

APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID

SKRIPSI



Disusun Oleh :

**FIRDAUS GEDE RIZQI
NIM. 0812525**

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
2015**

SECRET AA DISSEMINATION OF THIS INFORMATION IS PROHIBITED

SECRET

SECRET

SECRET
SECRET

SECRET
SECRET
SECRET
SECRET
SECRET

LEMBAR PERSETUJUAN

APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat
Guna Mencapai Gelar Sarjana Teknik*

Disusun Oleh :

FIRDAUS GEDE RIZQI

NIM : 08.12.525

Mengetahui,

Ketua Jurusan Teknik Elektro S-1



M. Ibrahim Ashari, ST, MT

NIP.P.1030100358

Diperiksa dan Disetujui

Pembimbing I

M. Ibrahim Ashari, ST, MT

NIP. P.1030100358

Pembimbing II

Irmalia Suryani Faradisa, ST, MT

NIP. P.1030000365

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2015**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : FIRDAUS GEDE RIZQI

NIM : 08.12.525

Program Studi : TEKNIK ELEKTRO S-1

Konsentrasi : TEKNIK KOMPUTER

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, September 2015

Yang membuat Pernyataan,



Firdaus Gede Rizqi
NIM : 08.12.525



APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID

Firdaus Gede Rizqi

08.12.525

Jurusan Teknik Elektro S-1, Konsentrasi Teknik Komputer

Fakultas Teknologi Industri, Institut Teknologi Nasional Malang

Jl. Raya Karanglo Km 2 Malang

Abstrak

Pada 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 ATM berbasis Android dengan tujuan untuk mempermudah mengetahui lokasi ATM.

Pada penelitian ini, GPS yang terdapat pada Smartphone Android digunakan sebagai satelit penghubung untuk menampilkan Rute (Map), Database sebagai server untuk Menyimpan data-data ATM yang telah di kumpulkan, dan Smartphone Android sebagai Media Perangkat keras untuk menjalankan Aplikasi Pencarian ATM yang hanya menampilkan List ATM dalam Radius 1 Km.

Pada Penelitian ini aplikasi telah berhasil menampilkan List ATM, Lokasi dan Rute (Map). Dengan dibangunnya aplikasi sistem informasi ini, di harapkan dapat memberikan kemudahan bagi user untuk mengetahui lokasi ATM terdekat di kota madya Malang dalam Radius 1 Km.

Kata kunci : ATM, GPS, Android smartphone

KATA PENGANTAR

Dengan mengucapkan syukur kehadiran Tuhan Yang Maha Esa yang dengan segala Kasih dan Anugerah-Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul:

” APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID”

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata I di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Dr. Ir. Lalu Mulyadi, MTA. Selaku rektor ITN Malang
2. Bapak Ir. Anang Subardi, MT. Selaku Dekan Fakultas Teknologi Industri.
3. Bapak M. Ibrahim Ashari, ST, MT. Selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang dan Dosen Pembimbing I.
4. Bapak Dr. Eng. I Komang somawirata, ST, MT. Selaku Sekretaris Jurusan Teknik Elektro S-1 ITN Malang dan Dosen Penguji I.
5. Ibu Irmalia Suryani Faradisa, ST, MT. Selaku Dosen Pembimbing II.
6. Ibu Yuli Wahyuni, ST, MT. Selaku Dosen Penguji II.
7. Dan semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan. Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, September 2015

Penulis

DAFTAR ISI

Lembar Persetujuan.....	i
Surat Pernyataan Orisinalitas	ii
Abstrak	iii
Kata Pengantar	iv
Daftar isi	v
Daftar Gambar	vii
Daftar Tabel.....	viii

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metode Penelitian.....	3
1.6 Sistematika Penulisan.....	4

BAB II LANDASAN TEORI

2.1 GPS.....	5
2.2 Google Map	5
2.3 Android	7
2.3.1 Versi Android	7
2.4 Sistem Operasi Android	10
2.4.1Fitur Antarmuka Android	12
2.4.2Fitur Aplikasi Android	13
2.4.3Fitur Pengolahan Memori Android	14
2.5 Arsitektur Android	15
2.6 Android Software Development Kit	19
2.7 ADT (Android Development Tools)	19
2.8 Eclipse IDE	20
2.9 Xampp	21
2.9.1Bagian Penting Xampp.....	21
2.9.2Komponen	22
2.10 Location Based Service	22
2.10.1 Pengertian LBS	22

2.10.2 Komponen LBS	23
2.10.3 Dua Unsur Utama LBS	26
2.10.4 Jenis LBS	26
2.10.5 Kelebihan dan Kekurangan LBS	27

BAB III ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa Sistem	28
3.1.1 Deskripsi Umum Sistem	28
3.1.2 Spesifikasi Aplikasi	28
3.2 Perancangan Sistem	29
3.2.1 Analisa Kebutuhan Perangkat Keras	30
3.2.2 Analisa Kebutuhan Fungsional	31
3.2.3 Analisa Kebutuhan Non Fungsional	32
3.2.4 Perhitungan LBS	32
3.2.5 Data ATM	33
3.3 Eclipse	34
3.3.1 MapActivity	35
3.4 Desain Tampilan.....	36
3.5 Flowchart	38

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 implementasi Sistem	40
4.2 persiapan Pembuatan Aplikasi	40
4.2.1 Database Server.....	40
4.2.2 Perhitungan LBS	41
4.3 pengujian	44
4.4 Pengujian Aplikasi Pada Smartphone Android.....	45
4.4.1 Pengujian pada LG Lfino	45
4.4.2 Proses Penginstalan	46
4.5 Pengujian Aplikasi	47
4.5.1 Pengujian Pada Menu Utama	47
4.5.2 Pengujian Aplikasi di ITN 2	47
4.5.3 Pengujian Aplikasi di Perum. Borobudur	49
4.5.4 Pengujian Aplikasi di Soekarno Hatta	51

BAB V PENUTUP

5.1 Kesimpulan	54
5.2 Saran	54

Daftar Pustaka	55
----------------------	----

Lampiran

DAFTAR GAMBAR

Gambar 2.1 Sistem Operasi Android	10
Gambar 2.2 Arsitektur Platform Android	15
Gambar 2.3 Teknologi LBS	22
Gambar 2.4 komponen LBS	23
Gambar 3.1 Project Aplikasi Android	34
Gambar 3.2 MapActivity	35
Gambar 3.3 Antar Muka menu	36
Gambar 3.4 Antar Muka Map View	37
Gambar 3.5 Antar Muka List	37
Gambar 3.6 flowchart Aplikasi	38
Gambar 4.1 Database server	41
Gambar 4.2 hasil perhitungan	43
Gambar 4.3 Proses Penginstalan Aplikasi pada Handphone LG L Fino ..	46
Gambar 4.4 menu utama aplikasi	47
Gambar 4.5 List ATM BNI	47
Gambar 4.6 Rute ATM BNI	48
Gambar 4.7 List ATM BRI	48
Gambar 4.8 Rute Atm BRI	49
Gambar 4.9 List ATM BCA.....	49
Gambar 4.10 Rute ATM BCA	50
Gambar 4.11 List ATM BRI	50
Gambar 4.12 Rute ATM BRI.....	51
Gambar 4.13 List ATM MANDIRI	51
Gambar 4.14 Rute ATM MANDIRI	52
Gambar 4.15 List ATM BNI.....	52
Gambar 4.16 Rute ATM BNI.....	53

DAFTAR TABEL

Tabel 3.1 Deskripsi Kebutuhan Fungsional	31
Tabel 3.2 Kebutuhan Non Fungsional	32
Tabel 3.3 Data ATM	33
Tabel 4.1 Data ATM	42
Tabel 4.2 Hasil Perhitungan Data ATM	42
Tabel 4.3 Hasil Perhitungan Data ATM	44

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pada era globalisasi kini telah berkembang berbagai jenis *Smartphone* yang mengusung banyak sistem operasi, salah satunya adalah sistem operasi *Android*. *Android* merupakan sistem operasi yang di kembangkan oleh perusahaan bernama android Inc yang kemudian di kembangkan oleh *google*, karena *google* melihat banyaknya user yang online dengan perangkat mobile, maka *google* mengira bahwa perangkat mobile ini mempunyai masa depan yang cerah, sehingga *Android* Inc diakusisi oleh *google* di tahun 2005.

Di dunia ini terdapat dua jenis distributor sistem operasi *Android*. Pertama yang dapat dukungan penuh dari *google* atau *google Mail Service* (GMS) dan kedua adalah benar-benar bebas distribusinya tanpa dukungan langsung *google* atau di kenal sebagai *Open Headset Distribution* (OHD).

Automated Teller Machine (ATM) adalah sebuah alat elektronik yang mengijinkan nasabah Bank untuk mengambil uang dan mengecek rekening tabungan mereka tanpa perlu di layani oleh seorang "teller" manusia. Setiap orang pasti mengalami kesulitan dalam mencari ATM, padahal pada saat itu orang tersebut tidak membawa uang tunai. Dari permasalahan sebelumnya penulis menawarkan sebuah solusi yang dapat memecahkan permasalahan tersebut dengan lebih mudah. Yaitu aplikasi layanan lokasi atm menggunakan *LBS*. Aplikasi ini akan membantu *user* menemukan informasi atm yang ada di sekitarnya dengan cara yang baru yaitu menggunakan GPS yang ada di perangkat *smartphone*. Disini penulis menggunakan teknologi *LBS* yang akan menampilkan daftar tempat atm berdasarkan lokasi GPS pengguna melalui *google map* dan juga rute menuju ketempat tujuan.

1.2. Rumusan Masalah

Mengacu pada permasalahan yang telah di uraikan pada latar belakang, maka rumusan masalah dapat di susun sebagai berikut :

1. Bagaimana membangun sebuah aplikasi berbasis GPS yang nantinya akan menunjukan lokasi ATM yang tersebar di kota madya malang.
2. Bagaimana mengkoneksikan aplikasi ini ke dalam lingkup google map menggunakan aplikasi *Eclipse* berbasis *Android platform*.

1.3. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan, maka pembahasan di batasi pada hal-hal sebagai berikut :

1. Menggunakan *Eclipse* sebagai aplikasi *open source* yang berfungsi untuk membangun aplikasi pencarian.
2. Wilayah yang di cari hanya berada di lingkup kota madya malang.
3. Menggunakan *Smartphone Android* sebagai media GPS.
4. ATM yang di cari meliputi ATM BCA, ATM BRI, ATM BNI dan ATM MANDIRI.
5. Menampilkan ATM dalam Radius 1 km.

1.4. Tujuan

Dengan dibangunnya sistem ini diharapkan dapat memberikan kemudahan bagi user untuk mengetahui lokasi ATM di kota madya malang dalam radius 1 km.

1.5. Metode Penelitian

Metodologi yang digunakan pada penelitian skripsi ini adalah sebagai berikut :

1. Studi Literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa Kebutuhan Aplikasi

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem baik hardware maupun software, dimana nantinya akan digunakan sebagai acuan perancangan aplikasi.

3. Perancangan Aplikasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun aplikasi ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari aplikasi yang akan dibuat.

4. Coding

Tahap ini dilakukan implementasi hasil rancangan kedalam baris-baris kode program yang dapat dimengerti oleh komputer.

5. Uji coba aplikasi

Pada tahap ini, aplikasi yang telah selesai dibuat dan akan diuji coba, yaitu pengujian berdasarkan struktur data tampilan program dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.6. Sistematika Penulisan

Sistematika penulisan yang diuraikan dalam penyusunan skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Bab ini berisi tentang Latar Belakang, Rumusan Masalah, Tujuan, Batasan Masalah, Metodologi Penelitian dan Sistematikan Penulisan.

BAB II : LANDASAN TEORI

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

BAB III : PERANCANGAN DAN DESAIN APLIKASI

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

BAB IV : UJI COBA APLIKASI

Bab ini berisi tentang pengujian aplikasi pada hardware yang sudah dibuat.

BAB V : PENUTUP

Pada bab ini berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II

LANDASAN TEORI

2.1 GPS (*Global Positioning System*)

Global Positioning System (GPS) adalah suatu sistem navigasi yang memanfaatkan satelit. Penerima GPS memperoleh sinyal dari beberapa satelit yang mengorbit bumi. Satelit yang mengitari bumi pada orbit pendek ini terdiri dari 24 susunan satelit, dengan 21 satelit aktif dan 3 buah satelit sebagai cadangan. Dengan susunan orbit tertentu, maka satelit GPS bisa diterima diseluruh permukaan bumi dengan penampakan antara 4 sampai 8 buah satelit. GPS dapat memberi informasi posisi dan waktu dengan ketelitian sangat tinggi.

Nama lengkapnya NAVSTAR GPS (*Navigational Satellite Timing and Ranging Global Positioning System*), ada juga yang mengartikan “*Navigation System Using Timing and Ranging*”. Dari perbedaan singkatan itu, orang lebih mengenalnya dengan nama GPS. GPS mulai diaktifkan pada tanggal 17 juli 1995.

2.2 Google Map

Google Maps adalah layanan gratis yang diberikan oleh *Google* dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat di gunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. dapat menambahkan fitur *Google Maps* dalam *web* yang telah di buat atau pada blog yang berbayar maupun gratis sekalipun dengan *Google Maps API*. *Google Maps API* adalah suatu *library* yang berbentuk *JavaScript*.

Cara membuat *Google Maps* untuk ditampilkan pada suatu web atau blog sangat mudah hanya dengan membutuhkan pengetahuan mengenai **HTML** serta **JavaScript**, serta koneksi Internet yang sangat stabil. Dengan menggunakan *Google Maps API*, dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, hanya membuat suatu data sedangkan peta yang

akan ditampilkan adalah milik *Google* sehingga tidak dipusingkan dengan membuat peta suatu lokasi, bahkan dunia. Dalam pembuatan *program Google Map API* menggunakan urutan sebagai berikut :

1. Memasukkan *Maps API JavaScript* ke dalam HTML kita.
2. Membuat *element div* dengan nama *map_canvas* untuk menampilkan peta.
3. Membuat beberapa *objek literal* untuk menyimpan properti-properti pada peta.
4. Menuliskan fungsi *JavaScript* untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan *event onload*.

Pada *Google Maps API* terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

1. ROADMAP, ini yang saya pilih, 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 diatasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota).

2.3 Android

Android adalah sistem operasi untuk ponsel yang berbasis *Linux*. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

2.3.1 Versi Android

sampai saat ini google telah meluncurkan banyak versi. Peluncuran versi-versi terbaik android lebih diutamakan untuk perkembangan dan perbaikan dari versi sebelumnya, versi Android tersebut adalah :

1. Android Beta

Dirilis pada tanggal 5 November 2007, kemudian SDK (*Software Development Kit*) dirilis tanggal 12 November 2007.

2. Android 1.0

Versi ini merupakan android komersial yang pertama kali dan dirilis pada tanggal 23 November 2008. Perangkat yang pertama kali menggunakan versi ini adalah HTC Dream.

3. Android 1.1

Versi ini hanya dirilis untuk HTC Dream saja pada tanggal 9 februari 2009 dengan perbaikan bug dan beberapa fitur tambahan.

4. Android 1.5 (*cupcake*)

Pada pertengahan mei 2009, google kembali merilis telepon seluler dengan menggunakan android dan SDK (*Software Development Kit*) dengan versi 1.5 (*cupcake*). Terdapat beberapa pembaharuan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke youtube dan gambar ke picasa langsung dari telepon, dukungan bluetooth A2DP, kemampuan terhubung secara otomatis ke headset bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

5. Android 1.6 (*Donut*)

Donut (versi 1.6) dirilis pada september dengan menampilkan proses pencarian yang lebih baik di banding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang diintegrasikan; CDMA/EVDO, 802.1x, VPN, *Gestares*, dan *Text-to-speech engine*.

6. Android 2.0/2.1 (*Eclair*)

Pada 3 desember 2009 kembali diluncurkan ponsel Android versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan hardware, perubahan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML 5, daftar kontak yang baru, dukungan flash untuk camera 3,2 MP, *digital Zoom* dan *bluetooth 2.1*.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka pada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi android adalah *shazam*, *backgrounds*, dan *weatherBug*. Sistem operasi android dalam situs internet juga dianggap penting untuk menciptakan aplikasi *Android* asli, contohnya oleh *Myspace* dan *Facebook*.

7. Android 2.2.x(*Frozen Yoghurt/Froyo*)

Pada 20 mei 2010, android versi 2.2 (*Froyo*) diluncurkan. Perubahan-perubahan umumnya terhadap umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript *engine* yang dipakai *Google Chrome* yang mempercepat kemampuan rendering pada *browser*, pemasangan aplikasi dalam SD card, kemampuan WIFI Hostpot portabel, dan kemampuan *auto update* dalam aplikasi Android Market.

8. Android 2.3.x (*Gingerbread*)

Pada 6 desember 2010, android versi 2.3 (*Gingerbread*) di luncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan fungsi *copy paste*, layar antarmuka (*user interface*)

didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb, equalization, headphone virtualization*, dan *bass boost*) dan dukungan jumlah kamera yang lebih dari satu.

9. Android 3.x (*Honeycomb*)

Android *Honeycomb* di rancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. *User interface* pada *Honeycomb* juga berbeda karena sudah di desain untuk tablet. *Honeycomb* juga mendukung multi prosesor dan akselerasi perangkat keras (*hardware*) untuk grafis.

10. Android 4.0.x (*Ice Cream Sandwich*)

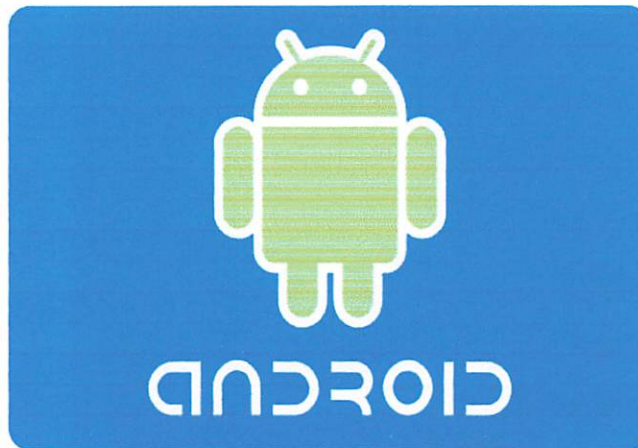
Di umumkan pada tanggal 19 oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan geografi, mencari *email* secara *offline*, dan berbagai informasi.

11. Android 4.1 (*Jelly Bean*)

Android Jelly Bean diluncurkan oleh google pada tanggal 27 juni 2012 pada acara Google I/O Conference. Versi ini berbasiskan linux kernel 3.0.1.

Google Now juga menjadi bagian yang harus di perbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu lintas, atau hasil pertandingan olahraga. OS Android *Jelly Bean* 4.1 muncul pertama kali dalam produk tablet Asus, yakni *Google Nexus 7*.

2.4 Sistem Operasi Android



Gambar 2.1 Sistem Operasi Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Antarmuka pengguna Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar. Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa

pemrograman Java. Pada bulan Oktober 2012, ada sekitar 700.000 aplikasi yang tersedia untuk Android, dan sekitar 25 juta aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survey pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi seluler.

Faktor-faktor di atas telah memberikan kontribusi terhadap perkembangan Android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan Symbian pada tahun 2010. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Akibatnya, meskipun pada awalnya sistem operasi ini dirancang khusus untuk telepon pintar dan tablet, Android juga dikembangkan menjadi aplikasi tambahan di televisi, konsol permainan, kamera digital, dan perangkat elektronik lainnya. Sifat Android yang terbuka telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain.

Pada November 2013, Android menguasai pangsa pasar telepon pintar global, yang dipimpin oleh produk-produk Samsung, dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat Android berbeda dengan beragam versi. Keberhasilan sistem operasi ini juga menjadikannya sebagai target litigasi paten "perang telepon pintar" antar perusahaan-perusahaan teknologi. Hingga bulan Mei 2013, total 900 juta perangkat Android telah diaktifkan di seluruh dunia, dan 48 miliar aplikasi telah dipasang dari Google Play. Pada tanggal 3 September 2013, 1 miliar perangkat Android telah diaktifkan.

2.4.1 Fitur Antarmuka Android

Antarmuka pengguna pada Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, misalnya menggesek (*swiping*), mengetuk (*tapping*), dan mencubit (*pinching*), untuk memanipulasi obyek di layar. Masukan pengguna direspon dengan cepat dan juga tersedia antarmuka sentuh layaknya permukaan air, seringkali menggunakan kemampuan getaran perangkat untuk memberikan umpan balik haptik kepada pengguna. Perangkat keras internal seperti akselerometer, giroskop, dan sensor proksimitas digunakan oleh beberapa aplikasi untuk merespon tindakan pengguna, misalnya untuk menyesuaikan posisi layar dari potret ke lanskap, tergantung pada bagaimana perangkat diposisikan, atau memungkinkan pengguna untuk mengarahkan kendaraan saat bermain balapan dengan memutar perangkat sebagai simulasi kendali setir.

Ketika dihidupkan, perangkat Android akan memuat pada layar depan (*homescreen*), yakni navigasi utama dan pusat informasi pada perangkat, serupa dengan desktop pada komputer pribadi. Layar depan Android biasanya terdiri dari ikon aplikasi dan *widget*; ikon aplikasi berfungsi untuk menjalankan aplikasi terkait, sedangkan widget menampilkan konten secara langsung dan terbaru otomatis, misalnya prakiraan cuaca, kotak masuk surel pengguna, atau menampilkan tiker berita secara langsung dari layar depan. Layar depan bisa terdiri dari beberapa halaman, pengguna dapat menggeser bolak balik antara satu halaman ke halaman lainnya, yang memungkinkan pengguna Android untuk mengatur tampilan perangkat sesuai dengan selera mereka. Beberapa aplikasi pihak ketiga yang tersedia di *Google Play* dan di toko aplikasi lainnya secara ekstensif mampu mengatur kembali tema layar depan Android, dan bahkan bisa meniru tampilan sistem operasi lain, misalnya *Windows Phone*. Kebanyakan produsen telepon seluler dan operator nirkabel menyesuaikan tampilan perangkat Android buatan mereka untuk membedakannya dari pesaing mereka.

Di bagian atas layar terdapat status bar, yang menampilkan informasi tentang perangkat dan konektivitasnya. Status bar ini bisa "ditarik" ke bawah untuk membuka layar notifikasi yang menampilkan informasi penting atau pembaruan aplikasi, misalnya surel diterima atau SMS masuk, dengan cara tidak

mengganggu kegiatan pengguna pada perangkat. Pada versi awal Android, layar notifikasi ini bisa digunakan untuk membuka aplikasi yang relevan, namun setelah diperbarui, fungsi ini semakin disempurnakan, misalnya kemampuan untuk memanggil kembali nomor telepon dari notifikasi panggilan tak terjawab tanpa harus membuka aplikasi utama. Notifikasi ini akan tetap ada sampai pengguna melihatnya, atau dihapus dan di nonaktifkan oleh pengguna.

2.4.2 Fitur Aplikasi Android

Android memungkinkan penggunanya untuk memasang aplikasi pihak ketiga, baik yang diperoleh dari toko aplikasi seperti Google Play, Amazon Appstore, ataupun dengan mengunduh dan memasang berkas APK dari situs pihak ketiga. Di Google Play, pengguna bisa menjelajah, mengunduh, dan memperbarui aplikasi yang diterbitkan oleh Google dan pengembang pihak ketiga, sesuai dengan persyaratan kompatibilitas Google. Google Play akan menyaring daftar aplikasi yang tersedia berdasarkan kompatibilitasnya dengan perangkat pengguna, dan pengembang dapat membatasi aplikasi ciptaan mereka bagi operator atau negara tertentu untuk alasan bisnis. Pembelian aplikasi yang tidak sesuai dengan keinginan pengguna dapat dikembalikan dalam waktu 15 menit setelah pengunduhan. Beberapa operator seluler juga menawarkan tagihan langsung untuk pembelian aplikasi di Google Play dengan cara menambahkan harga pembelian aplikasi pada tagihan bulanan pengguna. Pada bulan September 2012, ada lebih dari 675.000 aplikasi yang tersedia untuk Android, dan perkiraan jumlah aplikasi yang diunduh dari Play Store adalah 25 miliar.

Aplikasi Android dikembangkan dalam bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK). SDK ini terdiri dari seperangkat perkakas pengembangan,[61] termasuk debugger, perpustakaan perangkat lunak, emulator handset yang berbasis QEMU, dokumentasi, kode sampel, dan tutorial. Didukung secara resmi oleh lingkungan pengembangan terpadu (IDE) Eclipse, yang menggunakan plugin Android Development Tools (ADT). Perkakas pengembangan lain yang tersedia di antaranya adalah Native Development Kit untuk aplikasi atau ekstensi dalam C atau C++, Google App

Inventor, lingkungan visual untuk pemrogram pemula, dan berbagai kerangka kerja aplikasi web seluler lintas platform.

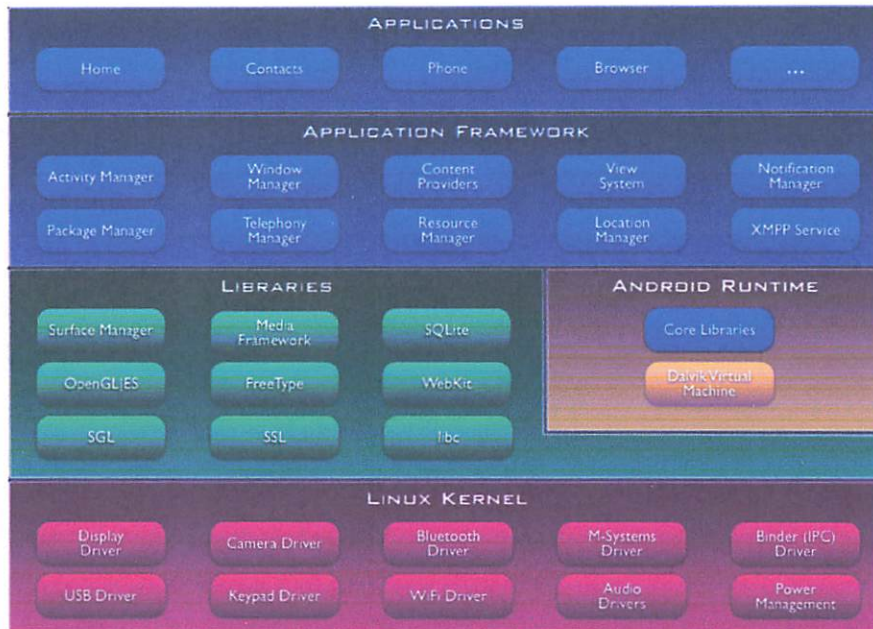
2.4.3 Fitur Pengelolaan Memori Android

Android dikembangkan secara pribadi oleh Google sampai perubahan terbaru dan pembaruan siap untuk dirilis, dan informasi mengenai kode sumber juga mulai diungkapkan kepada publik. Kode sumber ini hanya akan berjalan tanpa modifikasi pada perangkat tertentu, biasanya pada seri Nexus. Ada binari tersendiri yang disediakan oleh produsen agar Android bisa beroperasi.

Logo Android yang berwarna hijau awalnya dirancang untuk Google pada tahun 2007 oleh desainer grafis Irina Blok. Tim desain ditugaskan dengan sebuah proyek untuk membuat sebuah ikon universal yang mudah dikenali dengan menyertakan ikon robot secara spesifik dalam desain akhir. Setelah sejumlah perkembangan desain yang didasarkan pada tema-tema fiksi ilmiah dan film luar angkasa, tim akhirnya mendapat inspirasi dari simbol manusia yang terdapat di pintu toilet, dan memodifikasi bentuknya menjadi bentuk robot. Karena Android adalah perangkat lunak sumber terbuka, disepakati bahwa logo tersebut juga harus terbuka, dan sejak diluncurkan, logo hijau tersebut telah didesain ulang kembali dalam berbagai variasi yang tak terhitung jumlahnya.

2.5 Arsitektur Android

Android terdiri dari susunan arsitektur atau lapisan dari komponen Platform Android. Pada gambar 2.2 dapat dilihat lapisan-lapisan dari arsitektur Android.



Sumber: http://elinux.org/Android_Architecture

Gambar 2.2 Arsitektur Platform Android

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Berikut ini susunan dari lapisan-lapisan tersebut jika di lihat dari lapisan dasar hingga lapisan teratas.

1. Linux Kernel

Tumpukan paling bawah pada arsitektur Android ini adalah kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun sistem Android, yang mencakup memory management, security setting, power management, dan beberapa driver hardware. Kernel berperan sebagai abstraction layer antara hardware dan keseluruhan software. Sebagai contoh, HTC GI dilengkapi dengan kamera. Kernel Android terdapat driver kamera

yang memungkinkan pengguna mengirimkan perintah kepada hardware kamera.

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (Interprocess Communication) untuk mengatur aplikasi dan lubang keamanan.

2. Android Run Time

Lapisan setelah Kernel Linux adalah Android Runtime yang berisi Core Libraries dan Dalvik. **Core Libraries** mencakup serangkaian inti library Java, artinya Android menyertakan satu set library-library dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada library-library dasar bahasa pemrograman Java. **Dalvik** adalah Java Virtual Machine yang memberi kekuatan pada sistem Android. Dalvik VM ini di optimalkan untuk telepon seluler. Setiap aplikasi yang berjalan pada Android berjalan pada processnya sendiri, dengan instance dari Dalvik Virtual Machine. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi Virtual Machine dengan efisien. Dalvik VM dapat mengeksekusi file dengan format Dalvik Executable (.dex) yang telah dioptimasi untuk menggunakan minimal memory footprint. Virtual Machine ini register-based, dan menjalankan class-class yang dicompile menggunakan compiler Java yang kemudian ditransformasi menjadi format .dex menggunakan “dx” tool yang telah disertakan. Dalvik Virtual Machine (VM) menggunakan kernel Linux untuk menjalankan fungsi-fungsi seperti threading dan low-level memory management.

3. Libraries

Bertempat di level yang sama dengan Android Runtime adalah Libraries. Android menyertakan satu set library-library dalam bahasa C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem Android. Kemampuan ini dapat diakses oleh programmer melewati Android application framework. Sebagai contoh Android mendukung pemutaran format audio, video, dan gambar. Berikut ini beberapa core library tersebut :

1) Sistem C library

Diturunkan dari implementasi *standard C* sistem *library (libc)* milik BSD, dioptimasi untuk piranti embedded berbasis *Linux*.

2) Media Libraries

Berdasarkan *PacketVideo's OpenCORE*; *library-library* ini mendukung *playback* dan *recording* dari berbagai format audio and video populer, meliputi MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.

3) Surface Manager

Mengatur akses pada display dan lapisan *composites 2D and 3D graphic* dari berbagai aplikasi.

4) LibWebCore

Web browser engine modern yang *mensupport* Android browser maupun *embeddable web view*.

5) SGL

the underlying 2D graphics engine.

6) 3D libraries

implementasi berdasarkan OpenGL ES 1.0 APIs; *library* ini menggunakan *hardware 3D acceleration* dan *highly optimized 3D software rasterizer*.

7) FreeType

bitmap dan vector font rendering

8) SQLite

Relational database engine yang *powerful* dan ringan tersedia untuk semua aplikasi.

Library-library tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan *Native Development Toolkit (NDK)*.

4. Application Framework

Lapisan selanjutnya adalah application framework, yang mencakup program untuk mengatur fungsi-fungsi dasar smartphone. Application Framework merupakan serangkaian tool dasar seperti alokasi resource smartphone, aplikasi telepon, pergantian antar – proses atau program, dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada tool-tool dasar tersebut, dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks.

Programmer mendapatkan akses penuh untuk memanfaatkan API-API (*Android Protocol Interface*) yang juga digunakan core applications. Arsitektur aplikasi didesain untuk menyederhanakan pemakaian kembali komponen-komponen, setiap aplikasi dapat menunjukkan kemampuannya dan aplikasi lain dapat memakai kemampuan tersebut. Mekanisme yang sama.

2.6 Android Software Development Kit (Android SDK)

Android SDK merupakan *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode dan tutorial. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan *plugin Android Development Tools (ADT)*, dengan ini pengembang dapat menggunakan teks editor untuk mengedit file Java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasi Android dan pengendalian perangkat Android.

2.7 ADT (Android Development Tools)

Android Development Tools (ADT) adalah *plug-in* untuk *Eclipse* IDE yang dirancang untuk memberikan lingkungan yang *powerfull* dan terpadu untuk membangun aplikasi Android.

ADT memperluas kemampuan *Eclipse* untuk mempercepat dalam pembuatan project Android baru, membuat UI aplikasi, menambahkan komponen berdasarkan *Android Framework API*, debug aplikasi menggunakan Android tool SDK, dan bahkan ekspor unsigned .apk file dalam rangka untuk mendistribusikan aplikasi. Mengembangkan aplikasi Android di Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi Android. Dengan dipandu *set-up project*, serta integrasi peralatan, custom XML editor, dan debug panel output, ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.

2.8 Eclipse Integrated Development Environment (Eclipse IDE)

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

1. Multi-platform

Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

2. Multilanguage

Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

3. Multi-role

Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya.

2.9 XAMPP



Xampp adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.

Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, mySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari web resminya

2.9.1 Bagian penting XAMPP

Mengenal bagian XAMPP yang biasa digunakan pada umumnya :

1. Htdoc adalah folder tempat meletakkan berkas – berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain.
2. phpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada dikomputer. Untuk membukanya, buka browser lalu ketikkan alamat *http://localhost/phpMyAdmin*, maka akan muncul halaman *phpMyAdmin*.
3. Kontrol panel yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

2.9.2 Komponen

XAMPP terdiri dari :

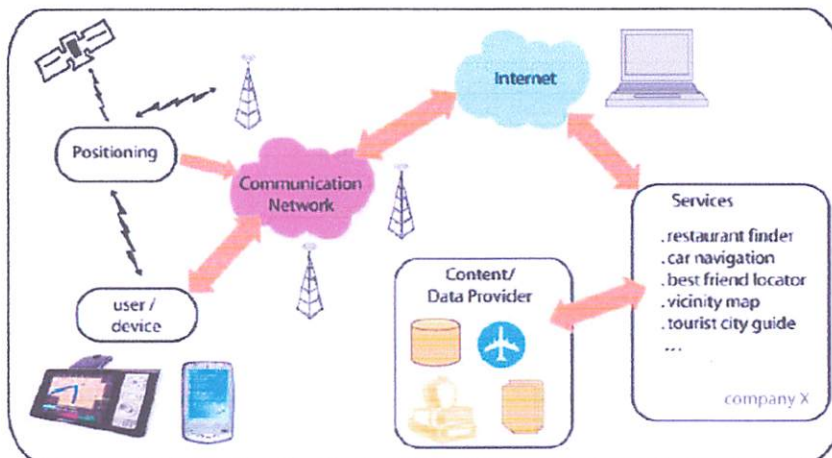
1. Apache
2. mySQL
3. PHP
4. phpMyAdmin
5. Filezilla FTP Server
6. Tomcat (dengan mod_proxy_ajp sebagai konektor)

2.10 Location Based Service

2.10.1 Pengertian LBS

Layanan berbasis lokasi (*location based service/LBS*) adalah layanan informasi yang mengikuti kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak melalui jaringan telekomunikasi bergerak.

LBS mengarah pada aplikasi aplikasi dan layanan-layanan yang menyediakan informasi lokasi mengenai pengguna *mobile*, untuk menyediakan sebuah layanan atau informasi pada pengguna.



Gambar 2.3 Teknologi LBS

Dalam gambar 2.3, di tunjukan ilustrasi teknologi LBS. LBS dapat di gambarkan sebagai layanan yang berada pada pertemuan tiga teknologi yaitu : Sistem informasi Geografi (SIG), *internet service* dan *mobile service*.

Selain dari sisi pengguna, LBS juga bisa di lihat dari sisi layanan yang di berikan. Dari sisi layanan yang berikan, LBS bisa di bagi menjadi dua yaitu *reactive* LBS dan *proactive* LBS. *Reactive* LBS adalah layanan yang hanya aktif jika aksi yang di lakukan pengguna. Sedangkan *Proactive* LBS merupakan layanan yang akan selalu memberi informasi kepada pengguna walaupun pengguna tidak melakukan permintaan terhadap layanan.

2.10.2 Komponen LBS

Terdapat 4 komponen penting dalam teknologi layanan berbasis lokasi yang mendukung dalam pengguna teknologi ini, seperti yang ditunjukan dalam gambar 2.4



Gambar 2.4 komponen LBS

Seperti di gambarkan dalam gambar 2.4, LBS terdiri dari beberapa komponen, antara lain :

1. Perangkat Mobile

Sebuah perangkat pengguna yang berfungsi untuk meminta informasi yang dibutuhkan. Hasilnya bisa di tampilkan dalam bentuk suara, gambar, teks dan lainnya. Perangkat yang mungkin seperti PDA, Smartphone, Laptop. Handphone dan alat navigasi dalam kendaraan.

2. Jaringan komunikasi

Jaringan komunikasi menjadi media mengirimkan data pengguna dan meminta layanan dari terminal perangkat *mobile* ke *provider* pemberi layanan dan kemudian informasi yang di peroleh di kirimkan kembali ke pengguna.

3. Komponen pengambil posisi

Location based service membutuhkan data posisi pengguna yang berupa koordinat yang di peroleh dari perangkat pengambil posisi seperti GPS untuk mendapatkan informasi dari layanan *location based service* dari *provider*. Selain GPS media pengambil posisi bisa di tentukan oleh *cell tower* dan kombinasi antara GPS dan *cell tower* (aGPS).

4. Provider Layanan dan Aplikasi

Provider informasi ini memberikan informasi berdasarkan permintaan pengguna berdasarkan informasi lokasi atau informasi yang di kirim oleh pengguna. *Provider* melakukan proses komputasi seperti menemukan rute perjalanan, informasi tempat terdekat, mencari ke database *external* seperti *yellow pages* atau *google API* untuk menghasilkan informasi yang di butuhkan oleh pengguna.

5. Data dan Provider Konten

Penyedia layanan biasanya tidak menyimpan dan mempertahankan semua informasi yang dapat di minta oleh pengguna. Oleh karena itu basis data geografi dan data informasi lokasi biasanya di minta atau berasal dari otoritas penyedia peta.

Untuk dapat meminta informasi yang user, maka terlebih dahulu di perlukan informasi mengenai lokasi user. Lokasi user dapat di ketahui dengan dua cara, dengan menggunakan alat navigasi yang sudah terpasang langsung pada mobile device yang di gunakan (contohnya ponsel dengan GPS), atau melalui jaringan komunikasi yang telah ada.

Kemudian melalui jaringan komunikasi, informasi tersebut di teruskan ke jaringan internet untuk selanjutnya di hubungkan dengan *service and aplication provider*. *Service and aplication provider* selanjutnya mengakses content provider untuk mendapatkan data yang di miliki untuk memproses informasi yang di inginkan user. Selanjutnya service and aplikasi provider mengirim informasi yang telah di olah melalui jaringan internet dan jaringan komunikasi. Pada akhirnya user dapat menerima informasi yang di inginkan. Ada lima kegiatan yang di dasari kebutuhan user terhadap informasi geografis :

1. Mengetahui di mana lokasi user atau objek tertentu berada(*Locating*).
2. Mencari lokasi seseorang, suatu objek atau kejadian pada suatu daerah(*searching*).
3. Menunjukkan arah untuk menuju suatu lokasi (*navigating*).
4. Mengenali suatu objek pada suatu lokasi dan menampilkan data yang berkaitan dengan objek tersebut.
5. Mencari dan memeriksa suatu kejadian pada atau sekitar suatu lokasi (*checking*).

2.10.3 Dua Unsur Utama LBS

Dua unsur utama dari *Location Based Service* :

1. *Location Manager (API Maps)*

Menyediakan *tools/resource* untuk LBS, *application programming interface* (API) maps menyediakan fasilitas untuk menampilkan tampilan *satelit*, *street*(jalan), maupun gabungannya. Paket ini berada pada *com.google.android.maps*.

2. *Location Providers (API Location)*

Menyediakan teknologi pencarian lokasi yang di gunakan oleh *device/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/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

2.10.4 Jenis LBS

Secara garis besar, *Location Based Service* di bagi menjadi dua jenis :

1. **Pull Service**

Layanan di berikan berdasarkan permintaan dari pelanggan akan kebutuhan suatu informasi, misalnya seperti mengakses suatu web pada jaringan internet.

2. **Push Service**

Layanan ini di berikan langsung oleh *service provider* tanpa menunggu permintaan dari pelanggan.

2.10.5 Kelebihan dan Kekurangan LBS

kelebihan LBS, tetap berfungsi bila berada di dalam gedung dan pengaruh medan elektromagnetik lain yang tidak terlalu besar. Sedangkan, kekurangan LBS adalah jangkauan area yang sangat bergantung pada jangkauan seluler.

BAB III

ANALISA DAN PERANCANGAN SISTEM

Bab ini berisi penjelasan tentang desain sistem keseluruhan yang akan dibuat. Tahap-tahap dalam pengerjaan penelitian sesuai dengan metode penelitian. Berikut ini adalah tahap Analisa dan Perancangan sistem, diantaranya :

1. Analisa sistem
2. Deskripsi umum
3. Spesifikasi Aplikasi
4. Perancangan sistem
5. Analisa kebutuhan fungsional
6. Analisa kebutuhan Non fungsional
7. Perhitungan LBS
8. Data ATM
9. Eclipse
10. Map activity
11. Desain Tampilan
12. Flowchart

3.1 Analisa Sistem

Analisa dan perancangan sistem merupakan tahapan selanjutnya yang dilakukan penulis untuk memecahkan masalah dari penelitian ini secara sistematis, sehingga solusi dapat diimplementasikan lebih jelas. Pada penelitian kali ini digunakan *Flowchart* untuk penggambaran analisa dan rancangan sistem.

3.1.1 Diskripsi Umum sistem

Aplikasi yang dibuat adalah aplikasi *mobile* untuk memberikan informasi pencarian mesin ATM diantaranya ATM BCA, ATM BRI, ATM BNI dan ATM MANDIRI di Kota Madya Malang dalam bentuk aplikasi *smartphone*. Dalam aplikasi ini *user* dapat memilih ATM terdekat yang akan di tuju.

3.1.2 Spesifikasi Aplikasi

Dalam pembuatan aplikasi ini terdapat beberapa menu yang tersedia yang memiliki kemampuan sebagai berikut :

1. Kategori ATM : menu ini berguna untuk menampilkan semua ATM yang akan di pilih user. Jika user memilih ATM BCA, maka akan menampilkan semua ATM BCA. Jika user memilih ATM BRI, maka akan menampilkan semua ATM BRI. Jika user memilih ATM BNI, maka akan menampilkan semua ATM BNI. Jika user memilih ATM MANDIRI, maka akan menampilkan semua ATM MANDIRI.
2. Pilih ATM Tujuan : menu ini terdapat di menu ATM BCA, ATM BRI, ATM BNI dan ATM MANDIRI yang berfungsi untuk menampilkan lokasi ATM terdekat yang akan di tuju.
3. Pilih ATM : menu ini terdapat di menu ATM yang di pilih user untuk menentukan ATM mana yang akan di pilih.
4. Lihat ATM : setelah *user* menentukan ATM, maka akan menampilkan lokasi ATM yang di pilih.

3.2 Perancangan Sistem

Dari permasalahan sebelumnya penulis menawarkan sebuah solusi yang dapat memecahkan permasalahan tersebut dengan lebih mudah. Yaitu aplikasi layanan lokasi atm menggunakan *LBS*. Aplikasi ini akan membantu *user* menemukan informasi atm yang ada di sekitarnya dengan cara yang baru yaitu menggunakan GPS yang ada di perangkat *smartphone*. Disini penulis menggunakan teknologi *LBS* yang akan menampilkan daftar tempat atm berdasarkan lokasi GPS pengguna melalui *google map* dan juga rute menuju ketempat tujuan.

Selain itu aplikasi ini akan dilengkapi informasi peta digital yang dapat membantu pengguna menemukan posisinya dan atm di sekitar lebih jelas, selain itu juga dilengkapi navigasi yang dapat menuntun pengguna sampai ke lokasi atm. Selain GPS, perangkat juga memerlukan koneksi *internet* agar bisa terhubung ke *Google Map server* sebagai *API provider* peta digital.

Terakhir pengguna dapat melihat semua daftar atm yang ada di sistem beserta informasi detail tentang atm. Fitur ini adalah fitur tambahan yang dapat digunakan pengguna untuk melihat keseluruhan atm yang ada di sistem sekaligus dalam satu daftar. Sehingga aplikasi ini terdiri dari 3 bagian, yaitu : *Menu*, *map view*, dan *list atm*.

3.2.1 Analisa Kebutuhan Perangkat Keras

perangkat keras yang dibutuhkan untuk membangun sistem ini adalah:

1. CPU dengan spesifikasi :
 - a. Processor intel® Core™ 2 Duo T6400
 - b. Memory: 1GB DDR2 PC5300, up to 4GB
 - c. Hard Drive: 320GB SATA
 - d. Optical Drive: DVDRW±RW
 - e. Chipset: Intel® 40GL Chipset
 - f. Display: 14.1" WXGA BrightView TFT with 1280x800 Max Resolution
2. Koneksi internet
3. *Smartphone* Android dengan spesifikasi
 - a. Android 2.3. sampai dengan Android 4.4
 - b. RAM minimal 128 Mb
 - c. ROM minimal 512 Mb
 - d. GPS support.
 - e. Koneksi GPRS.
- b. Analisa kebutuhan perangkat lunak

perangkat lunak yang dibutuhkan untuk sistem ini adalah:

 1. *Java Development*, untuk kompilasi kode program
 2. *Java Runtime Edition*, sebagai *platform* untuk menjalankan sistem
 3. *Dreamweaver*, sebagai perangkat lunak yang digunakan untuk membangun *web service*.
 4. *Eclipse*, sebagai perangkat lunak yang digunakan untuk membangun aplikasi *client* pada Android.
 5. Android SDK, sebagai pengembang aplikasi Android.

6. *Android development tool*, sebagai *plugin* Android pada *eclipse*.
7. *Android virtual device*, sebagai *simulator* untuk menjalankan sistem Android.

3.2.2 Analisa Kebutuhan Fungsional

Analisa kebutuhan fungsional adalah analisa kebutuhan sistem yang berdampak langsung terhadap sistem. Menentukan fungsional sistem yang dapat digunakan oleh pengguna. Kebutuhan fungsional dapat di lihat pada tabel 3.1

Tabel 3.1 Deskripsi Kebutuhan Fungsional

No.	Deskripsi Kebutuhan Fungsional
1.	Menyimpan data atm dan koordinat (<i>latitude</i> dan <i>longitude</i>) lokasi atm
2.	Menampilkan data atm dan jarak dari lokasi pengguna dengan metode <i>LBS</i>
3.	Menampilkan posisi atm dan posisi pengguna dalam tampilan peta digital
4.	Menampilkan navigasi atau rute jalan dari lokasi pengguna menuju atm
5.	Menampilkan informasi atm secara keseluruhan berupa daftar

3.2.3 Analisa Kebutuhan Non Fungsional

Analisa kebutuhan non fungsional menentukan kualitas secara keseluruhan suatu sistem. Menempatkan batasan pada program yang sedang dikembangkan, proses pengembangannya, dan menentukan batasan-batasan eksternal yang harus dipenuhi oleh program. Kebutuhan non fungsional dapat di lihat pada tabel 3.2

Tabel 3.2 Kebutuhan non fungsional

No.	Parameter	Deskripsi
1.	Availability	Sistem dapat bekerja memberikan layanan selama 24 jam sehari, 7 hari seminggu
2.	Performance	Kecepatan waktu akses tidak lebih dari 30 detik
3.	Usability	Antarmuka aplikasi berupa GUI menggunakan bahasa inggris
4.	Portability	Digunakan pada lingkungan sistem operasi android

3.2.4 Perhitungan LBS

Berikut ini merupakan rancangan perhitungan LBS dengan menggunakan rumus berikut ini :

$$a = \sin^2(\Delta \text{Lat} / 2) + \cos(\text{Lat}_1) * \cos(\text{Lat}_2) * \sin^2(\Delta \text{Lng} / 2)$$

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

$$R = 6.371.000 \text{ meter}$$

$$d = R \cdot c$$

Keterangan Rumus:

d = jarak yang dicari

R = nilai konstanta koordinat bumi dalam meter

c dan a = rumus bantuan perhitungan jarak

3.2.5 Data ATM

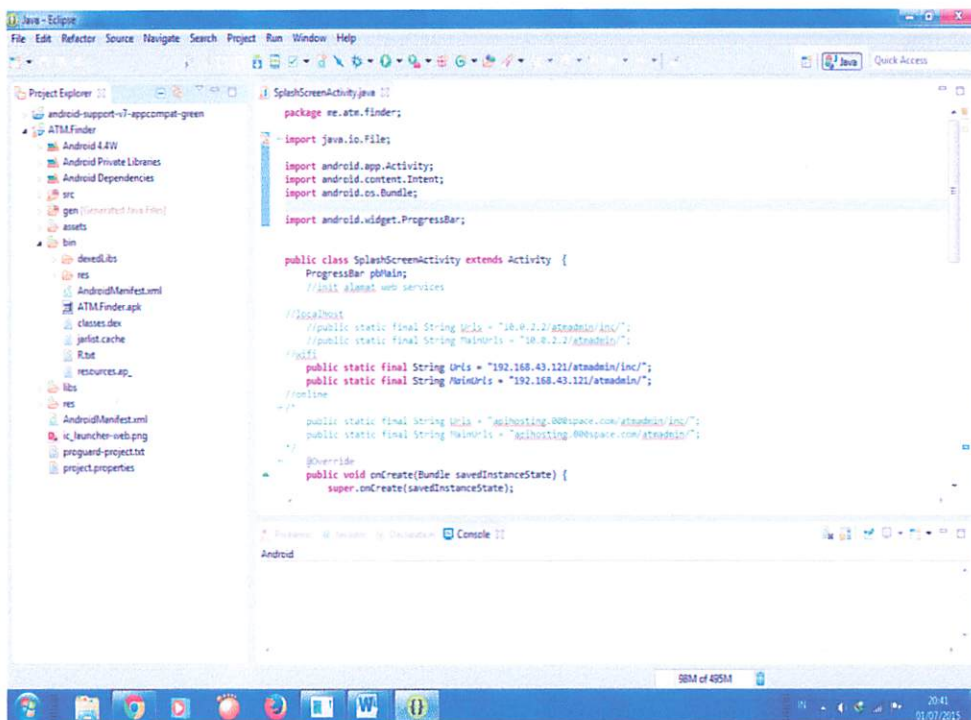
Data ATM digunakan untuk mengetahui lokasi ATM yang akan di tampilkan dalam peta. Untuk mengetahui lokasi ATM tersebut di butuhkan titik koordinat berupa garis lintang (Latitude) dan garis bujur (Longitude). Data ATM dapat di lihat pada tabel 3.3

Tabel 3.3 Data ATM

No.	Bank	Nama ATM	Alamat	Latitude (°)	Longitude (°)
1	BCA	RS Islam Malang	Jl. MT. Haryono No.139 Malang Utara/Lowokwaru	- 7.940436	112.608670
2	BCA	Universitas Brawijaya	Jl. Veteran, malang	- 7.953666	112.615833
3	BCA	Pondok Borobudur Indah	Jl. Blimbing Megah indah 1, malang	- 7.936437	112.650087
4	BCA	STIE Malangkucewara	Jl. T. Candi Kalasan, Malang	- 7.940884	112.622727
5	BCA	RS Permata Bunda	Jl. Soekarno Hatta no.75 ,malang	- 7.938844	112.608638
6	BCA	Carrefour Express	Jl. A.Yani no.2A, Malang	- 7.944518	112.641205
7	BCA	Malang Town Square	Jl. Veteran No.2A, Malang	- 7.956968	112.618481
8	BCA	Dieng Plaza	Jl. Raya Langsep no.2 Malang	- 7.973728	112.612436
9	BCA	Univ. Widyagama	Jl. Borobudur No.12 Malang Utara/Blimbing	- 7.939366	112.665660
10	BCA	Hotel Kartika Graha	Jl. Jaksa Agung Suprpto no.17 ,Malang	- 7.973247	112.630286

3.3 Eclipse

Eclipse yang di gunakan untuk membuat aplikasi ini merupakan eclipse juna. Sebelum membuat aplikasi pada android perlu di lakukan pengaturan terhadap terhadap eclipse terlebih dahulu, yaitu dengan menginstal ADT yang merupakan plug-in dari dari eclipse agar bisa membuat project Android. Project pembuatan aplikasi dapat di lihat pada gambar 3.1

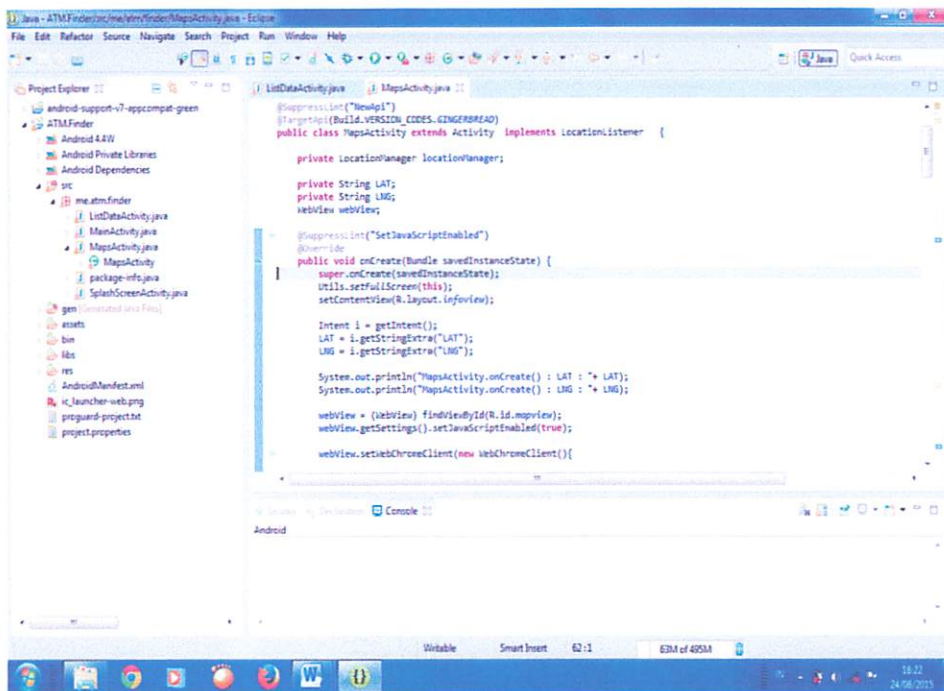


Gambar 3.1 project aplikasi android

Dari gambar 3.1 di tampilkan potongan code program untuk pembuatan project aplikasi android. Code program secara detail dapat di lihat pada halaman Lampiran.

3.3.1 MapActivity

Untuk menampilkan Google Maps dalam aplikasi Android Anda, modifikasi file main.xml yang terletak di folder res / layout. Anda harus menggunakan elemen `<com.google.android.maps.MapView>` untuk menampilkan Google Maps dalam aktivitas Anda. menggunakan elemen `<RelativeLayout>` posisi peta dalam activity. Project pembuatan Map Activity dapat di lihat pada gambar 3.2



Gambar 3.2 project MapActivity

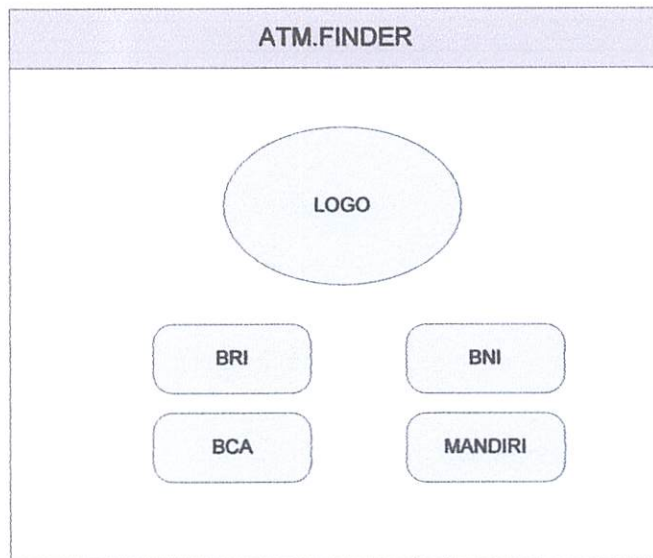
Dari gambar 3.2 di tampilkan potongan code program untuk pembuatan project MapActivity. Code program secara detail dapat di lihat pada halaman Lampiran.

3.4 Desain Tampilan

Secara garis besar, desain tampilan untuk aplikasi ini terdiri atas beberapa bagian, yaitu :

1. Antar Muka Menu

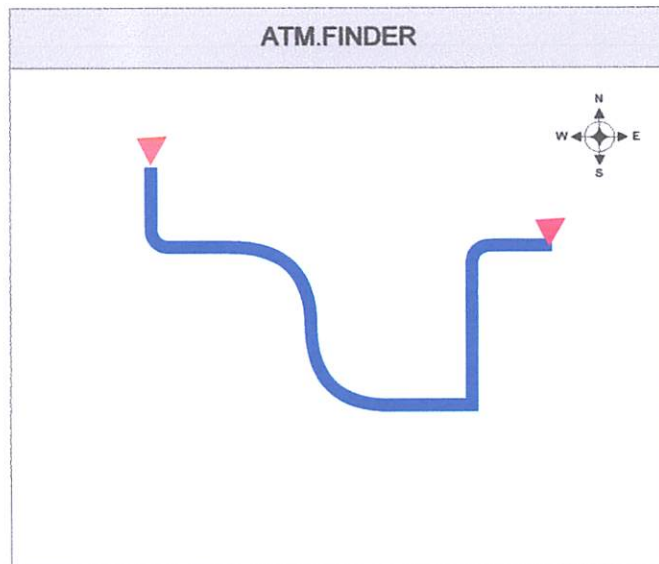
Antar Muka menu berfungsi untuk memilih menu pada aplikasi pencarian ATM. Terdapat 4 pilihan menu, yaitu menu ATM BCA, ATM BRI, ATM BNI dan ATM MANDIRI. Desain *form* menu utama diperlihatkan pada gambar 3.3



Gambar 3.3 Antar Muka Menu

2. Antar Muka Map View

Antar Muka Map View berfungsi menampilkan rancangan antarmuka *map view*. Pengguna dapat menemukan atm di sekitarnya tidak hanya dengan *LBS* namun juga dapat menggunakan bantuan peta digital. Melalui fitur ini pengguna dapat memilih 3 tampilan peta, yaitu normal, *satelite*, dan *terrain*. Dengan *map view* pengguna dapat melihat lokasi atm pada marker dari atas permukaan peta. Desain Antar muka map view dapat di lihat pada gambar 3.4



Gambar 3.4 Antar muka Map View

3. Antar Muka List

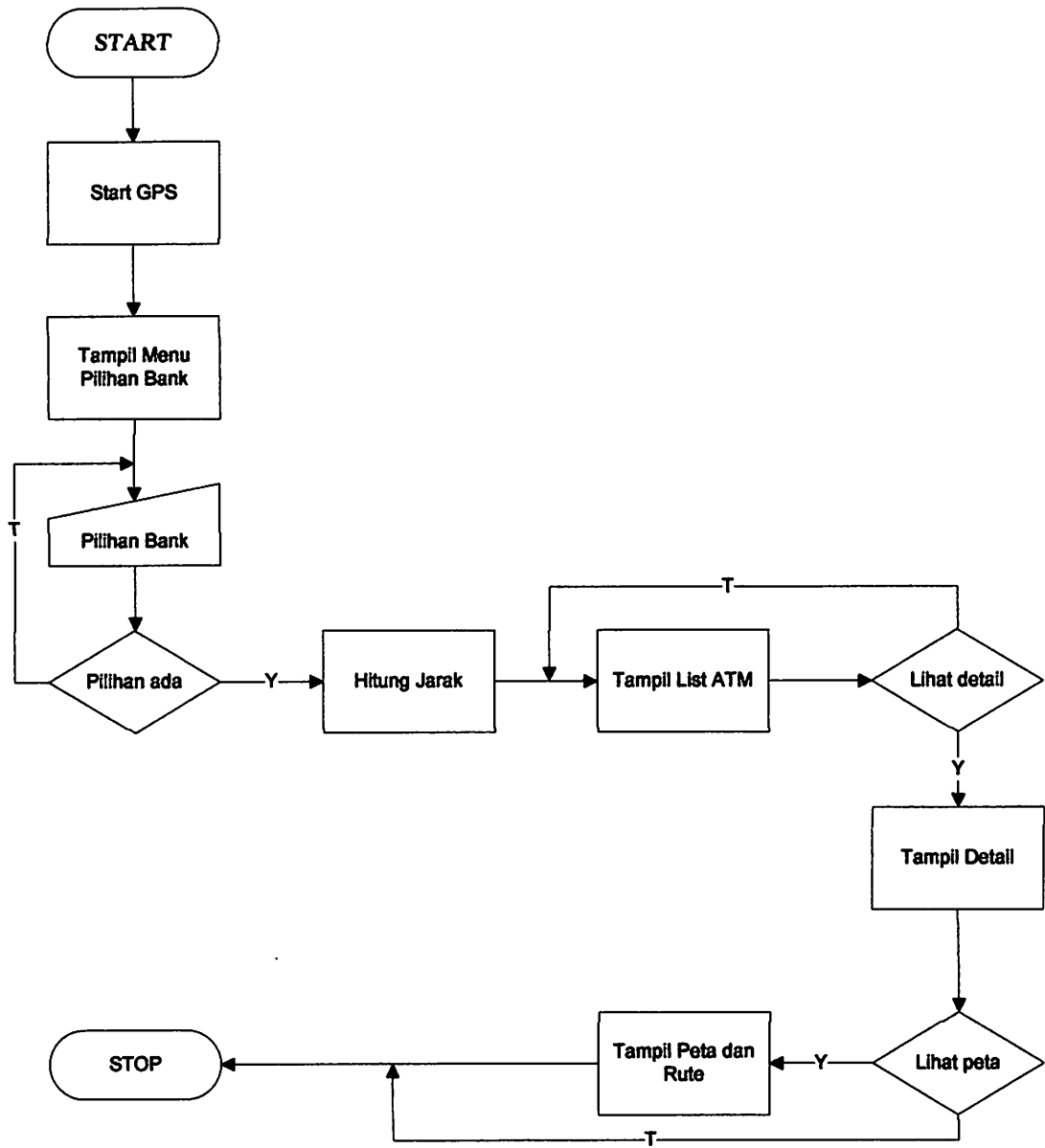
Antar Muka list menampilkan antarmuka list yang membantu pengguna menemukan atm disekitarnya dalam bentuk daftar. Desain Antar Muka List dapat di lihat pada gambar 3.5

ATM.FINDER
ATM BCA A
ATM BCA B
ATM BCA C
ATM BCA D
ATM BCA E
ATM BCA F
ATM BCA G

Gambar 3.5 Antar Muka List

3.4 Flowchart

Alur Aplikasi secara lengkap dapat digambarkan dengan flowchart pada gambar 3.6



Gambar 3.6 Flowchart Aplikasi Pencarian ATM

Keterangan :

1. Start GPS untuk mulai menggunakan satelit GPS.
2. Tampil menu pilihan bank untuk menampilkan menu ATM yang terdapat pada aplikasi.
3. Pilihan Bank untuk menampilkan list ATM Bank yang telah dipilih.
4. Hitung jarak untuk menampilkan List ATM terdekat dalam Radius 1 Km.
5. Tampil Detail untuk mengetahui lokasi/tempat ATM yang sudah di pilih.
6. Tampil Peta untuk mengetahui rute ATM yang sudah di pilih.

BAB IV

IMPLEMENTASI DAN PENGUJIAN APLIKASI

Bab ini menjelaskan tentang hasil yang diperoleh berdasarkan pengujian yang telah dilakukan. Tujuan dari dilakukannya pengujian adalah mengetahui apakah Aplikasi Pencarian ATM sudah berjalan dengan baik atau tidak. Berikut tahap-tahap pengujian yang dilakukan:

1. Implementasi sistem
2. Database Server
3. Perhitungan LBS
4. Proses penginstalan aplikasi pada Smartphone Android
5. Pengujian Aplikasi

4.1 Implementasi Sistem

Tahap implementasi pengembangan perangkat lunak merupakan proses pengubahan spesifikasi sistem menjadi sistem yang dapat di jalankan. Tahap ini merupakan lanjutan dari proses perancangan sesuai dengan spesifikasi dan desain sistem.

Aplikasi Pencarian ATM di Malang berbasis *Android* ini menggunakan Eclipse sebagai editor dan emulator dalam pengembangan sistem operasi Android, kemudian di-compile agar dapat dijalankan pada smartphone Android.

4.2 Persiapan Pembuatan Aplikasi

4.2.1 Database Server

Database merupakan suatu pengorganisasi data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun memanipulasi data, seperti menambah, mengedit, mengupdate data dan menghapus data dari sebuah table maupun beberapa table. Database server dapat dilihat pada gambar 4.1



Gambar 4.1 Database server

4.2.2 Perhitungan LBS

Berikut ini merupakan rancangan perhitungan LBS dengan menggunakan rumus berikut ini :

$$a = \sin^2(\Delta \text{Lat} / 2) + \cos(\text{Lat}_1) * \cos(\text{Lat}_2) * \sin^2(\Delta \text{Lng} / 2)$$

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

$$R = 6.371.000 \text{ meter}$$

$$d = R * c$$

Keterangan Rumus:

d = jarak yang dicari

R = nilai konstanta koordinat bumi dalam meter

c dan a = rumus bantuan perhitungan jarak

Pada contoh perhitungan jarak ini digunakan 4 contoh data ATM di antaranya ATM BCA, BNI, BRI dan MANDIRI. Data ATM dapat di lihat pada tabel 4.1

Tabel 4.1 Data ATM

No.	Bank	Nama ATM	Alamat	Latitude (°)	Longitude (°)
1	BCA	BCA indomart	Jl. Borobudur no.80, malang	-7.939412	112.636112
2	BNI	Univ. Widyagama	Jl. Borobudur no.12, malang	-7.939366	112.636015
3	BRI	Univ. Widyagama	Jl. Borobudur no.12, malang	-7.939366	112.636015
4	MANDIRI	Univ. Widyagama	Jl. Borobudur no.12, malang	-7.939366	112.636015

Dari data pada table 4.1 dilakukan perhitungan dengan lokasi sekarang di Perum. Pondok indah Borobudur Blimbing dengan titik koordinat :

-7.934848	112.639393
-----------	------------

dengan menggunakan rumus perhitungan jarak sehingga diperoleh hasil data jarak yang diperlihatkan pada Table 4.2

Tabel 4.2 Hasil Perhitungan Data ATM

No.	Bank	Nama ATM	Jarak Tempu pada Aplikasi (KM)
1	BCA	BCA indomart	0,8
2	BNI	Univ. Widyagama	0,7
3	BRI	Univ. Widyagama	0,7
4	MANDIRI	Univ. Widyagama	0,7

Hasil perhitungan jarak masing-masing ATM dapat di lihat pada gambar 4.2



BCA



BNI



BRI



MANDIRI

Gambar 4.2 hasil perhitungan

Dari hasil perhitungan pada gambar 4.2, di lakukan pengujian hasil secara nyata (*realtime*) dengan menggunakan speedometer motor untuk mengetahui jarak sebenarnya. Hasil pengujian dapat di lihat pada tabel 4.2

Tabel 4.3 Hasil Perhitungan Data ATM

No.	Bank	Nama ATM	Jarak Tempu pada Aplikasi (KM)	Jarak Tempuh Nyata (KM)	Error (KM)
1	BCA	BCA indomart	0,8	0,8	0
2	BNI	Univ. Widyagama	0,7	0,8	0,1
3	BRI	Univ. Widyagama	0,7	0,8	0,1
4	MANDIRI	Univ. Widyagama	0,7	0,8	0,1

4.3 Pengujian

Pengujian dilakukan pada Smartphone Android, dengan sistem yang digunakan adalah Android versi 4.4.2 (KitKat)

Pengujian ini dilakukan untuk mengetahui tingkat keberhasilan dari tugas akhir yang telah direncanakan sebelumnya pada bab perancangan.

Smartphone yang digunakan pada saat pengujian adalah sebagai berikut:

Tipe	: LG L Fino
Operation Sistem	: Android OS, v4.4.2 (KitKat)
Design	: Full Touch Screen Capacitive
CPU	: Dual-core 1.2 GHz Cortex-A7
Dimension	: 127.5 x 67.9 x 11.9 mm
Resolution	: 480 x 800 pixels, 4.5 inches
Bluetooth	: v4.0, A2DP
Wifi	: Wi-Fi 802.11 b/g/n, hotspot
GPRS	: class 12
EDGE	: class 12
3G	: HSDPA 900/2100, HSPA 21.1/5.76 Mbps

4.4 Pengujian Sistem Pada smartphone Android

Pengujian terhadap program yang telah selesai, dilakukan dengan cara melakukan uji coba terhadap proses yang ada secara satu persatu. Hal ini diperlukan untuk dapat melihat apakah program mampu berjalan dengan baik. Uji coba proses yang dilakukan akan difokuskan pada pengujian proses fungsionalitas aplikasi pada smartphone Android.

Pada perangkat lunak aplikasi Pencarian ATM di Malang berbasis Android yang dibuat, pengembangan aplikasi menggunakan editor Eclipse IDE kemudian di-compile ke dalam smartphone Android.

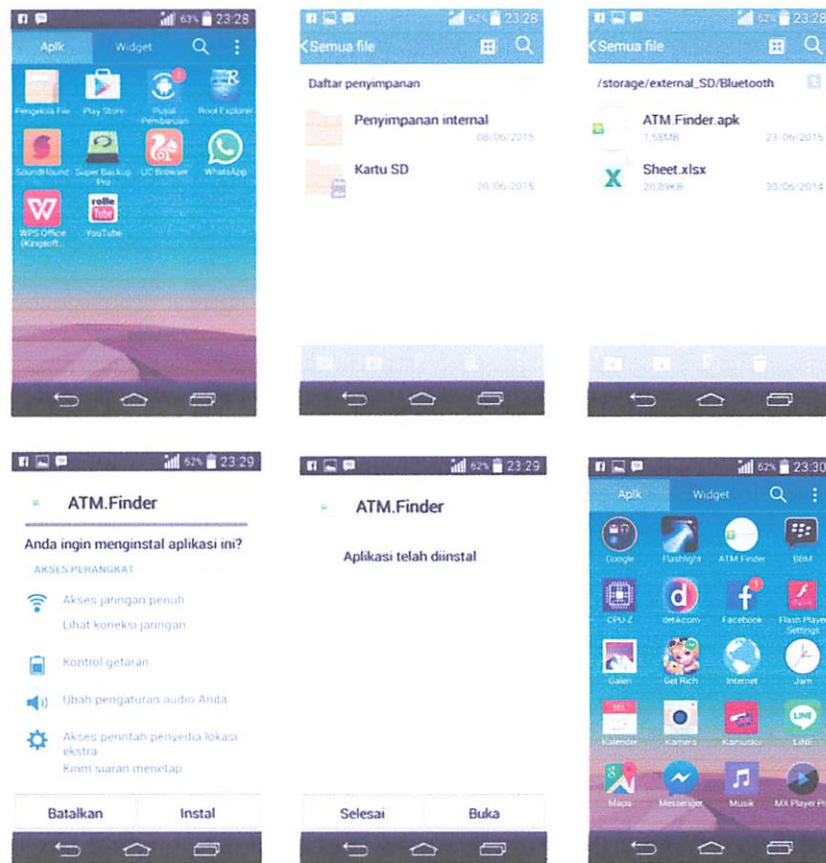
4.4.1 Pengujian Pada LG L Fino

Sistem operasi Android telah mengalami banyak pembaharuan sejak dirilis atau peluncuran awalnya, LG L Fino menggunakan Android versi 4.4.2 atau Kitkat, Android versi 4.4.2 ini merupakan pengembangan dari Android 4.0 atau Ice Cream Sandwich. Pengembangan dilakukan dengan pengoptimalan pada perangkat keras juga peningkatan pada perangkat lunaknya, diantaranya menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan geografi, mencari email secara offline, dan berbagai informasi.

Aplikasi Pencarian ATM di Malang Berbasis Android diuji pada LG L fino. Pada saat diuji coba aplikasi berjalan dengan lancar.

4.4.2 Proses Penginstalan

Tahap awal yang dilakukan sebelum diinstall yaitu, file *apk* terlebih dahulu kita copy kan ke perangkat handphone, caranya bisa lewat langsung dari PC ataupun lewat Bluetooth. Pada gambar 4.8 berikut adalah langkah-langkah dalam proses pengistalannya.



Gambar 4.3 Proses Penginstalan Aplikasi pada Handphone LG L Fino

Pada gambar 4.3 di jelaskan secara urut proses penginstalannya yang dimulai dari pembukaan Pengelola file kemudian folder Bluetooth dan buka file *apk*, kemudian akan keluar sebuah *form* keterangan konfirmasi instalasi, selanjutnya saat kita pilih Install, maka akan muncul lagi sebuah *form* aplikasi telah berhasil diinstall, maka aplikasipun sudah siap untuk dijalankan.

4.5 Pengujian Aplikasi

4.5.1 Pengujian Pada Menu Utama

Menu utama pada aplikasi pencarian ATM di malang ini terdapat fitur-fitur yang dapat dipilih oleh user, dapat di lihat pada gambar 4.4



Gambar 4.4 menu utama aplikasi

Dari gambar 4.4 ditunjukkan bahwa menu utama pada aplikasi pencarian ATM telah berhasil dijalankan pada smartphone Android.

4.5.2 Pengujian Aplikasi di ITN 2

1. ATM BNI

a. Pengujian Atm BNI pada titik koordinat -7.910786, 112.635534. hasil list yang di tampilkan dalam radius 1km. Dapat di lihat pada gambar 4.5



Gambar 4.5 List ATM BNI

b. pengujian lokasi ATM

Pada pengujian lokasi atm, jalur yang di gambarkan pada peta berupa garis berwarna biru dengan tujuan lokasi atm yang di pilih pada menu sebelumnya, sedangkan marker warna hijau adalah posisi pengguna dan marker warna merah adalah atm yg di tuju. Lebih jelasnya dapat di lihat pada gambar 4.6



Gambar 4.6 Rute Atm BNI

2. ATM BRI

a. Pengujian Atm BRI pada titik koordinat -7.910786, 112.635534. hasil list yang di tampilkan dalam radius 1km. Dapat di lihat pada gambar 4.7



Gambar 4.7 list ATM BRI

b. pengujian lokasi ATM

Pada pengujian lokasi atm, jalur yang di gambarkan pada peta berupa garis berwarna biru dengan tujuan lokasi atm yang di pilih pada menu sebelumnya, sedangkan marker warna hijau adalah posisi pengguna dan marker warna merah adalah atm yg di tuju. Lebih jelasnya dapat di lihat pada gambar 4.8



Gambar 4.8 Rute ATM BRI

4.5.3 Pengujian Aplikasi di Perum. Borobudur

1. ATM BCA

a. Pengujian Atm BCA pada titik koordinat -7.934855, 112.639329. hasil list yang di tampilkan dalam radius 1km. Dapat di lihat pada gambar 4.9



Gambar 4.9 List ATM BCA

b. pengujian lokasi ATM

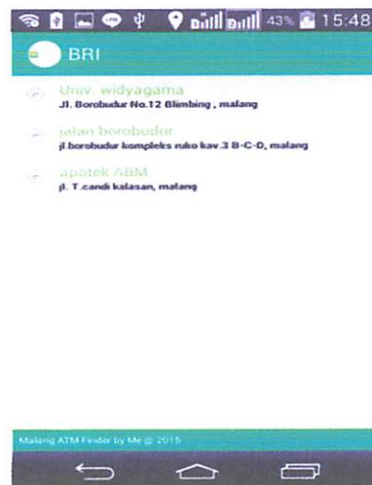
Pada pengujian lokasi atm, jalur yang di gambarkan pada peta berupa garis berwarna biru dengan tujuan lokasi atm yang di pilih pada menu sebelumnya, sedangkan marker warna hijau adalah posisi pengguna dan marker warna merah adalah atm yg di tuju. Lebih jelasnya dapat di lihat pada gambar 4.10



Gambar 4.10 Rute ATM BCA

2. ATM BRI

a. Pengujian Atm BRI pada titik koordinat -7.934855, 112.639329. hasil list yang di tampilkan dalam radius 1km. Dapat di lihat pada gambar 4.11



Gambar 4.11 List ATM BRI

b. pengujian lokasi ATM

Pada pengujian lokasi atm, jalur yang di gambarkan pada peta berupa garis berwarna biru dengan tujuan lokasi atm yang di pilih pada menu sebelumnya, sedangkan marker warna hijau adalah posisi pengguna dan marker warna merah adalah atm yg di tuju. Lebih jelasnya dapat di lihat pada gambar 4.10



Gambar 4.12 Rute ATM BRI

4.5.4 Pengujian Aplikasi di Soekarno Hatta

1. ATM MANDIRI

a. Pengujian Atm MANDIRI pada titik koordinat -7.942007, 112.621844. hasil list yang di tampilkan dalam radius 1km. Dapat di lihat pada gambar 4.13



Gambar 4.13 List ATM Mandiri

b. pengujian lokasi ATM

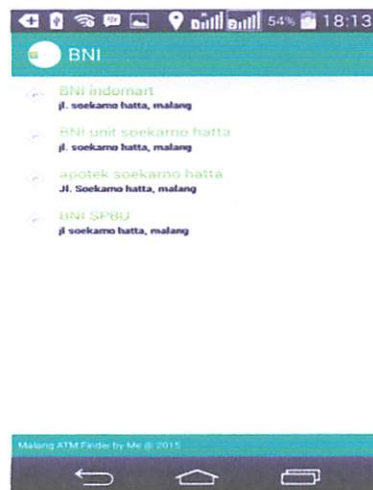
Pada pengujian lokasi atm, jalur yang di gambarkan pada peta berupa garis berwarna biru dengan tujuan lokasi atm yang di pilih pada menu sebelumnya, sedangkan marker warna hijau adalah posisi pengguna dan marker warna merah adalah atm yg di tuju. Lebih jelasnya dapat di lihat pada gambar 4.14



Gambar 4.14 Rute ATM MANDIRI

2. ATM BNI

a. Pengujian Atm BNI pada titik koordinat -7.942007, 112.621844. hasil list yang di tampilkan dalam radius 1km. Dapat di lihat pada gambar 4.15



Gambar 4.15 List ATM BNI

b. pengujian lokasi ATM

Pada pengujian lokasi atm, jalur yang di gambarkan pada peta berupa garis berwarna biru dengan tujuan lokasi atm yang di pilih pada menu sebelumnya, sedangkan marker warna hijau adalah posisi pengguna dan marker warna merah adalah atm yg di tuju. Lebih jelasnya dapat di lihat pada gambar 4.16



Gambar 4.16 Rute ATM BNI

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan segala rangkaian perencanaan dan pembuatan desain serta pembuatan Aplikasi Pencarian ATM di Malang Berbasis Android yang berjalan pada mobile berbasis Android ini, maka dalam laporan skripsi ini penulis mengambil kesimpulan sebagai berikut :

1. ATM yang di tampilkan hanya dalam *radius* 1 km.
2. ATM yang di tampilkan hanya data ATM yang telah masuk kedalam *database*.
3. Aplikasi berhasil mengambil nilai *Latitude* dan *Longitude* sebagai akurasi lokasi pengguna berada.
4. Kecepatan untuk mengakses informasi dari *database server* bergantung pada kecepatan akses *Internet*.
5. Aplikasi diterapkan pada *Smartphone Android*.

5.2 Saran

Aplikasi Pencarian ATM di Malang Berbasis *Android* ini dapat dikembangkan lebih jauh lagi karena dalam pembuatannya masih jauh dari sempurna dan masih banyak kekurangan. Adapun saran yang dapat dikemukakan agar aplikasi ini bisa berfungsi dengan lebih optimal adalah :

1. Dapat ditambahkan menu ATM dari Bank lain.
2. Dapat di lakukan pengembangan untuk beberapa OS lainnya, seperti *iOS* dan *Windows Phone*.
3. Dapat di lakukan Pengembangan dengan menggunakan Metode Lain.

DAFTAR PUSTAKA

- [1] Micheal Siregar, Ivan ST MT. *Membongkar Source Code Berbagai Aplikasi Android*, Penerbit Gava Media, Yogyakarta 2011.
- [2] Safaat, Nazaruddin H. *Pemrograman aplikasi mobile smartphone dan tablet Pc berbasis android*, Penerbit Informatika, Bandung 2012.
- [3] Mulyadi, ST., 2010, *Membuat Aplikasi untuk Android*. Multimedia Center Publishing, Yogyakarta.
- [4] Paul Deitel, Harvey Deitel, Abbey Deitel. *Android, How to Program*, Penerbit Deitel & Associates Inc, San Francisco 2013.
- [5] Prashasta, Eddy, *Sistem informasi Geografis*, cetakan kedua, CV. Informatika Bandung, 2005.
- [6] Wulan Liyanti. 2012. "Manfaat ATM". <http://blogspot.com/2012/manfaat-dalam-penggunaan-atm.html>, di akses 04 oktober 2014.
- [7] Vogella. 2014. "Source Code Android". <http://vogella.de.net>, diakses tanggal 04 oktober 2014.
- [8] Wikipedia. 2014. "Definisi Andoid". <http://id.wikipedia.org/wiki/android>, di akses tanggal 04 oktober 2014.
- [9] Wikipedia. 2014. "Dokumentasi Map android". [http:// id.wikipedia.org/wiki/maps/documentation/android/?hl=id](http://id.wikipedia.org/wiki/maps/documentation/android/?hl=id), di akses 04 oktober 2014.
- [10] Wikipedia. 2014. "Dokumentasi Map View". [http:// id.wikipedia.org/wiki/maps/documentation/android/hello-mapview/?hl=id](http://id.wikipedia.org/wiki/maps/documentation/android/hello-mapview/?hl=id), diakses 04 desember 2014.
- [11] Wikipedia. 2014. "Belajar membuat aplikasi Sederhana Android ". <http://id.wikipedia.org/wiki/android.com/index.html>, di akses tanggal 04 oktober 2014.

Lampiran

nerlomsJ

Code Progam List Data Activity

```
package me.atm.finder;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map.Entry;

import javax.net.ssl.HttpURLConnection;

import org.apache.http.HttpException;
import org.apache.http.util.EncodingUtils;

import me.atm.finder.MapsActivity.GettingLocation;
import me.atm.finder.R.id;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.base.org.sun.java.Utills;
```

```

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.services.ApiParam;
import android.util.services.ApiServices;
import android.view.View;
import android.webkit.ConsoleMessage;
import android.webkit.WebChromeClient;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class ListDataActivity extends Activity implements LocationListener {
    //init var
    ApiServices apiServices ;
    ListView lvData;

```

Utils u ;

StringBuilder FinalResult = new StringBuilder();

boolean isFinish = false;

private LocationManager locationManager;

WebView webView ;

@Override

protected void onCreate(Bundle savedInstanceState) {

 // TODO Auto-generated method stub

 super.onCreate(savedInstanceState);

 setContentView(R.layout.list_data);

 u = new Utils(getApplicationContext(), this);

 lvData = (ListView) findViewById(id.lvData);

 //amil param

 Intent i = getIntent();

 ID = i.getStringExtra("ID");

 NAME = i.getStringExtra("NAME");

 setTitle(NAME);

 locationManager = getSystemService(Context.LOCATION_SERVICE); (LocationManager)

 new GettingLocation().execute("");

 StrictMode.ThreadPolicy policy = new
 StrictMode.ThreadPolicy.Builder().permitAll().build();

 StrictMode.setThreadPolicy(policy);

 webView =(WebView) findViewById(R.id.webView1);

```

webView.setWebChromeClient(new WebChromeClient() {
    public boolean onConsoleMessage(ConsoleMessage cmsg)
    {
        if (cmsg.message().startsWith("TAG"))
        {
            String                searchResult                =
cmsg.message().substring(22).replace("</body>", "");
            FinalResult = new StringBuilder( searchResult);
            isFinish = true;
            System.out.println("Hasil : "+searchResult);
            return true;
        }

        return false;
    }
});

```

```

webView.setWebViewClient(new WebViewClient() {
    @Override
    public void onPageStarted(WebView view, String url, Bitmap favicon) {
        // TODO Auto-generated method stub
        super.onPageStarted(view, url, favicon);
        isFinish = false;

        FinalResult= new StringBuilder("");
    }

    public void onPageFinished(WebView view, String address)
    {

```

```
view.loadUrl("javascript:console.log('TAG'+document.getElementsByTagName('html')[0].innerHTML);");
```

```
}
```

```
@Override
```

```
public void onReceivedError(WebView view, int errorCode,
```

```
String description, String failingUrl) {
```

```
// TODO Auto-generated method stub
```

```
super.onReceivedError(view, errorCode, description,  
failingUrl);
```

```
FinalResult= new StringBuilder("-") ;
```

```
}
```

```
});
```

```
webView.getSettings().setLoadsImagesAutomatically(false);
```

```
webView.getSettings().setAppCacheEnabled(true);
```

```
webView.getSettings().setJavaScriptEnabled(true);
```

```
String postData = "CID="+u.getApiID()+"&" +
```

```
"ID="+ID+"&" +
```

```
"NAME="+NAME ;
```

```
webView.loadUrl(  
"http://" + SplashScreenActivity.UrIs + "api.php?type=get_tempat_by_jenis&&" + (  
postData)); //, EncodingUtils.getBytes(postData, "BASE64"));
```

```
}
```

```

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub

    finish();
    super.onBackPressed();
}

boolean isFirst = true;

class GettingLocation extends AsyncTask<String, Void, Void> {

private ProgressDialog Dialog = new ProgressDialog(ListDataActivity.this);

protected void onPreExecute() {
    //progress
    Dialog.setMessage(" Loading GPS Location .");
    Dialog.show();
    Dialog.setCancelable(true) ;
    locString = "";

    locationManager.requestLocationUpdates ( LocationManager.GPS_PROVIDER,
        1000, // 1 sec
        10, ListDataActivity.this);
}

protected Void doInBackground(String... urls) {
    while (locString.trim().equals("")) {

```

```

    }
    return null;
}

```

```

protected void onPostExecute(Void unused) {

```

```

    Dialog.dismiss();
}

```

```

}

```

```

String locString = "";

```

```

/***** Called after each 3 sec *****/

```

```

double lastLNG=0,lastLAT=0;

```

```

private String Kata;

```

```

public void onLocationChanged(Location location) {

```

```

    String str = "Lokasi Anda : \n-Latitude: "+location.getLatitude()+"\n-Longitude: "+location.getLongitude();

```

```

    locString =location.getLatitude()+","+location.getLongitude();

```

```

    if (isFirst){

```

```

        String lat2 = String.valueOf(location.getLatitude()).replace(",", ".");

```

```

        String lng2 = String.valueOf(location.getLongitude()).replace(",", ".");

```

```

        isFirst = false;

```

```

        Toast.makeText(getApplicationContext(), str,
        Toast.LENGTH_LONG).show();

```

```

        this.setPosition(location.getLatitude(), location.getLongitude());

        new GettingData().execute("");

    }

}

    public double CalculationByDistance(double startLatitude, double startLongitude,
double endLatitude, double endLongitude ) {

        float [] dist = new float[1];

        Location.distanceBetween(startLatitude, startLongitude, endLatitude,
endLongitude, dist);

        System.out.println( " Dist => startLatitude : "+ startLatitude +" " +
                                ", startLongitude : "+ startLongitude+"", endLatitude :
"+endLatitude+", endLongitude : "+endLongitude+", dist : " + dist[0]);

        return dist[0];

    }

    public void onProviderDisabled(String provider) {

        /***** Called when User off Gps *****/

        Toast.makeText(getBaseContext(), "Gps turned off ",
Toast.LENGTH_LONG).show();

    }

    public void onProviderEnabled(String provider) {

        /***** Called when User on Gps *****/

        Toast.makeText(getBaseContext(), "Gps turned on ",
Toast.LENGTH_LONG).show();

```

```
}
```

```
public void onStatusChanged(String provider, int status, Bundle extras) {
```

```
    // TODO Auto-generated method stub
```

```
}
```

```
    public class GettingData extends AsyncTask<String, Void, Void> {
```

```
        private ProgressDialog Dialog = new ProgressDialog(ListDataActivity.this);
```

```
        protected void onPreExecute() {
```

```
            //progress
```

```
            Dialog.setMessage(" Loading data.");
```

```
            Dialog.show();
```

```
            Dialog.setCancelable(true) ;
```

```
            apiServices = new ApiServices(ListDataActivity.this, SplashScreenActivity.Urls);
```

```
        }
```

```
        int Tunggu = 0;
```

```
        protected Void doInBackground(String... urls) {
```

```
            while ( FinalResult.toString().equals("") ) {
```

```
            }
```

```
            //if (!FinalResult.equals(""))
```

```

        return null;
    }

    protected void onPostExecute(Void unused) {

        Dialog.dismiss();

        //tampil data ke list
        // u.Alert(FinalResult.toString());

        getData();

        if (dataListTempat != null )
        {

            ListAdapter adapter = new
SimpleAdapter(ListDataActivity.this, dataListTempat , R.layout.list_row,

                new String[] { "C", "D" },

                new int[] { R.id.title, R.id.detail });

            lvData.setAdapter(adapter);

            lvData.setOnItemClickListener(new
OnItemClickListener() {

                public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {

                    gotoItem(position);

                }

            });

        }

    }

} else{

    u.Alert( "Koneksi dengan server gagal..");

    // finish();

```

```

    }

}

}

    ArrayList<HashMap<String, String>> dataListTempat ;
    private String ID;
    private String NAME;
    double Curlat=0;
    double Curlon=0;
    public void setPosition(double lat, double lon) {
        Curlat = lat;
        Curlon = lon;
    }

    public String performPostCall(String requestURL,
    HashMap<String, String> postDataParams) {

    URL url;
    String response = "";
    try {
        url = new URL(requestURL);

        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setReadTimeout(15000);
        conn.setConnectTimeout(15000);
        conn.setRequestMethod("POST");
        conn.setDoInput(true);
        conn.setDoOutput(true);

```

```

OutputStream os = conn.getOutputStream();
BufferedWriter writer = new BufferedWriter(
    new OutputStreamWriter(os, "UTF-8"));
writer.write(getPostDataString(postDataParams));

writer.flush();
writer.close();
os.close();
int responseCode=conn.getResponseCode();

if (responseCode == HttpURLConnection.HTTP_OK) {
    String line;
    BufferedReader br=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
    while ((line=br.readLine()) != null) {
        response+=line;
    }

}
else {
    response="";

    throw new HttpException(responseCode+"");
}
} catch (Exception e) {
    e.printStackTrace();
}

return response;

```

```
}
```

```
private String getPostDataString(HashMap<String, String> params) throws  
UnsupportedEncodingException{
```

```
    StringBuilder result = new StringBuilder();
```

```
    boolean first = true;
```

```
    for(Entry<String, String> entry : params.entrySet()){
```

```
        if (first)
```

```
            first = false;
```

```
        else
```

```
            result.append("&");
```

```
            result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
```

```
            result.append("=");
```

```
            result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
```

```
        }
```

```
    return result.toString();
```

```
}
```

```
private boolean getData() {
```

```
    StringBuffer sb = new StringBuffer();
```

```
    String response="";
```

```
    try {
```

```
        //parameter yang dikirim ke api
```

```
        response = FinalResult.toString();
```

```
        System.out.println("Response : " + response);
```

```

        if (response.equals( "{\"items\":[ ] } ")) {

            u.Alert("Data tidak ditemukan..");

            //return false;
        }

        //parsing json dari api
        dataListTempat= apiServices.getResult(response, 9);

        lvData = (ListView) findViewById(id.lvData);

        ArrayList<HashMap<String, String>> TmpdataList = new
        ArrayList<HashMap<String, String>>();

        TmpdataList
        =
        new
        ArrayList<HashMap<String,String>>(dataListTempat);

        dataListTempat= new ArrayList<HashMap<String, String>>();
        ArrayList<Double> JarakPos = new ArrayList<Double>();
        for (int i = 0; i < TmpdataList.size(); i++) {
            if (Curlat!=0 )
            {
                double d = Double.parseDouble(
                TmpdataList.get(i).get("E").replace(",","."));
                double c = Double.parseDouble(
                TmpdataList.get(i).get("F").replace(",","."));
                double jarak
                =CalculationByDistance(Curlat, Curlon, d, c);
                System.out.println("ATM
                :"+TmpdataList.get(i).get("C"));
                System.out.println("LAT :"+d);
                System.out.println("LNG :"+c);
                System.out.println("Dist :"+jarak+"
                Meter ");
                JarakPos.add(jarak);
            }
        }
    }
}

```

```

dataListTempat.add(TmpdataList.get(i)) ;

}

if (Curlat!=0 ) {
    double[] fixJarak = new double[JarakPos.size()];
    if (dataListTempat.size()!=0){
        double minDis = JarakPos.get(0);
        double[] tmpJarak = new
double[JarakPos.size()];

        for (int i = 0; i < JarakPos.size(); i++) {
            tmpJarak[i] =JarakPos.get(i);
            fixJarak[i] =JarakPos.get(i);
        }
        double a=0;
        double b=0;
        double abTemp=0;
        HashMap<String, String> tmpDataList;
        for(int i=0;i<tmpJarak.length-1;i++)
        {
            for(int j=1;j<tmpJarak.length;j++)
            {
                a= tmpJarak[j] ;
                b= tmpJarak[j-1] ;
                if(a<b)
                {

                    abTemp=tmpJarak[j];

```

```

        tmpDataList = dataListTempat.get(j);
        dataListTempat.set(j, dataListTempat.get(j-1));
        dataListTempat.set(j-1, tmpDataList);
        tmpJarak[j]=tmpJarak[j-1];
        tmpJarak[j-1]=abTemp;
    }
}

}

for (int i = 0; i < tmpJarak.length; i++) {
    fixJarak[i] = tmpJarak[i];
}

for (int i = 0; i < fixJarak.length; i++) {
    System.out.println("Fix    Jarak
:"+fixJarak[i] +" Meter ");
}

    TmpdataList                =                new
ArrayList<HashMap<String,String>>(dataListTempat);

    dataListTempat=                new
ArrayList<HashMap<String, String>>());

    for (int i = 0; i < fixJarak.length; i++) {
        if (fixJarak[i] <1000) {

dataListTempat.add(TmpdataList.get(i)) ;

        }
    }

    /*for (int i = 0; i < dataListTempat.size(); i++) {
        String tmp =dataList.get(i).get("K");

```

```

String.format("##,###", tmpJarak[i]));
        tmp = tmp.replace("pcs", "pcs "+

        // dataList.get(i). = tmp;

    }*/

    }

}

if (dataListTempat.size()==0) {
    Toast.makeText(getApplicationContext(), "Data Tidak
ditemukan..", Toast.LENGTH_LONG);
}

} catch ( Exception e) {

    System.out.println(e.toString());
    return false;
}

return true;
}

// memilih salah satu list
protected void gotoItem(int position) {
    // TODO Auto-generated method stub

```

```

//tampil detail
/*Intent intent = new Intent(this, DetailDataActivity.class);
intent.putExtra("ID", dataListTempat.get(position).get("A"));
intent.putExtra("NAME",NAME);
intent.putExtra("TITLE",dataListTempat.get(position).get("C"));
intent.putExtra("DETAIL",dataListTempat.get(position).get("G"));
intent.putExtra("ALAMAT",dataListTempat.get(position).get("D"));
intent.putExtra("LAT",dataListTempat.get(position).get("E"));
intent.putExtra("LNG",dataListTempat.get(position).get("F"));

```

```

startActivity(intent);*/

```

```

final Dialog dialog = new Dialog(this);
dialog setContentView(R.layout.help);
dialog.setTitle( dataListTempat.get(position).get("C"));
dialog.setCancelable(true);

```

```

final String mimeType = "text/html";
final String encoding = "utf-8";
final String html = "<p><img src=\"file:///android_asset/img/\"+NAME.toLowerCase()+\".png\" width=\"100%\" height=\"80%\" /></p>"+
+ "<b>Alamat : </b>" +
dataListTempat.get(position).get("D")+"<br>";

```

```

WebView ww = (WebView) dialog.findViewById(R.id.webView1);
ww.loadDataWithBaseURL("", html, mimeType, encoding, "");

```

```

//set up button
Button button = (Button) dialog.findViewById(R.id.btnOk);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        dialog.cancel();
    }
});

final String lat = dataListTempat.get(position).get("E").trim();
final String lng = dataListTempat.get(position).get("F").trim();
//set up button
Button button2 = (Button) dialog.findViewById(R.id.btnRute);
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(ListDataActivity.this, MapsActivity.class);

        intent.putExtra("LAT", lat);
        intent.putExtra("LNG", lng);
        startActivity(intent);
        dialog.cancel();
    }
});
dialog.show();

}
}

```

Main activity

```
package me.atm.finder;
```

```
import java.util.ArrayList;
```

```
import me.atm.finder.R.id;
```

```
import android.app.Activity;
```

```
import android.app.AlertDialog;
```

```
import android.app.Dialog;
```

```
import android.content.DialogInterface;
```

```
import android.content.Intent;
```

```
import android.graphics.Bitmap;
```

```
import android.os.Bundle;
```

```
import android.os.StrictMode;
```

```
import android.view.View;
```

```
import android.webkit.ConsoleMessage;
```

```
import android.webkit.WebChromeClient;
```

```
import android.webkit.WebView;
```

```
import android.webkit.WebViewClient;
```

```
import android.widget.Button;
```

```
import android.widget.ProgressBar;
```

```
import android.widget.TextView;
```

```
public class MainActivity extends Activity {
```

```

/*      A1      BCA
        A2      BNI
        A3      BRI
        A4      Mandiri

*/

//init view

Button btn1,btn2,btn3,btn4,btn5,btn6;

//init kode

String[] ID_Tempat = new String[]{

        "A1",

        "A2",

        "A3",

        "A4"

};

//init jenis

String[] Jenis_Tempat = new String[]{

        "BCA",

        "BNI",

        "BRI",

        "Mandiri"

};

@Override

protected void onCreate(Bundle savedInstanceState) {

    // TODO Auto-generated method stub

    super.onCreate(savedInstanceState);

    setContentView(R.layout.menu);

    //init view

```

```
btn1=(Button) findViewById(id.btn1);  
btn1.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        show(0);  
    }  
});
```

```
btn2=(Button) findViewById(id.btn2);  
btn2.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        show(1);  
    }  
});
```

```
btn3=(Button) findViewById(id.btn4);  
btn3.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        show(3);  
    }  
});
```

```

        btn4=(Button) findViewById(id.btn5);
        btn4.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                // TODO Auto-generated method stub
                show(2);
            }

        });
    }

```

```

protected void show(int i) {

    // TODO Auto-generated method stub

    //tampil list tempat

    Intent intent = new Intent(this, ListDataActivity.class);
    intent.putExtra("ID", ID_Tempat[i]);
    intent.putExtra("NAME",Jenis_Tempat [i]);
    startActivity(intent);

}

```

```

@Override
public void onBackPressed() {

    // TODO Auto-generated method stub
    close();

}

```

```

public void close(){
    //konfirmasi close
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setMessage("Apakah yakin untuk menutup aplikasi?")
        .setCancelable(false)
        .setPositiveButton("Ya",new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,
                int id) {
                    finish();
                }
        })
        .setNegativeButton("Tidak",new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog,
        int id) {
            dialog.cancel();

        }
    }
    ).show();
}
}

```

Code Progam MapActivity

```
package me.atm.finder;
```

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStreamReader;
```

```
import java.net.HttpURLConnection;
```

```
import java.net.URL;
```

```
import java.util.List;
```

```
import java.util.Locale;
```

```
import org.apache.http.HttpRequest;
```

```
import org.apache.http.HttpRequestFactory;
```

```
import org.json.JSONArray;
```

```
import org.json.JSONException;
```

```
import org.json.JSONObject;
```

```
import android.annotation.SuppressLint;
```

```
import android.annotation.TargetApi;
```

```
import android.app.Activity;
```

```
import android.app.AlertDialog;
```

```
import android.app.ProgressDialog;
```

```
import android.base.org.sun.java.Utills;
```

```
import android.content.Context;
```

```
import android.content.DialogInterface;
```

```
import android.content.Intent;
```

```
import android.location.Location;
```

```
import android.location.LocationListener;
```

```
import android.location.LocationManager;
```

```

import android.location.GpsStatus.Listener;

import android.net.Uri;

import android.os.AsyncTask;

import android.os.Build;

import android.os.Bundle;

import android.os.StrictMode;

import android.speech.tts.TextToSpeech;

import android.speech.tts.TextToSpeech.OnInitListener;

import android.util.Log;

import android.utils.services.ApiServices;

import android.view.View;

import android.webkit.WebChromeClient;

import android.webkit.WebView;

import android.webkit.GeolocationPermissions.Callback;

import android.widget.Button;

import android.widget.ListView;

import android.widget.Toast;


@SuppressLint("NewApi")
@TargetApi(Build.VERSION_CODES.GINGERBREAD)
public class MapsActivity extends Activity implements LocationListener {


    private LocationManager locationManager;


    private String LAT;

    private String LNG;

    WebView webView;


    @SuppressLint("SetJavaScriptEnabled")
    @Override

```

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Utils.setFullScreen(this);
    setContentView(R.layout.infoview);

    Intent i = getIntent();
    LAT = i.getStringExtra("LAT");
    LNG = i.getStringExtra("LNG");

    System.out.println("MapsActivity.onCreate() : LAT : "+ LAT);
    System.out.println("MapsActivity.onCreate() : LNG : "+ LNG);

    webView = (WebView) findViewById(R.id.mapview);
    webView.getSettings().setJavaScriptEnabled(true);

    webView.setWebChromeClient(new WebChromeClient(){

        @Override
        public void onGeolocationPermissionsShowPrompt(final String
origin, final Callback callback) {
            final boolean remember = false;

            AlertDialog.Builder builder = new
AlertDialog.Builder(MapsActivity.this);

            builder.setTitle("Acces Permission..");

            builder.setMessage("Allow your application to access your
location? ")

            .setCancelable(true).setPositiveButton("Yes", new
DialogInterface.OnClickListener() {

                public void onClick(DialogInterface dialog, int id) {

                    callback.invoke(origin, true, remember);

```

```

        }
    }).setNegativeButton("No", new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int id) {

        callback.invoke(origin, false, remember);
    }
});
AlertDialog alert = builder.create();
alert.show();
}
});

```

```

webView.getSettings().setGeolocationDatabasePath("/data/data/me.atm.finder");

locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

new GettingLocation().execute("");

}

```

```

@Override
public void onBackPressed() {
    // TODO Auto-generated method stub

    finish();
    super.onBackPressed();
}

```

```
boolean isFirst = true;
```

```
class GettingLocation extends AsyncTask<String, Void, Void> {
```

```
private ProgressDialog Dialog = new ProgressDialog(MapsActivity.this);
```

```
protected void onPreExecute() {
```

```
    //progress
```

```
    Dialog.setMessage(" Loading GPS Location .");
```

```
    Dialog.show();
```

```
    Dialog.setCancelable(true);
```

```
    locString = "";
```

```
    locationManager.requestLocationUpdates( LocationManager.GPS_PROVIDER,
```

```
        100, // 10 sec
```

```
        10, MapsActivity.this);
```

```
}
```

```
protected Void doInBackground(String... urls) {
```

```
    while (locString.equals("")){
```

```
    }
```

```
    return null;
```

```
}
```

```

protected void onPostExecute(Void unused) {

    Dialog.dismiss();
}

}

class GettingNavigasi extends AsyncTask<String, Void, Void> {

private ProgressDialog Dialog = new ProgressDialog(MapsActivity.this);

protected void onPreExecute() {
    //progress
    Dialog.setMessage(" Loading data .");
    Dialog.show();
    Dialog.setCancelable(true) ;
}

protected Void doInBackground(String... urls) {

    return null;
}

protected void onPostExecute(Void unused) {

```

```

        Dialog.dismiss();
    }
}

String locString = "";

/***** Called after each 3 sec *****/

double lastLNG=0,lastLAT=0;

private String Kata;

public void onLocationChanged(Location location) {

    String str = "Lokasi Anda : \n-Latitude: "+location.getLatitude()+"\n-Longitude: "+location.getLongitude();

    locString =location.getLatitude()+","+location.getLongitude();

    if (isFirst){

        String lat2 = String.valueOf(location.getLatitude()).replace(".", ".");

        String lng2 = String.valueOf(location.getLongitude()).replace(".", ".");

        /*LNG = lng2 ;

        LAT = lat2 ;*/

        isFirst = false;

        webView.loadUrl("file:///android_asset/wgx/index.hmx?lat2="+LAT.replace(".", ".")+ "&&lng2="+LNG.replace(".", ".")+ "&&lat="+lat2+"&&lng="+lng2);

        Toast.makeText(getApplicationContext(), str, Toast.LENGTH_LONG).show();

    }

}

```

```

public void onProviderDisabled(String provider) {

    /******* Called when User off Gps *****/

    Toast.makeText(getBaseContext(), "Gps turned off ",
    Toast.LENGTH_LONG).show();

}

public void onProviderEnabled(String provider) {

    /******* Called when User on Gps *****/

    Toast.makeText(getBaseContext(), "Gps turned on ",
    Toast.LENGTH_LONG).show();

}

public void onStatusChanged(String provider, int status, Bundle extras) {
    // TODO Auto-generated method stub
}
}

```

Splash Screen Activity

```
package me.atm.finder;
```

```
import java.io.File;
```

```
import android.app.Activity;
```

```
import android.content.Intent;
```

```
import android.os.Bundle;
```

```
import android.widget.ProgressBar;
```

```
public class SplashScreenActivity extends Activity {
```

```
    ProgressBar pbMain;
```

```
    //init alamat web services
```

```
    //localhost
```

```
        //public static final String Urls = "10.0.2.2/atmadmin/inc/";
```

```
        //public static final String MainUrls = "10.0.2.2/atmadmin/";
```

```
    //wifi
```

```
    /*    public static final String Urls = "192.168.100.1/atmadmin/inc/";
```

```
        public static final String MainUrls = "192.168.100.1/atmadmin/";*/
```

```
    //online
```

```
        public static final String Urls = "apihosting.000space.com/atmadmin/inc/";
```

```
        public static final String MainUrls = "apihosting.000space.com/atmadmin/";
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.splash);
```

```
        setTitle("Selamat Datang..");
```

```
        //ini view
```

```
        final int welcomeScreenDisplay = 100;
```

```
        pbMain = (ProgressBar) findViewById(R.id.progressBar1 );
```

```
pbMain.setMax(welcomeScreenDisplay);
```

```
Thread threads = new Thread() {
```

```
    int wait = 0;
```

```
    @Override
```

```
    public void run() {
```

```
        try {
```

```
            super.run();
```

```
            //delay
```

```
            while (wait < welcomeScreenDisplay) {
```

```
                sleep(30);
```

```
                wait += 1;
```

```
                pbMain.setProgress(wait);
```

```
            }
```

```
        } catch (Exception e) {
```

```
            System.out.println("Error:" + e);
```

```
        } finally {
```

```
            newIntens();
```

```
        }
```

```
    }
```

```
};
```

```
threads.start();
```

```
}
```

```
protected void newIntens() {
```

```
// TODO Auto-generated method stub

//tampil menu utama
startActivity(new Intent( this,MainActivity.class ));
finish();
}
}
```



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : **FIRDAUS GEDE RIZQI**
Nim : **08.12.525**
Jurusan : **Teknik Elektro**
Konsentrasi : **Teknik Komputer S-1**
MasaBimbingan : **Semester Genap 2014-2015**
Judul : **APLIKASI PENCARIAN ATM DI MALANG
BERBASIS ANDROID**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)

Pada Hari : Selasa
Tanggal : 18 Agustus 2015
Dengan Nilai : 78,5 (B+)

Panitia Ujian Skripsi:

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358

Sekretaris Majelis Penguji

Dr. Eng. I Komang Somawirata, ST, MT
NIP. Y. 1030100361

Anggota Penguji :

Dosen Penguji I

Dr. Eng. I Komang Somawirata, ST, MT
NIP. Y. 1030100361

Dosen Penguji II

Yuli Wahyuni, ST, MT
NIP. P.1031200456



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Komputer, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : **FIRDAUS GEDE RIZQI**
Nim : **08.12.525**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**
Masa Bimbingan : **Semester Genap 2014-2015**
Judul Skripsi : **APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID**

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	18 Agustus 2015	1. Potongan Code Program di bahas di BAB III	
2	Penguji II	18 Agustus 2015	1. Abstrak 2. Penulisan	

Disetujui:

Penguji I



Dr. Eng. I Komang Somawirata, ST, MT
NIP. Y. 1030100361

Penguji II



Yuli Wahyuni, ST, MT
NIP. P.1031200456

Mengetahui:

Dosen Pembimbing I



M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Dosen Pembimbing II



Irmalia Suryani Faradisa, ST, MT
NIP. P. 1030000365



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Raya Karangla, Km. 2 MALANG

Formulir Perbaikan Ujian Skripsi

Dalam Pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T.Energi Listrik, /
T. Elektronika, /T. Komputer, / T.Telekomunikasi, Maka Perlu Adanya Perbaikan Skripsi Untuk Mahasiswa:

Nama : FIRDAUS GEDD R.

NIM : 08.12.525.

Perbaikan Meliputi :

- pefungsi code program di kelas pd Bus

Malang, 18-8-2018

(*[Signature]*)



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Raya Karangas, Km. 2 MALANG

Formulir Perbaikan Ujian Skripsi

Dalam Pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi T.Energi Listrik,/
T. Elektronika, /T. Komputer, / T.Telekomunikasi, Maka Perlu Adanya Perbaikan Skripsi Untuk Mahasiswa:

Nama : Firdaus Gede Rizki

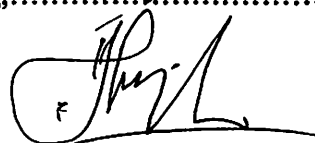
NIM : 08.12.525

Perbaikan Meliputi :

① Abstrak

② penulisan

Malang, 18 Agustus2018


(Yuli Wahyuni, ST.MT.)



FORMULIR BIMBINGAN SKRIPSI

Nim : Firdaus Gede Rizqi
Nama : 0812525
Masa Bimbingan : Semester Genap 2013 – 2014
Judul : APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID

Tanggal	Uraian	Paraf Pembimbing
12 Juli 2014	acc Bab I revisi Bab II.	
11 Nov 2014	revisi Bab II revisi Bab III.	
28 Nov 2014	acc Bab II dan III	
11 Des 2014	acc laporan progress.	
24 Jan 2015	acc Bab IV.	
1 Juli 2015	kesimpulan ditambahkan.	

Malang, 2014

Dosen Pembimbing

M. Ibrahim Ashari, ST, MT
NIP. P1030100358



FORMULIR BIMBINGAN SKRIPSI

Nim : Firdaus Gede Rizqi
Nama : 0812525
Masa Bimbingan : Semester Genap 2013 – 2014
Judul : APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID

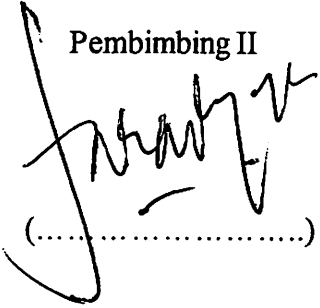
Tanggal	Uraian	Paraf Pembimbing
4 - 11 - 2014	perancangan sistem.	
25 - 10 - 2014	perancangan algoritma. Dsikstra	
11 - 12 - 14	blm ada kemajuan	
26 - 6 - 2015	Rumus LBS	
7 - 7 - 2015	Keterangan Rumus dan flowchart	
1 - 8 - 2015	Revisi tata tulis, kesimpulannya & saran	

Malang, 2014
Dosen Pembimbing

Irmalia Suryani Faradisa, ST, MT
NIP.P1030000365

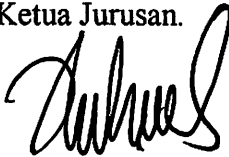
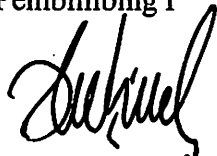


BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Komputer

1.	Nim	: 0812525		
2.	Nama	: FIRDAUS GEDE RISQI		
3.	Konsentrasi Jurusan	: Teknik Komputer		
4.	Jadwal Pelaksanaan:	Waktu	Tempat	
	15 Mei 2013	09:00	III.1.5	
5.	Judul proposal yang diseminarkan Mahasiswa	APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID		
6.	Perubahan judul yang diusulkan oleh Kelompok Dosen Keahlian			
7.	Catatan : → Spesifikasi pecahan Uang pada ATM → metode pencarian yg digunakan diperjelas			
8.	Catatan :			
	Persetujuan judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
	(.....)	(.....)	(.....)	
	Disetujui, Calon Dosen Pembimbing ybs			
	Mengetahui, Ketua Program Studi Teknik Elektro S-1	Pembimbing I	Pembimbing II	
	 M. Ibrahim Ashari, ST, MT NIP. P 1030100358	(.....)	 (.....)	



KONSENTRASI	T. KOMPUTER
--------------------	-------------

1.	Nama Mahasiswa	FIRDAUS GEDE RISQI	NIM	812525
2.	Keterangan	Tanggal	Waktu	Tempat / Ruang
	Pelaksanaan	12 Desember 2014		
3.	Judul Skripsi	APLIKASI Pencarian ATM di Malang Berbasis Android		
4.	Perubahan Judul			
5.	Catatan : lanjutan - Design penemuan koordinat - Penemuan di google map. - Penemuan pencarian ATM. - Pengujian - Kesimpulan dan saran. - Daftar pustaka			
6.	Mengetahui, Ketua Jurusan.  <u>M. Ibrahim Ashari, ST, MT</u>	Disetujui, Dosen Pembimbing		
		Pembimbing I  M. Ibrahim Ashari, ST, MT	Pembimbing II Irmalia Suryani Faradisa, ST, MT	



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-215/EL-FTI/2015
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Tanggal, 28 Mei 2015

Kepada : Yth. Bapak/Ibu **M. Ibrahim Ashari, ST, MT**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **FIRDAUS GEDE RISQI**
Nim : **0812525**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

“Semester Genap Tahun Akademik Genap 2014 - 2015”

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui
Dosen Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358



PROGRAM STUDI TEKNIK ELEKTRO S-1

**FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **FIRDAUS GEDE RISQI**

Nim : **0812525**

Semester : **X (Sepuluh)**

Jurusan : **Teknik Elektro S-1**

Konsentrasi : **Teknik Komputer**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

*) Coret yang tidak perlu



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-215/EL-FTI/2015
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Tanggal, 28 Mei 2015

Kepada : Yth. Bapak/Ibu **Irmalia Suryani Faradisa, ST, MT**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **FIRDAUS GEDE RISQI**
Nim : **0812525**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

“ Semester Genap Tahun Akademik Genap 2014 - 2015 “

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui
Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **FIRDAUS GEDE RISQI**

Nim : **0812525**

Semester : **X (Sepuluh)**

Jurusan : **Teknik Elektro S-1**

Konsentrasi : **Teknik Komputer**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" APLIKASI PENCARIAN ATM DI MALANG BERBASIS ANDROID"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

Irmalia Suryani Faradisa, ST, MT

NIP.P. 1030000365

Catatan :

Setelah disetujui agar formulir ini Diserahkan mahasiswa/i yang bersangkutan kepada jurusan untuk diproses lebih lanjut

*) Coret yang tidak perlu