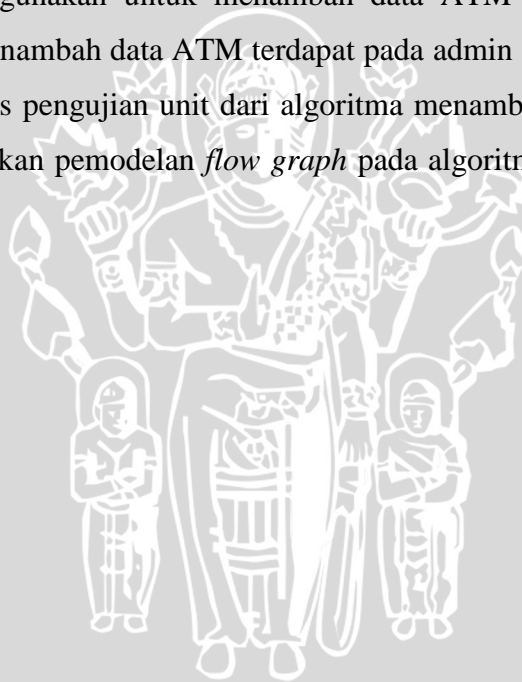


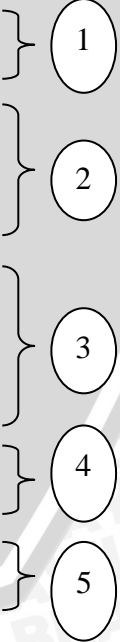
### 5.1.1 Pengujian Unit

Perangkat lunak yang dikembangkan dengan paradigma *object-oriented programming* menerapkan pengujian unit untuk suatu *method* (operasi) dari suatu *class*. Pada pengujian unit aplikasi pencarian lokasi ATM terdekat di kota Malang digunakan teknik *White-Box Testing* dengan teknik *Basis Path Testing*. Pada teknik *Basis Path Testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*, menentukan jumlah kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan.

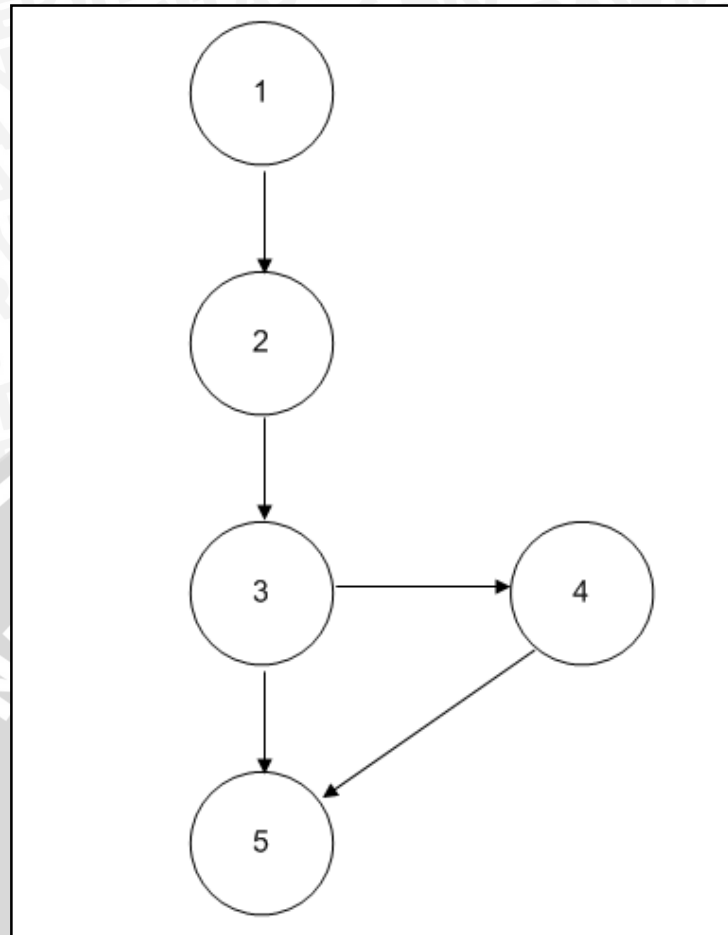
#### 5.1.1.1 Pengujian Unit Algoritma Menambah Data ATM

Fungsi ini digunakan untuk menambah data ATM yang ada di kota Malang. Algoritma menambah data ATM terdapat pada admin (web). Gambar 5.2 menggambarkan proses pengujian unit dari algoritma menambah data ATM, dan Gambar 5.3 menunjukkan pemodelan *flow graph* pada algoritma menambah data ATM.



<p>NAMA ALGORITMA Menambah ATM</p> <p>DEKLARASI</p> <pre>\$db; \$user; \$data = \$_POST;</pre> <p><b>Masukan:</b></p> <ol style="list-style-type: none"> <li>8. Memasukkan nilai variabel atm_id</li> <li>9. Memasukkan nilai variabel kategori</li> <li>10. Memasukkan nilai variabel lokasi</li> <li>11. Memasukkan nilai variabel alamat</li> <li>12. Memasukkan nilai variabel keterangan</li> <li>13. Memasukkan nilai variabel latitude</li> <li>14. Memasukkan nilai variabel longitude</li> </ol> <p><b>Proses:</b></p> <ol style="list-style-type: none"> <li>2. Memanggil method save pada halaman tambah yang berisi:             <ol style="list-style-type: none"> <li>e. Variabel \$data → \$_POST</li> <li>f. Variabel \$aksi → memanggil method aksi() pada model operasi yang berisi:                     <ol style="list-style-type: none"> <li>v. Variabel \$input → query insert data</li> <li>vi. Variabel query → eksekusi query</li> <li>vii. Seleksi kondisi jika kembalian query variable result → query</li> <li>viii. Mengembalikan nilai result</li> </ol> </li> <li>g. Jika tidak berhasil menambahkan data maka:                     <ol style="list-style-type: none"> <li>i. Menuju halaman tambah ATM</li> <li>h. Menuju halaman tambah ATM</li> </ol> </li> </ol> </li> </ol> <p><b>Keluaran:</b></p> <p>Sistem menampilkan data di halaman Kategori ATM</p>	
--	---

**Gambar 5.2** Pengujian Unit Algoritma Menambah Data ATM



**Gambar 5.3** Pemodelan *Flow Graph* Algoritma Menambah Data ATM

Pemodelan ke dalam flow graph yang telah dilakukan terhadap algoritma menambah data ATM menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan  $V(G) = E - N + 2$ , dimana  $V(G)$  merupakan jumlah kompleksitas siklomatis,  $E$  merupakan sisi atau *edge* (garis penghubung antar node) dan  $N$  merupakan jumlah simpul (node).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 5 - 5 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan lima buah basis set dari jalur independent yang dijelaskan pada Tabel 5.1, yaitu:

$$\text{Jalur 1} = 1 - 2 - 4 - 5$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 5$$

**Tabel 5. 1** Kasus Uji Algoritma Menambah ATM

No.	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1.	Dapat memasukkan data	Data dapat ditampilkan pada sistem	Data dapat ditampilkan pada sistem
2.	Tidak dapat memasukkan data	Data tidak dapat ditampilkan pada sistem	Data tidak dapat ditampilkan pada sistem

### 5.1.2 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Daftar kebutuhan yang telah dirumuskan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Black-Box*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan. Pada tugas akhir ini dilakukan pengujian validasi terhadap aplikasi Android *mobile* pencarian lokasi ATM terdekat di kota Malang. Untuk mengetahui kesesuaian antara daftar kebutuhan dengan kinerja sistem, pada setiap kebutuhan sistem dilakukan proses pengujian.

#### 5.1.2.1 Kasus Uji Validasi Tampilan Admin

Proses pengujian tampilan admin dilakukan dengan kasus uji masing-masing yang akan ditunjukkan oleh tabel-tabel di bawah ini.

**Tabel 5. 2** Kasus Uji Validasi *Login* (data normal)

<b>Nomor Kasus Uji</b>	VAL_01
<b>Nama Kasus Uji</b>	Kasus uji admin <i>login</i>
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_001_01)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan admin dapat

	melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> yang valid
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka tampilan admin</li> <li>2. Masukkan <i>username</i> valid</li> <li>3. Masukkan <i>password</i> valid</li> <li>4. Klik <i>login</i></li> <li>5. Sistem menampilkan halaman utama admin</li> </ol>
<b>Hasil Yang Diharapkan</b>	Admin dapat masuk ke sistem

Tabel 5.3 Kasus Uji Validasi *Login* (data salah)

<b>Nomor Kasus Uji</b>	VAL_02
<b>Nama Kasus Uji</b>	Kasus uji <i>login</i> (data salah)
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_001_01)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk verifikasi <i>username</i> dan <i>password</i> yang salah atau kosong
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka tampilan admin</li> <li>2. Masukkan <i>username</i> invalid atau kosong</li> <li>3. Masukkan <i>password</i> invalid atau kosong</li> <li>4. Klik <i>login</i></li> <li>5. Pesan <i>error username</i> dan <i>password</i> yang dimasukkan salah</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pada sistem muncul pesan peringatan bahwa <i>username</i> dan <i>password</i> yang dimasukkan oleh admin tidak cocok, sehingga admin tidak dapat masuk ke sistem

Tabel 5.4 Kasus Uji Validasi *Input Data ATM*

<b>Nomor Kasus Uji</b>	VAL_03
<b>Nama Kasus Uji</b>	Kasus <i>input data ATM</i>
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_001_02)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam

	melakukan <i>input</i> data ATM
<b>Prosedur Uji</b>	Pengisian data
<b>Hasil Yang Diharapkan</b>	Aplikasi dapat menyimpan data ATM yang telah diinputkan ke dalam sistem

**Tabel 5. 5** Kasus Uji Validasi Ubah Data ATM

<b>Nomor Kasus Uji</b>	<b>VAL_04</b>
<b>Nama Kasus Uji</b>	Kasus ubah data ATM
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_001_03)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam melakukan ubah data ATM
<b>Prosedur Uji</b>	Pengubahan data
<b>Hasil Yang Diharapkan</b>	Aplikasi dapat mengubah data ATM yang sudah ada

**Tabel 5. 6** Kasus Uji Validasi Hapus Data ATM

<b>Nomor Kasus Uji</b>	<b>VAL_05</b>
<b>Nama Kasus Uji</b>	Kasus hapus data ATM
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_001_04)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam melakukan hapus data ATM
<b>Prosedur Uji</b>	Penghapusan data
<b>Hasil Yang Diharapkan</b>	Aplikasi dapat menghapus data ATM dan mampu menampilkan data ATM yang baru

### 5.1.2.2 Kasus Uji Validasi Tampilan Pengguna

Proses pengujian tampilan pengguna dilakukan dengan kasus uji masing-masing yang akan ditunjukkan oleh tabel-tabel di bawah ini.

Tabel 5. 7 Kasus Uji Validasi Buka Aplikasi

<b>Nomor Kasus Uji</b>	<b>VAL_06</b>
<b>Nama Kasus Uji</b>	Kasus uji buka aplikasi
<b>Objek Uji</b>	Kebutuhan Fungsional
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat membuka aplikasi dan menampilkan <i>splashscreen</i> awal
<b>Prosedur Uji</b>	Tekan <i>icon</i> aplikasi pada daftar menu <i>smartphone</i>
<b>Hasil Yang Diharapkan</b>	Aplikasi dapat terbuka dengan pertama kali menampilkan halaman <i>splashscreen</i> , setelah itu masuk pada halaman <i>home</i>

Tabel 5. 8 Kasus Uji Validasi Lihat Halaman *Home*

<b>Nomor Kasus Uji</b>	<b>VAL_07</b>
<b>Nama Kasus Uji</b>	Kasus uji lihat halaman <i>home</i>
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_006_01)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk menampilkan dan kembali ke halaman <i>home</i>
<b>Prosedur Uji</b>	1. Buka aplikasi 2. Masuk pada halaman <i>home</i>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat tampilan <i>home</i> dan dapat kembali ke menu <i>home</i> setelah membuka menu lainnya

Tabel 5. 9 Kasus Uji Validasi Lihat ATM Terdekat

<b>Nomor Kasus Uji</b>	<b>VAL_08</b>
<b>Nama Kasus Uji</b>	Kasus uji lihat ATM terdekat
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_002_01)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi



	dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk melihat ATM terdekat
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu ATM terdekat</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat informasi mengenai ATM terdekat dari posisi pengguna

Tabel 5. 10 Kasus Uji Validasi Pilih ATM

<b>Nomor Kasus Uji</b>	VAL_09
<b>Nama Kasus Uji</b>	Kasus uji pilih ATM
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_002_02)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk memilih ATM yang diinginkan oleh pengguna
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu ATM terdekat atau menu daftar lokasi</li> <li>4. Pilih ATM yang diinginkan</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat memilih ATM yang diinginkan untuk dapat memperoleh informasi

Tabel 5. 11 Kasus Uji Validasi Lihat Detail ATM

<b>Nomor Kasus Uji</b>	VAL_10
<b>Nama Kasus Uji</b>	Kasus uji lihat detail ATM
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_002_03)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk melihat detail ATM yang diinginkan
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> </ol>

	<ol style="list-style-type: none"> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu ATM terdekat atau menu daftar lokasi</li> <li>4. Pilih ATM yang diinginkan</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat informasi lebih detail mengenai ATM yang telah dipilih

Tabel 5. 12 Kasus Uji Validasi Lihat Peta

<b>Nomor Kasus Uji</b>	VAL_11
<b>Nama Kasus Uji</b>	Kasus uji lihat ATM pada peta
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_002_04)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk menampilkan posisi ATM dalam peta
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu ATM terdekat atau daftar lokasi</li> <li>4. Memilih ATM yang diinginkan</li> <li>5. Memilih menu peta pada halaman detail ATM</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat posisi ATM yang dipilihnya pada peta

Tabel 5. 13 Kasus Uji Validasi Lihat Jalur

<b>Nomor Kasus Uji</b>	VAL_12
<b>Nama Kasus Uji</b>	Kasus uji lihat jalur
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_002_05)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk menampilkan proses menunjukkan jalur yang harus ditempuh dari posisi pengguna ke lokasi ATM yang dipilih
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> </ol>

	<ol style="list-style-type: none"> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu ATM terdekat atau daftar lokasi</li> <li>4. Memilih ATM yang diinginkan</li> <li>5. Memilih menu petunjuk jalur pada halaman detail ATM</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat jalur yang harus dituju untuk bisa sampai ke lokasi ATM

Tabel 5. 14 Kasus Uji Validasi Lihat Rute

<b>Nomor Kasus Uji</b>	VAL_13
<b>Nama Kasus Uji</b>	Kasus uji lihat rute
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_002_06)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk menampilkan rute yang harus ditempuh oleh pengguna untuk dapat sampai ke lokasi ATM dari yang dipilih
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu ATM terdekat atau daftar lokasi</li> <li>4. Memilih ATM yang diinginkan</li> <li>5. Memilih menu petunjuk rute pada halaman detail ATM</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat rute yang harus ditempuh untuk bisa sampai ke lokasi ATM

Tabel 5. 15 Kasus Uji Validasi Lihat Daftar Lokasi

<b>Nomor Kasus Uji</b>	VAL_14
<b>Nama Kasus Uji</b>	Kasus uji lihat daftar lokasi
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_003_01)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam

	menyediakan fasilitas untuk melihat fitur daftar lokasi
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu daftar lokasi</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat daftar bank-bank yang ada di kota Malang

Tabel 5. 16 Kasus Uji Validasi *Search* ATM

<b>Nomor Kasus Uji</b>	VAL_15
<b>Nama Kasus Uji</b>	Kasus uji <i>search</i> ATM
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_003_02)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam melakukan <i>search</i> ATM dari tampilan daftar ATM yang ada di kota Malang
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu daftar lokasi</li> <li>4. Memasukkan nama salah satu ATM pada kolom <i>search</i></li> <li>5. Menampilkan nama ATM yang diinginkan</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat tampilan nama ATM yang diinginkan, sesuai dengan nama ATM yang dimasukkan pada kolom <i>search</i>

Tabel 5. 17 Kasus Uji Validasi Lihat Peta Lokasi

<b>Nomor Kasus Uji</b>	VAL_16
<b>Nama Kasus Uji</b>	Kasus uji lihat peta lokasi
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_004_01)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam

	menyediakan fasilitas untuk memvisualisasikan lokasi seluruh ATM dalam bentuk peta dan ditandai dengan <i>marker</i>
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu peta lokasi</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat posisi seluruh ATM pada peta yang ditandai dengan <i>marker</i>

**Tabel 5. 18** Kasus Uji Validasi Lihat Tentang Aplikasi

<b>Nomor Kasus Uji</b>	VAL_17
<b>Nama Kasus Uji</b>	Kasus uji lihat tentang aplikasi
<b>Objek Uji</b>	Kebutuhan Fungsional (SRS_005_01)
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan fungsional dalam menyediakan fasilitas untuk menampilkan detail informasi mengenai pembuat aplikasi
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Buka aplikasi</li> <li>2. Masuk pada halaman <i>home</i></li> <li>3. Pilih menu ATM tentang aplikasi</li> </ol>
<b>Hasil Yang Diharapkan</b>	Pengguna dapat melihat informasi mengenai pembuat aplikasi

### 5.1.2.3 Hasil Pengujian Validasi

Berdasarkan pada kasus uji yang dilakukan akan didapatkan hasil pengujian. Hasil pengujian validasi akan ditampilkan pada Tabel 5.19 berikut.

**Tabel 5. 19** Hasil Pengujian Validasi

No	Nomor Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Status Validasi
1	VAL_01	Admin memasukkan <i>username</i> dan <i>password</i>	Admin dapat masuk ke sistem	VALID

		yang valid sehingga dapat masuk ke sistem		
2	VAL_02	Admin memasukkan <i>username</i> dan <i>password</i> yang tidak valid atau kosong, sehingga sistem mengirimkan pesan error	Admin tidak dapat masuk ke sistem	VALID
3	VAL_03	Aplikasi dapat menyimpan data ATM yang telah diinputkan ke dalam sistem	Data baru telah berhasil di- <i>input</i>	VALID
4	VAL_04	Aplikasi dapat mengubah data ATM yang sudah ada di dalam sistem	Data lama sudah berhasil diubah	VALID
5	VAL_05	Aplikasi dapat menghapus data ATM yang ada pada sistem	Data yang ada pada sistem telah berhasil dihapus	VALID
6	VAL_06	Aplikasi dapat terbuka dengan pertama kali menampilkan halaman <i>splashscreen</i> , setelah itu dapat masuk pada halaman <i>home</i>	Aplikasi dapat terbuka dengan pertama kali menampilkan halaman <i>splashscreen</i> , setelah itu dapat masuk pada halaman <i>home</i>	VALID
7	VAL_07	Pengguna dapat masuk ke halaman <i>home</i>	Pengguna dapat masuk ke halaman <i>home</i>	VALID
8	VAL_08	Pengguna dapat melihat tampilan daftar ATM terdekat	Pengguna dapat melihat tampilan daftar ATM terdekat	VALID
		Pengguna dapat	Pengguna dapat	

9	VAL_09	melakukan pilih ATM yang diinginkan	melakukan pilih ATM yang diinginkan	VALID
10	VAL_10	Pengguna dapat melihat detail informasi ATM yang telah dipilih	Pengguna dapat melihat detail informasi ATM yang telah dipilih	VALID
11	VAL_11	Pengguna dapat melihat posisi ATM yang dipilihnya pada peta	Pengguna dapat melihat posisi ATM yang dipilihnya pada peta	VALID
12	VAL_12	Pengguna dapat melihat jalur yang harus dituju untuk bisa sampai ke lokasi ATM	Pengguna dapat melihat jalur yang harus dituju untuk bisa sampai ke lokasi ATM	VALID
13	VAL_13	Pengguna dapat melihat rute yang harus ditempuh untuk bisa sampai ke lokasi ATM	Pengguna dapat melihat rute yang harus ditempuh untuk bisa sampai ke lokasi ATM	VALID
14	VAL_14	Pengguna dapat melihat tampilan menu daftar lokasi berupa daftar bank-bank yang ada di kota Malang	Pengguna dapat melihat tampilan menu daftar lokasi berupa daftar bank-bank yang ada di kota Malang	VALID
15	VAL_15	Pengguna dapat melihat tampilan nama ATM yang diinginkan, sesuai dengan nama ATM yang dimasukkan pada kolom <i>search</i>	Pengguna menemukan nama ATM yang diinginkan, sesuai dengan nama ATM yang dimasukkan pada kolom <i>search</i>	VALID
16	VAL_16	Pengguna dapat melihat posisi seluruh ATM pada peta yang ditandai dengan <i>marker</i>	Pengguna dapat melihat posisi seluruh ATM pada peta yang ditandai dengan <i>marker</i>	VALID
		Pengguna dapat melihat	Pengguna dapat melihat	

17	VAL_17	informasi mengenai pembuat aplikasi	informasi mengenai pembuat aplikasi	VALID
----	--------	-------------------------------------	-------------------------------------	-------

### 5.1.3 Pengujian Kompatibilitas

Pengujian kompatibilitas digunakan untuk mengetahui kompatibilitas antarmuka sistem pada sistem operasi Android dengan beberapa versi API yaitu API versi 2.3.5, API versi 4.1.2 dan API versi 4.2.2

#### 5.1.3.1 Pengujian Kompatibilitas Android Versi 2.3.5

Pengujian kompatibilitas sistem pada sistem operasi Android versi 2.3.5 dilakukan untuk mengetahui validitas kinerja fitur-fitur yang disediakan oleh sistem terkait dengan antarmuka sistem. Tabel 5.20 menjelaskan prosedur dan hasil kasus uji pengujian kompatibilitas pada sistem operasi Android versi 2.3.5.

**Tabel 5. 20** Kasus Uji dan Hasil Pengujian Kompatibilitas pada Android versi 2.3.5

<b>Nama Kasus Uji</b>	Pengujian Kompatibilitas pada Android versi 2.3.5
<b>Objek Uji</b>	Kebutuhan NonFungsional
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui validitas kinerja dari sistem dalam menyediakan fitur-fitur dan antarmuka pengguna
<b>Prosedur Uji</b>	Membuka setiap halaman sesuai dengan spesifikasi kebutuhan system
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka system
<b>Hasil yang didapatkan</b>	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka system
<b>Status Validasi</b>	Valid

#### 5.1.3.2 Pengujian Kompatibilitas Android Versi 4.1.2

Pengujian kompatibilitas sistem pada sistem operasi Android versi 4.1.2 dilakukan untuk mengetahui validitas kinerja fitur-fitur yang disediakan oleh



sistem terkait dengan antarmuka sistem. Tabel 5.21 menjelaskan prosedur dan hasil kasus uji pengujian kompatibilitas pada sistem operasi Android versi 4.1.2

**Tabel 5. 21** Kasus Uji dan Hasil Pengujian Kompatibilitas pada Android Versi 4.1.2

<b>Nama Kasus Uji</b>	Pengujian Kompatibilitas pada Android versi 4.1.2
<b>Objek Uji</b>	Kebutuhan NonFungsional
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui validitas kinerja dari sistem dalam menyediakan fitur-fitur dan antarmuka pengguna
<b>Prosedur Uji</b>	Membuka setiap halaman sesuai dengan spesifikasi kebutuhan system
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka system
<b>Hasil yang didapatkan</b>	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka system
<b>Status Validasi</b>	Valid

### 5.1.3.3 Pengujian Kompatibilitas Android Versi 4.2.2

Pengujian kompatibilitas sistem pada sistem operasi Android versi 4.2.2 dilakukan untuk mengetahui validitas kinerja fitur-fitur yang disediakan oleh sistem terkait dengan antarmuka sistem. Tabel 5.22 menjelaskan prosedur dan hasil kasus uji pengujian kompatibilitas pada sistem operasi Android versi 4.2.2

**Tabel 5. 22** Kasus Uji dan Hasil Pengujian Kompatibilitas pada Android Versi 4.2.2

<b>Nama Kasus Uji</b>	Pengujian Kompatibilitas pada Android versi 4.2.2
<b>Objek Uji</b>	Kebutuhan NonFungsional
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk mengetahui validitas kinerja dari sistem dalam menyediakan fitur-fitur dan antarmuka pengguna
<b>Prosedur Uji</b>	Membuka setiap halaman sesuai dengan spesifikasi

	kebutuhan system
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka system
<b>Hasil yang didapatkan</b>	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan implementasi antarmuka system
<b>Status Validasi</b>	Valid

## 5.2 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian aplikasi Android *mobile* pencarian lokasi ATM terdekat di kota Malang yang telah dilakukan. Proses analisis yang dilakukan adalah analisis pada pengujian validasi atau fungsional sistem serta pengujian kompatibilitas.

### 5.2.1 Analisis Pengujian Unit

Proses analisis terhadap hasil pengujian unit dilakukan dengan melihat kesesuaian fungsi dari implementasi unit model yang diuji dengan hasil perancangan perangkat lunak yang telah dirancang sebelumnya. Berdasarkan hal tersebut, maka dapat diambil kesimpulan bahwa unit modul dari program sudah memenuhi kebutuhan fungsional yang telah dirancang pada tahap perancangan.

### 5.2.2 Analisis Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat kesesuaian antara hasil kinerja sistem dengan daftar kebutuhan. Berdasarkan hasil pengujian validasi dengan metode *Black-Box testing* pada aplikasi pencarian lokasi ATM terdekat di kota Malang, menunjukkan nilai dengan presentase 100%, aplikasi sudah memenuhi spesifikasi kebutuhan yang telah dianalisa.

### 5.2.3 Analisis Pengujian Kompatibilitas

Proses analisis terhadap hasil pengujian kompatibilitas yang dilakukan pada aplikasi *mobile* terhadap sistem operasi Android dengan versi yang berbeda, yaitu sistem operasi Android versi 2.3.5, API versi 4.1.2 dan API versi 4.2.2 dilakukan untuk melihat kesesuaian antara hasil kinerja sistem dengan daftar

kebutuhan. Berdasarkan hasil pengujian kompatibilitas dapat disimpulkan bahwa implementasi dan fungsionalitas aplikasi Android *mobile* pencarian lokasi ATM terdekat di kota Malang telah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.



## BAB VI PENUTUP

### 6.1 Kesimpulan

Berdasarkan hasil analisa, perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Aplikasi *mobile* pencarian lokasi ATM terdekat di kota Malang telah dibuat sesuai dengan spesifikasi kebutuhan yang telah dianalisa.
2. Aplikasi *mobile* pencarian lokasi ATM terdekat di kota Malang yang menggunakan konsep *hybrid mobile development* sesuai dengan perancangan yang telah dibuat dan dapat digunakan sebagai salah satu media untuk mencari informasi mengenai lokasi ATM terdekat.
3. Pengambilan data pada *online storage* aplikasi telah berhasil diimplementasikan dengan menggunakan JSON sebagai cara pengambilan data.
4. Berdasarkan hasil pengujian validasi dengan metode *Black-Box testing* pada sistem menunjukkan nilai dengan presentase 100%, sistem sudah memenuhi spesifikasi kebutuhan sesuai perancangan.
5. Hasil pengujian kompatibilitas pada sistem menunjukkan bahwa sistem dinyatakan kompatibel dengan sistem operasi Android dengan API versi 2.3.5, API versi 4.1.2, API versi 4.2.2.

### 6.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi *mobile* pencarian lokasi ATM terdekat di kota Malang selanjutnya adalah :

1. Perlu dilakukan pengembangan lebih lanjut dengan optimalisasi terhadap tampilan pada antarmuka pengguna menggunakan HTML5, CSS3, dan javascript agar kompatibel di semua *platform mobile*.
2. Dilakukan pengembangan untuk beberapa mobile OS lainnya, seperti windows phone, iOS, dan symbian.

## DAFTAR PUSTAKA

- [AZZ-12] Azzahra. 2012. Cara Kerja dan Arsitektur Android. [http://www.maxiandroid.blogspot.com/2014/03/cara\\_kerja\\_dan\\_arsitektur\\_Android.html](http://www.maxiandroid.blogspot.com/2014/03/cara_kerja_dan_arsitektur_Android.html) [diakses pada tanggal 19 Maret 2014]
- [EMS-13] EMS, Tim. 2013. Pemrograman Mobile dengan PhoneGap. PT. Elex Media Komputindo, Jakarta.
- [ESS-11] Essayad, Abdesslam. 2011. *Design and Implementation of a platform or location based services : a case study of GIS of archeological and handicraft of ez Medina*. International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3.
- [GIN-10] Gintoro, Iwan Wijaya Suharto, Febiyan Rachman, Daniel Halim. Analisis dan Perancangan Sistem Pencarian Taksi Terdekat dengan Pelanggan Menggunakan Layanan Berbasis Lokasi. Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010).
- [HER-11] Hermawan. S, Stepanus. 2011. Mudah Membuat Aplikasi Android. Andi, Yogyakarta.
- [JQU-14] jQuery, akses dari <http://www.jquery> [diakses pada tanggal 14 Mei 2014]
- [JSO-14] JSON, diakses dari <https://json.org/json-id.html> [diakses pada tanggal 19 Juli 2014]
- [LES-13] Lestari, Puji Sakti. 2013. Sistem Informasi Penjualan dan Pemesanan Barang pada FIM Store. Universitas Komputer Indonesia: Bandung.
- [MUL-10] Mulyadi, Adi. 2010. Membangun Aplikasi Android. Multimedia Center Publishing, Yogyakarta.
- [NUG-05] Nugroho, Bunafit. 2005. Database Relational dengan MySQL. Andi Offset, Yogyakarta.
- [PET-14] Pettit, Nick. 2014. *The Beginner's Guide to Ratchet: a Mobile Web App Framework*. <http://blog.teamtreehouse.com/beginners-guide-ratchet-mobile-web-app-framework> [diakses pada tanggal 10 Juni 2014]

- [RIY-11] Riyanto. 2011. Sistem Informasi Geografis Berbasis Mobile. Gava Media, Yogyakarta.
- [ROM-12] Rompas, B.R. 2012. Jurnal Aplikasi Location Based Service Pencarian Tempat di Kota Manado Berbasis Android. Jurusan Teknik Elektro-FT UNSRAT, Manado.
- [SAF-11] Safaat. H, Nazaruddin. 2011. Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Informatika, Bandung.
- [STE-06] Steiniger, Stefan., Noun, Moritz., Edwardes, Alistair. 2006. *Foundations of Location Based Services*.
- [SUN-07] Sunyoto, Andi. 2007. "Ajax Membangun Web dengan Teknologi Asynchronous Javascript & XML", Yogyakarta.
- [SUR-12] Suprianto, Dodit dan Agustina S.Kom, M.Pd, Rini. 2012. Pemrograman Aplikasi Android – *Step by Step* Membuat Aplikasi Android untuk Smartphone dan Tablet. Yogyakarta : MediaKom.
- [SVE-14] Svennerberg, Gabriel. 2014. *Adding Multiple Markers to Google Maps from JSON*. <http://www.svennerberg.com/2014/03/adding-multiple-markers-to-google-maps-from-json/> [diakses pada tanggal 18 Juli 2014]
- [VIR-01] Virrantaus, K., Markkula, J., Garmash, A., Terziyan, Y.V., 2001. Warnock, Matt. 2008. *Geolocation Via Cell Tower Data*.
- [WAH-13] Wahana Komputer. 2013. Membuat sendiri Aplikasi Web Mobile Menggunakan jQuery Mobile. CV Andi Offset, Semarang.
- [WAR-10] Wardana. 2010. Menjadi Master PHP dengan *Framework CodeIgniter*. PT. Elex Media Komputindo, Jakarta.
- [W3C14] W3C Working group. *Web services Architecture*. <http://www.w3.org/TR/2014/NOTE-ws-arch-20040211> [diakses pada tanggal 30 Maret 2014]
- [WIK-14] Wikipedia. The Free Encyclopedia. 2014. *Haversine Formula*. <http://en.wikipedia.org/wiki/Haversine-formula> [diakses pada tanggal 10 Juni 2014]

## LAMPIRAN

## Lampiran 1

## Data ATM di Kota Malang

Nama ATM	Alamat
ATM BCA Kiosk Blimbing	Jl. A. Yani 183 - Malang - 65125
ATM BCA Kiosk Borobudur	Jalan Candi Waringin Lawang, Lowokwaru, Malang City, East Java, Indonesia
ATM BCA Kiosk Sawojajar	Jl. Danau Toba Blok B6
ATM BCA Kiosk Gondanglegi	Jl. Gajah Mada 36 A - Gondanglegi - 65174
ATM BCCA Kiosk Galunggung	Jl. Galunggung No. 78, Malang
ATM BCA Kiosk Gatot Subroto. Mlg	Jl. Gatot Subroto 21 D - Malang - 65118
ATM BCA Kiosk Basuki Rachmat 2	Jl. Jend. Basuki Rachmat 70-74 - Malang- 65111
ATM BCA Kiosk Kawi	Jl. Kawi ( atas ) 36 - Malang - 65116
ATM BCA Kiosk Kudus	Jl. KH Zainul Arifin 78 - Malang - 65118
ATM BCA Kiosk Kantor Kas UNIBRAW	Jl. komplek Univ. Brawijaya, Malang
ATM BCA Kiosk Kyai Tamin	Jl. kyai Tamin No.109, Malang
ATM BCA Kiosk Dinoyo	Jalan Mayjend M.T. Haryono, Lowokwaru, Malang City, East Java 65145, Indonesia
ATM BCA Kiosk RSIA Husada Bunda	Jl. Pahlawan Trip No. 2, Malang
ATM BCA Kiosk Sudirman. Mlg	Jl. Panglima Sudirman 27 - Malang - 65111
ATM BCA Kiosk Tumpang	Jl. Raya 1, Tumpang - Malang - 65156
ATM BCA Kiosk Dieng Plaza	Jl. Raya Langsep No. 2
ATM BCA Kiosk Kantor Kas Sawojajar	Jl. Raya Sawojajar
ATM BCA Kiosk Sukun	Jl. S. Supriyadi 65 - Malang - 65117
ATM BCA Kiosk Dampit	Jl. Semeru Selatan 32 - Dampit – Malang-65181
ATM BCA Kiosk Sumber Pucung	Jl. Sudirman 183 - Sumberpucung - 65165
ATM BCA Kiosk Vokasi Brawijaya	Jl. Veteran No. 16, Malang

ATM BCA Kiosk RS. Lavalette	Jl. WR Supratman 10, Malang
ATM BCA Kiosk Univ Malangkececwara	Jl. STIE Malangkececwara (Kampus ABM) Jl. T candi Kalasan
ATM BCA Indomaret Esberg	Jl. Esberg 8A, karang Widoro Malang 65151
Mandiri ATM KCP Malang Suprpto	Jl. Jaksa Agung Suprpto 65, Malang
Mandiri ATM KC Malang Wahid Hasyim	Jl. KH Wahid Hasyim 5-7, Malang
Mandiri ATM KK Malang MT Haryono	Jl. MT Haryono n0. 131, Malang
Mandiri ATM KCP Malang Batu	Jl. Dewi Sartika No. 45, Batu, Malang
Mandiri ATM Plaza Dieng	Jl. Langsep No.2, Malang
Mandiri ATM Taman Sulfat	Jl. Sulfat, Malang
Mandiri ATM Malang Town Square	Jalan Veteran, Lowokwaru, Malang City, East Java 65145, Indonesia
Mandiri ATM Pertokoan Agus Salim	Jl. Agus Salim, Malang
Mandiri ATM RS Darmayu	Jl. Dr Sutomo no.50 Ponorogo, Malang
Mandiri ATM KCP Suprpto	Jl. Jaksa Agung Suprpto No. 65 Malang
Mandiri ATM Indomaret Kemuning	Jl. Kemuning, Malang
Mandiri ATM Luwes Supermarket	Jl. KH Ahmad Dahlan ponorogo, Malang
Mandiri ATM KC Malang Wahid Hasyim	Jl. KH Wahid Hasyim 5-7, Malang
Mandiri ATM Indomaret Kol Sugiono	Jl. Kol Sugiono 117 Mergosono, Malang
Mandiri ATM Indomaret Gadang	Jl. Kol Sugiono 2, Malang
Mandiri ATM Indomaret Wuni	Jl. Mayjen Sungkono No. 38 Malang
Mandiri ATM Indomaret Tlogomas	Jl. MT Haryono 210 a, Malang
Mandiri ATM Indomaret Bedali	Jl raya bedali, malang
Mandiri ATM Indomaret Univ Muhamadiyah	Jl raya Landungsari,malang
Mandiri ATM PO Zena	Jl. kol sugiyono,malang
Mandiri ATM Gedung Arsip ex	jl. a. yani, malang



bapindo	
Mandiri ATM Alfa Midi A. Yani	jl. a. yani 85, malang
Mandiri ATM Indomaret Bandulan	Jl. bandulan,malang
ATM Danamon Kawi	Jl. Kawi No. 15
ATM Danamon Adira Finance	Jl. Letjen S.Parman No.149
ATM Danamon Koperasi Mahasiswa Universitas Brawijaya	Jl. MT Haryono
ATM Danamon Malang plaza Dieng	Jl. Langsep No.2
ATM Danamon RS panti Nirmala	Jl. Kebalen Wetan
ATM Danamon Pasar Baru Raya (Ex. Gedung Bank Rama)	Jl. Raya Kawi No.15
ATM Danamon Giant Sawojajar	Jl. Danau Toba
ATM Danamon Indomaret Soekarno 3	Jalan Soekarno Hatta, Lowokwaru, Malang City, East Java 65141, Indonesia
ATM Danamon CDM Malang Kawi	Jl. Raya Kawi No.15
ATM Danamon Solusi Emas Syariah Pasar	Jl. Pasar Besar
ATM Danamon Solusi Emas Syariah Kapanjen	Jl. Ahmad Yani
ATM Bukopin Kapanjen	Jl. Kawi No. 37 Blok A6, Kapanjen
ATM Bukopin KK Dinoyo Malang	Gang 1, Lowokwaru, Malang City, East Java 65144, Indonesia
ATM Bukopin Cabang Blimbing	Jalan Letnan Jenderal S. Parman, Blimbing, Malang City, East Java 65126, Indonesia
ATM Bukopin Ijen Malang	Jl. Ijen No.90-92
ATM Bukopin Mitra I	Jl. KH Agus Salim No.10-16
ATM Bukopin Cabang Malang	Jl. Semeru No.35
ATM BRI KCP Pasar Besar	Jl. Sutan Syahrir No. 6, Kota Malang
ATM BRI KCP Sawojajar	Jl. Danau Toba E5/12 Sawojajar, Kota Malang
ATM BRI Kantor Kas Universitas Negeri Malang	Jl. Surabaya, Kota Malang
ATM BRI Kantor Kas RSUD Saiful	Jl. Jaksa Agung Suprpto No. 2, Kota Malang

Anwar	
ATM BRI Universitas Negeri Malang	Kampus Universitas Negeri Malang, Jl. Surabaya No. 6, Kota Malang
ATM BRI Politeknik Negeri Malang	Kampus Polinema, jl. Soekarno-Hatta No.9, Malang
ATM BRI MIN Malang 1	Jl. Bandung No. 7C, Kota Malang
ATM BRI RSU Saiful Anwar	Gedung UGD, jl. Jaksa Agung Suprpto No. 2, Kota Malang
ATM BRI RS. Panti Waluyo (RKZ)	Jl. Nusa Kambangan NO. 56, Kota Malang
ATM BRI Mitra I	Mall Mitra I, Jl. Agus Salim, Kota Malang
ATM BRI Alun-Alun Mall	Jl. Merdeka Timur, Kota Malang
ATM BRI Plasa Dieng	Jl. Raya Langsep No. 2, Kota Malang
ATM BRI SPBU Tidar	Jl. Puncak Mandala No. A1-A2, Kota Malang
ATM BRI SPBU Tlogomas	Jl. Raya Tlogomas, Kota Malang
Galeri ATM BRI Kantor Cabang Malang Martadinata	Jl. Laksamana Martadinata No. 80, Kota Malang
ATM BNI Apotik Kimia Farma	Jl. Kawi No. 22 Malang Kota /Klojen
ATM BNI Cabang BNI Syariah Malang	Jl. Jaksa Agung Suprpto No. 48 Malang Kota/Klojen
ATM BNI Batu Plaza	Jl. Gajah Mada No. 1, Batu Malang Utara/Batu
ATM BNI Cabang Malang 1	Jl. Basuki Rachmat No.75-77 Malang Kota/Klojen
ATM BNI Capem Batu	Jl. Dewi Sartika No.7D, Batu Malang Utara/Batu
ATM BNI Capem ITN 1	Jl. Bendungan Sigura-gura No.2 Malang Utara/Lowokwaru
ATM BNI Capem ITN 2	Jl. Raya Losawi Km 2, Karangploso Malang Utara/Karangploso
ATM BNI Capem Univ. Muhammadiyah 3	Jl. Raya Tlogo Mas Malang Utara/Lowokwaru
ATM BNI Dian Swalayan	Jl. MT Haryono No.11 Malang Utara/Lowokwaru
ATM BNI Dieng Plaza	Jl. Raya langsep No. 2 Malang Barat/Sukun
ATM BNI Kantor Telkom	Jl. Ahmad Yani No.9-11 Malang Utara/Blimbing
ATM BNI Kompleks Pindad Turen 1	Jl. Panglima Sudirman No.1, Turen Malang Selatan/Malang

Mitra Swalayan 1	Jl. KH. Agus Salim 10/16 Malang Kota/Klojen
Mitra Swalayan 2	Jl. Letjend Sutoyo No.32-34 Malang Kota/Klojen
Perpustakaan Unibraw	Jl. Veteran No.16 Malang Kota/Klojen
Pondok Blimbing Indah	Jl. Panji Suroso Malang Utara /Blimbing
RK. Hardlent Medika Utama	Jl. Galunggung Malang Kota/Klojen
RM Citra	Jl. Kawi Malang Kota/Klojen
RS. Panti Nirmala	Jl. Kol. Sugiono No.19-24 Malang Kota/Klojen
RS. islam Dinoyo	Jalan Mayjend M.T. Haryono, Lowokwaru, Malang City, Indonesia
Ruko Pasar Gondanglegi	Jl. Panglima Sudirman, Gondanglegi Malang Selatan/Gondanglegi
Sarinah Plaza	Jl. Jend. Bauki Rachmat Malang Kota/Klojen
Sarinah Sawojajar 2	Jl. danau Maninjau 14, Sawojajar Malang Timur/Gondanglegi
STAIN	Jl. Gajayana No.50 Malang Utara/Lowokwaru
Stasiun Kereta Api	Jl. Trunojoyo No.10 Malang Kota/Klojen
Telkom Singosari	Jl. Raya Singosari Malang Utara/Singosari
Univ. Widyagama	Jl. Borobudur No.12 Malang Utara/Blimbing
CIMB Niaga Achmad Yani	Jl. Achmad Yani No. 18E Malang
CIMB Niaga Galunggung	Jl. Galunggung No. 58 Malang
CIMB Niaga Basuki Rachmat	Jl. Basuki Rachmat No. 26-28 Malang
CIMB Niaga RS Syaiful Anwar	RS Syaiful Anwar, Jl. Jaksa Agung Suprpto No. 2 Malang
CIMB Niaga Soekarno Hatta	Jl. Soekarno Hatta No. 2-2A, Malang
OP Kawi	Jl. Kawi Atas No. 36A, Klojen
Mitra II, Pusat Perbelanjaan Mitra II	Jl. Letjen Sutoyo No. 32-34
Sampoerna, Pabrik Rokok HM Sampoerna	Jl. Industri BaratvNo. 2, Kec. Blimbing Malang
Op Lavalete, Rumah Sakit Lavalette	Jl. WR. Supratman No.10, Malang
CIMB Niaga Batu Malang	Jl. Diponegoro Batu
CIMB Niaga Kyai Tamin	Jl. Kyai Tamin 115-117
CIMB Niaga Merdeka	Jl. Merdeka Timur No. 4

Plaza Dieng	Jl. Raya Dieang
Univ Muhammadiyah	Jl. Raya Tlogomas km.8
CIMB Niaga Kepanjen	Jl. A. Yani 74 Kepanjen
CIMB Niaga Malang Town Square	Jl. Veteran GE 2 No. 12
CIMB Niaga S.Parman	Jl. S. Parman 120
Univ Brawijaya	Jl. MT Haryono 169
ATM BII Basuki Rahmat	Jl. Basuki Rahmat No.91-93 Malang Kota/Klojen
ATM BII Carrefour Express	Jl. Jend. A Yani 2 A Malang Utara/Blimbing
ATM BII Kawi Malang	Jl. Kawi No.11 Malang Kota/Klojen
ATM BII Malang Town Square	Jl. Veteran No.2 Malang Kota/Klojen
Alfamidi Bandulan	Jalan Bandulan No.12, Sukun, Malang City, East Java 65146, Indonesia
BII ATM Kyai Tamin	Jl. Kyai Tamin No. 22B Kota Malang
BII ATM Alfamidi Gatot Subroto, Malang	Jl. Gatot Subroto No.69 Kota Malang
ATM BII Carrefour Malang	Jl. Achmad Yani No. 2 Blimbing Malang
BII ATM KCP Kawi	Jl. Kawi Malang No. 11 Malang
BII ATM KCP Soekarno Hatta	Jl. Soekarno Hatta A-10 Kav 07 Kota Malang
BII ATM Mall Olympic Garden, Malang	Jl. Kawi No. 24 Malang
BII ATM Pasar Swalayan Mitra I, Malang	Jl. KH. Agus Salim No. 10-16 Malang
BII ATM Plaza Dieng	Jl. Raya Dieng No. 2 Malang
ATM Panin KCU Malang 1	Jl. Sultan Agung No. 14 Malang
ATM Panin KCP Pasar Basar	Jl. Kyai Tamin Komp. Pasar Besar Lt. 1 Malang 65117
ATM Panin KCP Blimbing	Jl. Terusan Borobudur No. 47 A, Lowok Waru Malang, Jawa Timur
ATM Panin KCP Lawang	Jl. Thamrin 190 Lawang - Malang
ATM Panin KCP Kepanjen	Jl. Kawi 37 B3, Kepanjen Malang
ATM Panin Plaza SPBU Malang	Jl. Danau Toba Sawojajar Malang
ATM Panin Mall Olympic Garden – Malang	Jl. Kawi 24 Malang
ATM Panin KCP Batu	Jl. Dewi Sartika No. 4 Temas, Batu Malang

ATM Panin KCU Malang 2	Jl. Sultan Agung No. 14 Malang
ATM BTN KC Malang 1	Jl. Ade Irma Suryani Nasution No. 2-4 Malang 65119
ATM BTN KCP Sawojajar	Perum Perumnas Sawojajar, Jl. Danau Toba No. 01 Malang 65138
ATM BTN KCP Jaksa Agung	Jl. Jaksa Agung Suprpto No. 54
ATM BTN KCP Unibraw	Universitas Brawijaya Gedung IKA, Jl. Veteran No. 16-A Malang 65145
ATM BTN KCP UIN Malang	Universitas Islam Negeri Malang, Jl. Gajayana No. 50 Malang 65144
ATM BTN Poltek Malang	Kampus Politeknik Malang, Jl. Soekarno Hatta No 9 Malang
ATM BTN KPRK Malang	JL MERDEKA SELATAN 5 MALANG
ATM BTN Singosari Malang	JL RAYA SINGOSARI, MALANG
ATM BTN Indomaret Malang	JL SURABAYA KOTA MALANG
ATM BTN Pesantren Alrifai	JL. RAYA GONDANGLEGI-MALANG
ATM BTN KCPS Malang	JL. SUKARNO HATTA, MALANG
ATM Permata Bank Giant Dinoyo	ATM Center Giant Dinoyo Jl. MT. Haryono No. 145 Malang
ATM Permata Bank Mall Olympic Garden	Mall Olympic Garden Jl. Kawi no 24, Malang 65116
ATM Permata Bank Bromo 1	Jalan Bromo, Malang, Kota Malang, Jawa Timur 65112, Indonesia
ATM Bank Permata Plaza Matos	Plaza Matos Kav GS 5 No. 8 Jl Veteran No.2, East Java 65145
ATM Bank Permata Carrefour Malang	Jl. A. Yani 2A,, East Java 65163