

UNIVERSITY OF MALAYA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING
KUALA LUMPUR

1999

Copyright © 1999 by the author(s).
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage and retrieval system, without permission in writing from the author(s).

Printed in Malaysia
by the author(s)
1999

PROGRAM STUDI TEKNIK ELEKTRO 2-1
KEPERAWATAN TEKNIK KOMPUTER DAN MANAJEMEN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALAYSIA
2013

LEMBAR PERSETUJUAN

APLIKASI PENCARIAN LOKASI SMARTPHONE ANDROID BERDASARKAN TITIK KOORDINAT GPS MELALUI AUTO-REPLY SMS

SKRIPSI

*Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai Gelar
Sarjana Teknik Informatika Strata Satu (S-1)*

Disusun Oleh :

Agung Pranoto

07.12.536

Diperiksa dan Disetujui oleh

Dosen Pembimbing I



M. Ibrahim Ashari, ST, MT
NIP.Y. 1028400082

Dosen Pembimbing II



Yuli Wahyuni, ST, MT
NIP.P. 1031200456

Mengetahui

Ketua Program Studi Teknik Informatika S-1



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

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Agung Pranoto
NIM : 07.12.536
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika S-1

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, 19 Agustus 2013

Yang membuat Pernyataan,



Agung Pranoto
07.12.536

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN	ii
SURAT PERNYATAAN	iii
ABSTRAK	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan	2
1.4. Batasan Masalah	2
1.5. Metodologi Penelitian	3
1.6. Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	5
2.1. Pengetahuan Dasar Tentang GPS	5
2.1.1. A-GPS (Assisted Global Positioning System)	6
2.1.2. Perbandingan GPS dan A-GPS	8
2.2. Android	9
2.2.1. Arsitektur Android	9
2.2.2. Komponen-komponen Android	12
2.2.3. Versi Android	17

2.2.4	The Dalvik Virtual Machine.....	20
2.2.5	Android SDK.....	20
2.2.6	ADT (Android Development Tools).....	21
2.2.7	Android Virtual Device (AVD)	21
2.2.8	Komponen Android	21
2.3.	Java	22
2.4.	Corel Draw	23
2.5.	MIT App Inventor	23
2.5.1	Memulai App Inventor.....	24
2.5.2	Instalasi MIT App Inventor.....	26
2.5.3	Lingkup Kerja App Inventor.....	29
BAB III ANALISA SISTEM DAN PERANCANGAN		32
3.1.	Analisa Sistem.....	32
3.1.1	Deskripsi Sistem	32
3.1.2	Analisa Kebutuhan	32
3.2.	Alur Sistem.....	33
3.2.1.	Desain Tampilan Flow Chart Utama	33
3.3	Perancangan Aplikasi	36
3.4	Desain Alur Menu.....	37
3.5	Desain Antar Muka Aplikasi.....	37
3.5.1	Desain Tampilan	37
BAB IV IMPLEMENTASI DAN PENGUJIAN		42
4.1.	Implementasi	42
4.2.	Pengujian	42

4.2.1. Tampilan Awal	42
4.2.2. Tampilan Menu	43
4.3 Perbandingan Antar OS	48
4.4 Tabel Pengujian	49
4.3. Pengujian Aplikasi	42
BAB V PENUTUP	51
5.1. Kesimpulan.....	51
5.2. Saran.....	51
DAFTAR PUSTAKA.....	52

DAFTAR TABEL

Tabel 3.1. Keterangan Diagram Alur	36
Tabel 4.1 Hasil Pengujian Beberapa Tempat.....	49
Tabel 4.2. Hasil Pengujian HP Peminta Lokasi	50

DAFTAR GAMBAR

BAB II TINJAUAN PUSTAKA

Gambar 2.1. Arsitektur Android.....	10
Gambar 2.2. Siklus Activity.....	15
Gambar 2.3. Siklus Service.....	16
Gambar 2.4. Java	22
Gambar 2.5. Corel Draw.....	23
Gambar 2.6. Jendela Kerja App Inventor	24
Gambar 2.7. App Inventor Setup	25
Gambar 2.8. Gambar Setingan Smartphone.....	26
Gambar 2.9. Pemilihan USB Debugging.....	27
Gambar 2.10. Membuat Project Baru	28
Gambar 2.11. Project Baru.....	29
Gambar 2.12. Komponen Desainer	29
Gambar 2.13. Block Editor	30
Gambar 2.14. Emulator.....	31

BAB IV ANALISA DAN DESAIN SISTEM

Gambar 3.1. Flowchart Sistem.....	34
Gambar 3.2. Diagram Alur Aplikasi	35
Gambar 3.3. Desain Tampilan Awal Aplikasi	37
Gambar 3.4. Block Edit Tampilan Awal	38
Gambar 3.5. Desain Tampilan Layar ke-2.....	38

Gambar 3.6. Block Edit Layar ke-2	39
Gambar 3.7. Desain Layar Titik Koordinat	39
Gambar 3.8. Block Editor Inti Aplikasi.....	40
Gambar 4.1. Tampilan Awal Aplikasi	42
Gambar 4.2. Tampilan Layar Ke-2.....	43
Gambar 4.3. Tampilan Menu Utama	44
Gambar 4.4. Tampilan SMS Meminta Lokasi	45
Gambar 4.5. Tampilan SMS Balasan	46
Gambar 4.6. Tampilan Pada Smartphone Dan SMS Yang Dikirim.....	47
Gambar 4.7. Tampilan Android Beda Versi	48

APLIKASI PENCARIAN LOKASI SMARTPHONE ANDROID BERDASARKAN TITIK KOORDINAT GPS MELALUI AUTO- REPLY SMS

AGUNG PRANOTO

Konsentrasi Teknik Komputer dan Informatika, Jurusan Teknik Elektro S-1
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km 2 Malang
e-mail : aggiua180818@gmail.com

Dosen Pembimbing : 1. M. Ibrahim Ashari, ST, MT
2. Yuli Wahyuni, ST, MT

Kata kunci : GPS, android, deteksilokasi, auto-reply sms

Abstrak

GPS atau Global Positioning System telah lama digunakan oleh pihak militer sebagai alat bantu navigasi. Dengan GPS, posisi koordinat lintang dan bujur dapat diketahui. GPS saat ini sering kita temukan diperangkat elektronik sekitar kita, seperti pada smartphone

Smartphone Android dengan salah satu fitur yang tertanam di dalamnya adalah fitur GPS yang bersistem A-GPS. Selain sebagai penunjuk arah juga dapat digunakan untuk menentukan posisi dimana user berada. Penulisan skripsi ini ingin memaparkan pembuatan sebuah rancang bangun aplikasi pencarian lokasi smartphone android berdasarkan titik koordinat GPS melalui auto-reply sms yang mudah digunakan. Pembangunan aplikasi ini di dukung dengan software MIT App Inventor.

Dengan aplikasi pencari titik koordinat smartphone ini, dapat membantu dalam mengetahui posisi dan letak smartphone lewat titik koordinat. Dibuat dengan lebih sederhana untuk kemudahan penggunaanya, agar dapat digunakan dengan baik oleh semua orang.

Abstract

GPS or Global Positioning System has long been used by the military as a navigational aid. With GPS, the coordinates of latitude and longitude can be determined. GPS is now often we found in electronic device saroundus, such ason a smartphone.

Android smartphone with one of the features embedded in it is a GPS feature that applying A-GPS. In addition to the directions can also be used to determine the position where the useris located. This thesis would like to describe the creation of a design application based location search smartphone android GPS coordinate points via the auto-reply sms is easy to use. Application development is supported by MIT App Inventor software.

With coordinate points smartphone seekers app, it can be helpful in knowing the position and location of the smart phone through the coordinate points. Created with much simpler for ease of use, to be used well by everyone.

KATA PENGANTAR

Dengan mengucapkan syukur kehadiran Allah SWT yang dengan segala Kasih dan Anugerah – Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul: **“APLIKASI Pencarian Lokasi Smartphone Android Berdasarkan Titik Koordinat GPS Melalui Auto-Reply SMS”**

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. Ir. Soeparno Djiwo, MT selaku Rektor ITN Malang.
2. Ir. Anang Soebardi, MT selaku Dekan Fakultas Teknologi Industri.
3. M. Ibrahim Ashari, ST, MT selaku Ketua Jurusan Teknik Elektro S – 1 ITN Malang
4. M. Ibrahim Ashari, ST, MT selaku Dosen Pembimbing I.
5. Yuli Wahyuni, ST, MT selaku Dosen Pembimbing II.
6. Kedua orang tua yang selalu memberikan do'a restu, dorongan dan semangat.
7. 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. Amiin.

Malang, 19 Agustus 2013

BAB I PENDAHULUAN

Pada bab pendahuluan ini akan dibahas tentang latar belakang, permasalahan, ruang lingkup, tujuan, metode penelitian, dan rencana susunan penulisan dari tugas akhir yang diambil.

1.1.Latar Belakang

Perkembangan teknologi saat ini sangatlah cepat. Saat ini handphone tidak hanya digunakan untuk berkomunikasi tetapi juga untuk ber-internet-an, bermain game, memproses dokumen, serta berfungsi sebagai peta atau pembaca lokasi. Dengan berkembangnya teknologi mobile saat ini, perangkat mobile menawarkan kemampuan komputansi canggih yang disebut sebagai smartphone. Salah satu smartphone yang sedang trend saat ini adalah smartphone berbasis operasi Android.

Titik koordinat GPS (*Global Positioning system*) adalah titik yang berpedoman pada garis latitude dan longitude suatu daerah. Dimana kaitannya dengan garis latitude dan longitude adalah kedua garis lintang dan bujur (*latitude* = garis lintang; *longitude* = garis bujur). Titik koordinat inilah sebagai tolak ukur GPS mengenali dan membaca letak geografis dan alamat. Dengan adanya GPS ini kita mampu mengetahui letak lokasi suatu tempat. Belakangan, sebagian besar smartphon berbasis android telah dibekali fitur GPS/AGPS. Ini memudahkan kita untuk mengetahui lokasi sekitar smartphone android kita. Jika kita dapat mengetahui lokasi smartphone android akan sangat berguna bagi kita. Seperti smartphone android kita hilang atau tertinggal di suatu tempat, mengetahui lokasi orang lain atau orang terdekat kita, dan beberapa kegunaan lain ketika kita mengetahui lokasi letak geografis smartphone android

Penulisan skripsi ini ingin memaparkan pembuatan sebuah rancang bangun aplikasi pencarian lokasi smartphone android berdasarkan titik koordinat gps melalui auto-reply sms. Yang dapat membantu dalam mengetahui lokasi dan letak

smartphone lewat titik koordinat. Membuat lebih mudah dan lebih sederhana untuk kemudahan penggunaanya, agar dapat digunakan dengan baik oleh semua orang.

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang, maka terdapat beberapa rumusan bagaimana merancang dan membangun suatu aplikasi pendeteksi lokasi yang berpedoman pada titik koordinat GPS yang bekerja pada A-GPS pada ponsel android dan bersistem *auto-reply* sms, dengan menggunakan perangkat lunak APP Invertor.

1.3.TUJUAN

Berdasarkan rumusan masalah yang telah dikemukakan, tujuan utama yang diharapkan adalah :

1. Mengetahui letak lokasi smartphone android dengan pendeteksi sinyal GPS;
2. Memudahkan semua orang melacak atau mengetahui letak smartphon android karena *auto-replay* SMS yang bisa diterima oleh semua jenis handphone..

1.4. Batasan Masalah

Berkaitan dengan indentifikasi masalah diatas, maka diberikan beberapa batasan masalah yaitu:

1. GPS yang digunakan adalah GPS/A-GPS yang tertanam pada smartphone android.
 2. Data GPS yang diolah adalah *Latitude*, *Longitude*, dan *Altitude*.
 3. Software utama menggunakan MIT App Invertor.
 4. Software pendukung menggunakan Corel Draw X4.
 5. Handphone yang digunakan menggunakan handphone android yang digunakan sebagai pengolah data aplikasi.
 6. Menggunakan HP yang mendukung fitur SMS untuk meminta dan menerima pesan.
-

1.5. METODOLOGI PENELITIAN

Metode yang diambil dalam merancang aplikasi Pencari Lokasi Smartphone Android Berdasarkan Titik Koordinat Gps Melalui Auto-Reply Sms adalah :

1. Analisa Sistem

Metode ini dilaksanakan dengan melakukan studi kepustakaan melalui membaca buku-buku maupun artikel-artikel yang dapat mendukung penulisan skripsi ini. Teori pemrograman MIT App Inventor, cara kerja GPS. Melakukan survei pada software GPS yang sudah ada sebelumnya

2. Design Sistem

Pada Tahap ini akan dilakukan perancangan aplikasi dengan memberikan gambaran melalui diagram agar didapatkan *design* yang sempurna sesuai yang diharapkan.

- Merancang dan membuat tampilan longitude, latitude, dan altitude.
- Merancang dan membuat auto reply sms dengan isi koordinat GPS.
- Merancang desain interface.
- Mengimplementasikan rancangan dan design yang telah di buat

3. Implementasi

Pada tahap ini dilakukan pengimplementasian menggunakan bahasa pemrograman atau yang berjalan pada OS Android yang kemudian dilakukan pengujian apakah sudah bekerja dengan baik dan melihat apakah perangkat lunak tersebut berjalan dengan baik.

4. Pengujian

Setelah proses pengkodean selesai, maka akan dilakukan proses pengujian terhadap program yang dihasilkan untuk mengetahui apakah program sudah berjalan dengan benar sesuai dengan perancangan yang dilakukan.

1.6. SISTEMATIKA PENULISAN

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

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang masalah, rumusan masalah, tujuan dan manfaat dari penulisan skripsi ini, batasan masalah, dan sistematika penulisan.

BAB II : DASAR TEORI

Bab ini mengungkapkan tentang konsep dasar dan teori-teori yang mendukung pembahasan untuk tema penulisan ini yang didapat dari beberapa literatur.

BAB III : PERANCANGAN SISTEM

Pada bab ini membahas tentang proses pengembangan perangkat lunak dan hasil yang didapatkan pada tahap analisis dan perancangan.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Berisi tentang implementasi dari perancangan aplikasi yang telah dibuat serta pengujian terhadap aplikasi tersebut.

BAB V : PENUTUP

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk penulisan selanjutnya.

BAB II

LANDASAN TEORI

Pada bab ini akan di bahas landasan-landasan teori apa saja yang di pakai. Dimulai dari pengetahuan dasar tentang GPS dan A-GPS secara umum, serta perangkat lunak yang akan digunakan.

2.1. Pengetahuan Dasar Tentang GPS (Global Positioning System)

GPS (Global Positioning System) merupakan sistem navigasi satelit yang dikembangkan oleh Departemen Pertahanan Amerika Serikat (US DoD = United States Department of Defense). GPS memungkinkan kita mengetahui lokasi geografis kita (lintang, bujur, dan ketinggian di atas permukaan laut). Jadi dimanapun kita berada di muka bumi ini, kita dapat mengetahui lokasi kita dengan tepat.

GPS terdiri dari 3 segmen, yaitu:

- Segmen angkasa: terdiri dari 24 satelit yang beroperasi dalam 6 orbit pada ketinggian 20.200 km dan inklinasi 55 derajat dengan periode 12 jam (satelit akan kembali ke titik yang sama dalam 12 jam). Satelit tersebut memutari orbitnya sehingga minimal ada 6 satelit yang dapat dipantau pada titik manapun di bumi ini. Satelit tersebut mengirimkan lokasi dan waktu kepada pengguna seluruh dunia.
- Segmen Kontrol/Pengendali: terdapat pusat pengendali utama yang terdapat di Colorado Springs, dan 5 stasiun pemantau lainnya dan 3 antena yang tersebar di bumi ini. Stasiun pemantau memantau semua satelit GPS dan mengumpulkan informasinya. Stasiun pemantau kemudian mengirimkan informasi tersebut kepada pusat pengendali utama yang kemudian melakukan perhitungan dan pengecekan orbit satelit. Informasi tersebut kemudian dikoreksi dan dilakukan pemuktahiran dan dikirim ke satelit GPS.

- **Segmen Pengguna:** Pada sisi pengguna dibutuhkan penerima GPS (selanjutnya kita sebut perangkat GPS) yang biasanya terdiri dari penerima, prosesor, dan antena, sehingga memungkinkan kita dimanapun kita berada di muka bumi ini (tanah, laut, dan udara) dapat menerima sinyal dari satelit GPS dan kemudian menghitung lokasi, kecepatan dan waktu.

2.1.1. A-GPS (Assisted Global Positioning System)

Perangkat GPS menangkap 3 sinyal dari 24 satelit GPS yang ada di luar angkasa. Perangkat GPS kemudian melakukan perhitungan jarak ke setiap satelit. Dari perhitungan itu, perangkat GPS kemudian mengetahui lokasi pemegangnya. Supaya perangkat GPS bisa menangkap sinyal dengan baik, perangkat harus berada di luar ruangan, bahkan harus dibawah langit terbuka. Kekuatan sinyal bisa berkurang kalau perangkat GPS berada di bawah pohon, dibawah gedung-gedung pencakar langit atau di dalam kendaraan. Sinyal hampir bisa dipastikan menghilang kalau pembawa perangkat GPS masuk ke dalam gedung. Itulah kelemahan GPS. Makanya, muncullah perbaikan untuk masalah itu. Ialah A-GPS, perbaikannya. A-GPS adalah kependekan dari Assisted-GPS. A-GPS adalah merupakan pengembangan dari GPS.

Karena sinyal handphone bisa menembus ke dalam ruangan, masalah yang dihadapi oleh GPS bisa pun teratasi. Assisted Global Positioning System (A-GPS) akan membantu perangkat Anda menemukan satelit. Saat menggunakan A-GPS, perangkat Anda akan menerima informasi satelit dari server data melalui jaringan seluler. Apabila perangkat Anda tidak menerima data tersebut, perangkat akan mencoba mendeteksi satelit lain. Dengan bantuan data tersebut, perangkat Anda akan mendeteksi satelit yang tepat, di saat satelit berada di sisi yang sama pada planet dengan perangkat Anda. A-GPS mempercepat penandaan kalkulasi lokasi.

A-GPS didesain untuk memungkinkan perangkat dalam mendapatkan sinyal satelit lebih cepat dan dapat dipercaya daripada dengan GPS standalone. Sebagai contoh, positioning dilakukan lebih cepat dalam keadaan cold-start, yaitu, saat koneksi GPS telah dimatikan dalam waktu yang lama, atau apabila pengguna melakukan perjalanan ke negara yang berbeda. Dengan A-GPS, waktu perkiraan untuk membangun koneksi GPS menurun secara signifikan. A-GPS menurunkan

waktu yang diperlukan umumnya puluhan detik.

A-GPS Merupakan teknologi yang menggunakan server bantuan untuk memotong waktu yang dibutuhkan untuk menentukan lokasi (pinpointed) penggunaan GPS. A-GPS membutuhkan 3 komponen dalam proses penentuan lokasi, yaitu Satelit, Assistance Server (GSM), Receiver A-GPS. Server bantuan penyedia data informasi satelit yang dibutuhkan oleh sistem GPS biasanya di supply oleh jaringan operator dengan menggunakan jalur GPRS. Server ini memiliki satelit dengan sinyal yang sangat baik dan mampu melakukan perhitungan yang rumit untuk menentukan lokasi handphone. Server itu juga bisa menyajikan data dari satelit GPS sehingga handphone terhubung dengan satelit kemudian menentukan lokasi handphone. Karena sinyal handphone bisa menembus ke dalam ruangan, masalah yang dihadapi oleh GPS bisa pun teratasi. Ketika berada di daerah-daerah yang tidak terjangkau oleh satelit seperti di bawah atap, A-GPS mengambil alih tugas mencari lokasi chip GPS dalam handphone.

Karena A-GPS dan server bantuan berbagi tugas, maka proses akan lebih cepat dan lebih efisien dibanding GPS biasa walaupun semua proses itu tergantung pada cakupan jaringan. Keuntungan A-GPS adalah Pengidentifikasian lokasi yang lebih cepat, hanya membutuhkan power yang kecil untuk proses komputasi data, dan lebih hemat baterai. Sedangkan kekurangannya masih bergantung pada cakupan jaringan operator.

GPS mempunyai beberapa batasan yang juga berlaku pada A-GPS. Sebagai contoh, keberadaan dan kualitas sinyal dipengaruhi oleh bangunan, gangguan alam, kondisi cuaca, dan lokasi pengguna. A-GPS menggunakan koneksi jaringan seluler 3G dan 2G dan koneksi data paket GPRS dan EGPRS. Anda juga harus mempunyai titik akses internet yang ditentukankan dalam perangkat kompatibel Anda. A-GPS dapat digunakan di seluruh dunia, selama Anda memiliki akses ke jaringan seluler dan koneksi data. Anda juga dapat menggunakan layanan ini di luar jaringan lokal Anda (roaming). A-GPS tidak tergantung pada layanan operator yang spesifik.

2.1.2. Perbandingan GPS dan A-GPS (Kelebihan dan Kekurangan)

Berikut ini adalah perbedaan GPS dan A-GPS, yaitu:

a. Kelebihan GPS

Perangkat GPS menentukan lokasi dari minimal 3 satelit yang membentuk kawasan segitiga dengan mencari longitude, latitude, dan data lainnya yang diperlukan sehingga penentuan lokasi akan lebih akurat dibanding A-GPS yang hanya menggunakan 2 satelit untuk menemukan lokasi.

b. Kekurangan GPS :

- Kekuatan sinyal bisa berkurang kalau perangkat GPS berada di bawah pohon, dibawah gedung-gedung pencakar langit atau di dalam kendaraan. Sinyal hampir bisa dipastikan menghilang kalau pembawa perangkat GPS masuk ke dalam gedung.
- Penggunaan GPS untuk mengetahui lokasi yang mengandalkan setidaknya tiga satelit ini tidak selamanya akurat.
- Terkadang, dibutuhkan satu satelit untuk memperbaiki sinyal yang diterima. Ketidakakuratan lokasi yang ditunjukkan
- GPS ini dipengaruhi oleh lokasi satelit yang berubah dan adanya proses sinyal yang ditunda. Kecepatan sinyal GPS ini juga seringkali berubah karena dipengaruhi oleh kondisi atmosfer yang ada.
- Sinyal GPS juga mudah berinterferensi dengan gelombang elektromagnetik lainnya

c. Kelebihan A-GPS

- A-GPS berbeda dari reguler GPS dengan menambahkan elemen lain ke dlm proses pencarian lokasi, yaitu Server Bantuan (Assistance Server). Pada jaringan A-GPS, penerima, yang terbatas dalam pengolahan daya dan biasanya berada pd lokasi tidak ideal, berkomunikasi dengan bantuan server yang memiliki kemampuan tinggi dlm pengolahan dan akses ke satelite. Karena A-GPS module dan Server Bantuan berbagi tugas, proses akan lebih cepat dan lebih efisien dibanding GPS.
 - Pengidentifikasian lokasi lebih cepat dari pada GPS karena A-GPS langsung mencari satelit yang terdekat dengan lokasi
 - Membutuhkan power lebih kecil utk proses komputasi data.
-

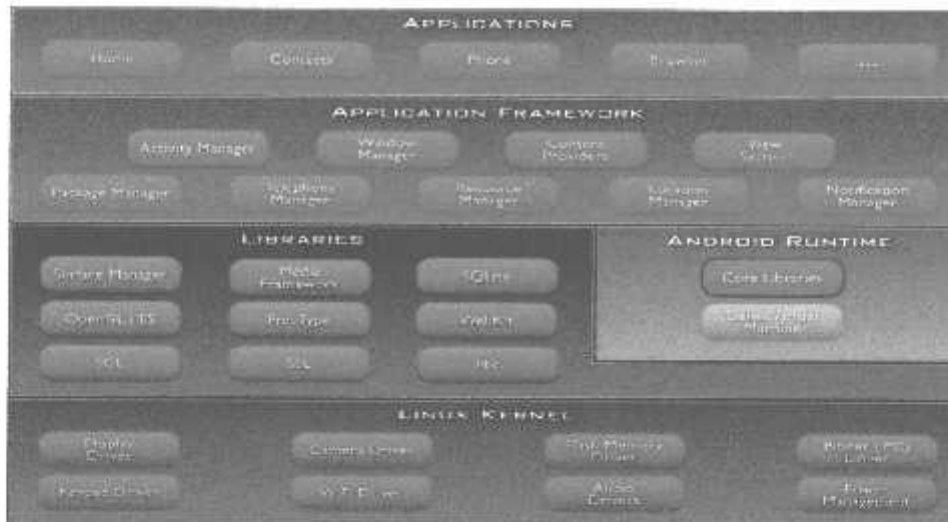
- Lebih hemat baterai.
 - Cocok utk lokasi perkotaan atau lokasi yg kurang optimal dlm menangkap sinyal satelite spt dlm gedung.
 - A-GPS memungkinkan pengguna untuk mengakses lebih banyak layanan di perangkat bergerak
- d. Kekurangan A-GPS:
- Masih tergantung pd GSM coverage operator.
 - A-GPS dapat menemukan lokasi dengan lebih cepat, karena A-GPS langsung mencari satelit yang terdekat dengan lokasi saat itu melalui operator telekomunikasi. Sehingga lokasi yang dilacak memang tidak seakurat GPS yang berpedoman pada minimal 3 satelit

2.2. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam peranti bergerak. Aplikasi Android ditulis dalam bahasa pemrograman Java. Kode Java dikompilasi bersama dengan data file resource yang dibutuhkan oleh aplikasi, sehingga menghasilkan file dengan ekstensi apk.

2.2.1 Arsitektur 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:



Gambar 2.1. Arsitektur Android

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.

2. Android Runtime

Lapisan setelah Kernel Linux adalah Android Runtime. Android Runtime ini berisi *Core Libraries* dan *Dalvik Virtual Machine*. *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.

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 memungkinkan pengguna mengganti komponen-

komponen yang dikehendaki. Di dalam semua aplikasi terdapat servis dan sistem yang meliputi :

- a. Satu set Views yang dapat digunakan untuk membangun aplikasi meliputi lists, grids, text boxes, buttons, dan embeddable web browser
- b. Content Providers yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain (misalnya Contacts), atau untuk membagi data yang dimilikinya.
- c. Resource Manager, menyediakan akses ke non-code resources misalnya localized strings, graphics, dan layout files .
- d. Notification Manager yang memungkinkan semua aplikasi untuk menampilkan custom alerts pada the status bar .
- e. Activity Manager yang manage life cycle of dari aplikasi dan menyediakan common navigation backstack .

5. Application

Di lapisan teratas bercokol aplikasi itu sendiri. Di lapisan inilah anda menemukan fungsi-fungsi dasar smartphone seperti menelepon dan mengirim pesan singkat, menjalankan web browser, mengakses daftar kontak, dan lain-lain. Bagi rata-rata pengguna, lapisan inilah yang paling sering mereka akses. Mereka mengakses fungsi-fungsi dasar tersebut melalui user interface.

2.2.2 Komponen – Komponen Android

Fitur penting android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur scroller dan aplikasi lain telah mengembangkan fitur scroller yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan *scroller* yang telah ada. Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya

android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi `main()` pada aplikasi lain. Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan. Komponen tersebut yaitu:

1. Activity

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena Activity menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap Activity dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari Views dan respon terhadap Event. Setiap aplikasi memiliki sebuah activity atau lebih.

Perpindahan antara activity dengan activity lainnya diatur melalui sistem, dengan memanfaatkan activity stack. Keadaan suatu activity ditentukan oleh lokasinya dalam tumpukan activity, LIFO (Last In First Out) dari semua aplikasi yang sedang berjalan. Bila suatu activity baru dimulai, activity yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan activity sebelumnya, cukup menekan tombol Back, atau menutup activity yang sedang digunakan, maka activity yang berada diatas akan aktif kembali. Memory Manager android menggunakan tumpukan ini untuk menentukan prioritas aplikasi berdasarkan activity, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

Ketika activity diambil dan disimpan dalam tumpukan activity terdapat 4 kemungkinan kondisi transisi yang akan terjadi, yaitu:

(a) Active

Setiap activity yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukan dari pengguna. Android akan berusaha untuk membuat activity aplikasi ini untuk tetap hidup dengan segala cara, bahkan akan menghentikan activity yang berada dibawah tumpukkannya jika diperlukan. Ketika activity sedang

aktif, maka yang lainnya akan dihentikan sementara.

(b) Paused

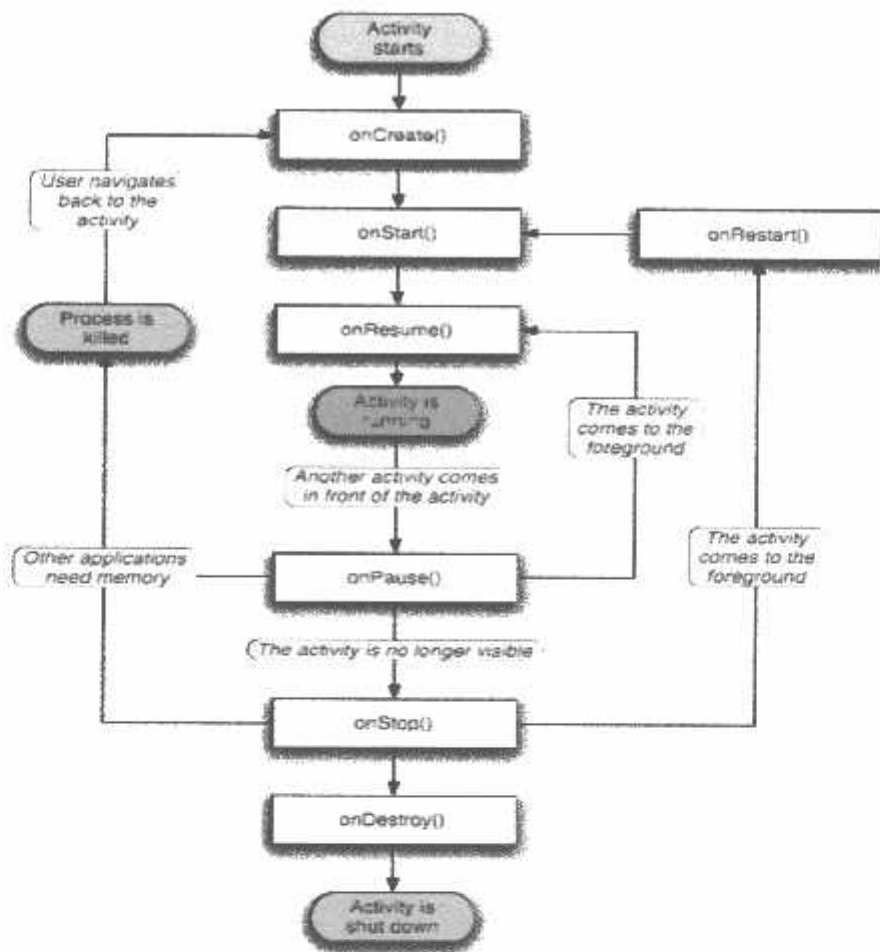
Dalam beberapa kasus activity akan terlihat tapi tidak terfokus pada kondisi inilah disebut paused. Keadaan ini terjadi jika activity transparan dan tidak fullscreen pada layar. Ketika activity dalam keadaan paused, dia terlihat active namun tidak dapat menerima masukan dari pengguna. Dalam kasus ekstrim, android akan menghentikan activity dalam keadaan paused ini, untuk menunjang sumber daya bagi activity yang sedang aktif.

(c) Stopped

Ketika sebuah activity tidak terlihat, maka itulah yang disebut stopped. Activity akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu activity dalam kondisi stopped maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika activity telah keluar atau ditutup, maka dia akan menjadi inactive.

(d) Inactive

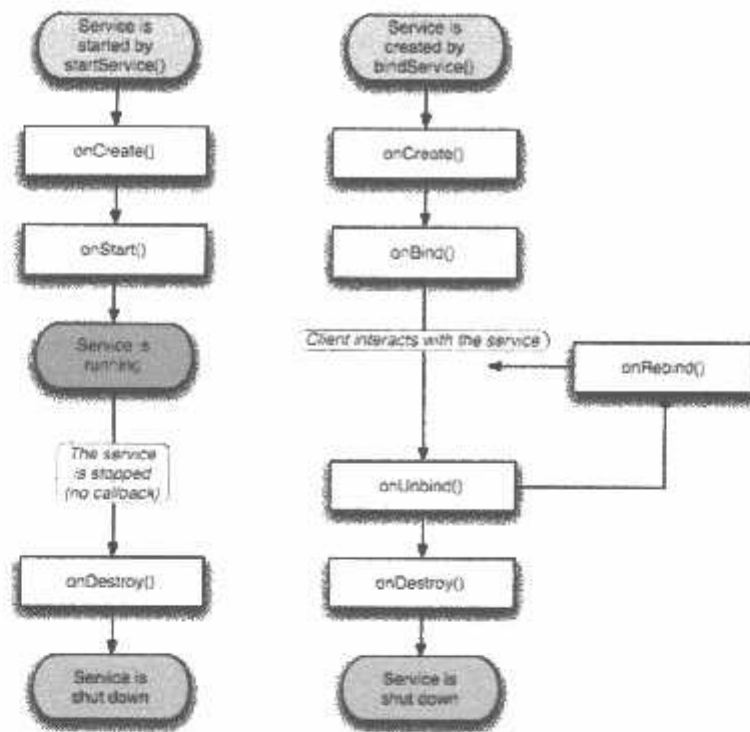
Kondisi ketika activity telah dihentikan dan sebelum dijalankan. Inactive activity telah ditiadakan dari tumpukan activity sehingga perlu restart ulang agar dapat tampil dan digunakan kembali.



Gambar 2.2. Siklus Activity

2. Service

Suatu service tidak memiliki tampilan antarmuka, melainkan berjalan di background untuk waktu yang tidak terbatas. Komponen service diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. Service digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika Activity tidak aktif atau tidak tampak.



Gambar 2.3 Siklus Service

3. Intent

Android menggunakan class *Intent* untuk melakukan aksi, berubah dari satu layar ke layar yang lain. Dua hal utama yang perlu diperhatikan pada struktur class *Intent* adalah aksi yang dilakukan dan data yang dibawa.

Sebuah class *intent* tidak dapat melakukan sendiri, namun membutuhkan sebuah *intent receiver* dan *intent filter*. *Intent receiver* akan menerima semua *activity* yang diberikan kemudian oleh *intent filter* akan dipilih aksi mana yang akan dijalankan.

4. Broadcast Receiver

Broadcast Receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar Broadcast berasal dari sistem misalnya, Baterai sudah hampir habis, informasi zona waktu telah berubah, atau pengguna telah merubah bahasa default pada perangkat. Sama halnya dengan service, Broadcast Receivers tidak menampilkan antarmuka

pengguna. Namun, *Broadcast Receivers* dapat menggunakan *Notification Manager* untuk memberitahukan sesuatu kepada pengguna.

5. *Content Provider*

Content Providers digunakan untuk mengelola dan berbagi database. Data dapat disimpan dalam file sistem, dalam database SQLite, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya Content Provider memungkinkan antar aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya.

2.2.3 Versi Android

1. Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 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 pembaruan 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.

3. Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet Virtual Private Network (VPN). Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera,

camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4. Android versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan User Interface (UI) dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (killer apps – aplikasi unggulan). Kompetisi ini berhadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada 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.

5. Android versi 2.2 (Froyo: Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan 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 Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

6. Android versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0 (Honeycomb)

Android 3.0 adalah versi yg secara khusus dioptimalkan untuk gadget dengan layar lebar, khususnya, dan yaitu tablet. Dalam versi ini diperkenalkan desain UI yang baru, virtual dan holografis tapi juga elegan dengan model interaksi memfokuskan pada konten. Dalam versi ini Google membuat multitasking, notifications, Home screen customization, widgets, dan yang lainnya juga menjadi lebih interaktif, vibrant, dan memberikan pengalaman 3D tapi tetap familiar dan juga lebih baik dari sebelumnya

Perubahan-perubahan yang dilakukan di versi ini antara lain pada System Bar (di bagian global status dan notifikasi), Action Bar (di bagian application control), customizable Home screens, recent Apps (untuk multitasking yang lebih mudah), keyboard (di desain ulang agar mengetik bisa lebih cepat dan akurat), text selection (copy dan paste yang lebih baik lagi), opsi connectivity yang baru, browser, camera dan gallery, contact, dan email. Semuanya itu ditujukan agar lebih maksimal di dalam tablet. dan memang versi ini sampai saat versi 4.0 resmi diumumkan hanya beredar dan dipakai di tablet.

8. Android versi 4.0 (Ice Cream Sandwich)

Ice Cream Sandwich didesain untuk baik itu telepon ataupun tablet. Android ICS menawarkan banyak peningkatan dari apa yg sudah ada di Gingerbread dan Honeycomb dengan pada saat yang sama memberikan inovasi-inovasi baru. Beberapa peningkatan itu antara lain kemampuan copy paste yang lebih baik, data logging dan warnings, dan kemampuan utk mengambilscreenshot dengan menekan power dan volume bersamaan. Selain itu keyboardnya dan kamus juga

mendapat perbaikan. Inovasi-inovasi baru di ICS antara lain penggunaan fony “Roboto.”

Di Android 4.0 Ice Cream Sandwich System Bar dan Action Bar, adanya Android 4.0 Ice Cream Sandwich voice control yang memungkinkan kita mendikte teks yang ingin kita ketik. Selain itu Face Unlock merupakan salah satu hal yang menonjol di Android versi baru ini. Juga ada NFC based app yang disebut Android Bump, yang memungkinkan kita untuk bertukar informasi/data hanya dengan menyentuhkan gadget kita.

2.2.4 The Dalvik Virtual Machine (DVM)

Android berjalan di dalam DVM bukan pada Java Virtual Machine (JVM) yang saya kira selama ini. Menurut saya banyak kesamaan antara DVM dan JVM, namun DVM memiliki feature yang lebih baik dibandingkan dengan JVM untuk perangkat mobile. Menurut buku yang saya baca DVM adalah register bases sementara JVM adalah stack based, DVM didesain dan ditulis Dan Bornsten dan beberapa engineers Google lainnya. Dalam mengatasi fungsionalitas tingkat rendah DVM menggunakan kernel Linux untuk keamanan, threading, proses dan manajemen memori. Itu memungkinkan kita menggunakan bahasa C / C++ dalam membuat aplikasi sama halnya dengan OS Linux kebanyakan. Oleh karena itu kita harus kita harus memahami arsitektur dan proses dari kernel Linux yang digunakan dalam Android tersebut.

Para pengembang tidak perlu khawatir bila ia tidak memiliki device Android, karena Android memiliki virtual machine untuk eksekusi aplikasi. DVM mengeksekusi executeable file, artinya sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangatlah kecil.

2.2.5 Android SDK (Software Development Kit)

Android SDK merupakan tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android dengan menggunakan bahasa pemrograman java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di-release oleh google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk memulai

mengembangkan aplikasi pada platform android dengan menggunakan bahasa pemrograman java. Android memberi pengembang kesempatan untuk membuat aplikasi sendiri yang bukan merupakan aplikasi bawaan Handphone / Smartphone. Untuk Source SDK android dapat dilihat dan diunduh secara gratis melalui situs resminya di <http://www.developer.android.com>.

2.2.6 ADT (Android Development Tools)

Android Development Tools (ADT) adalah plugin yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi project android, membuat GUI aplikasi, dan menambahkan komponen-komponen lainnya, begitu juga kita dapat melakukan running aplikasi Android SDK melalui eclipse. Dengan ADT kita juga dapat membuat package android (.apk) yang digunakan untuk distribusi aplikasi android.

2.2.7 Android Virtual Device (AVD)

Android Virtual Device (AVD) merupakan emulator untuk menjalankan program aplikasi android dalam komputer atau laptop. Aplikasi ini dapat dijadikan sebagai tempat test atau running aplikasi android. AVD dapat diinstall melalui eclipse sehingga setiap program android yang dibuat melalui eclipse dapat dijalankan dalam AVD.

2.2.8 Komponen Android

a. Activities

Suatu activities akan menyajikan user interface (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi Android bias jadi hanya memiliki satu activity, tetapi umumnya aplikasi memiliki banyak activity tergantung pada tujuan aplikasi dan desain aplikasi tersebut.

b. Service

Service tidak memiliki Graphic User Interface (GUI), tetapi service berjalan secara background, sebagai contoh pada saat memainkan musik kita

dapat menulis sms dalam waktu bersamaan. Untuk menjaga aplikasi musik tersebut tetap berjalan maka player (pemutar musik) menjalankan komponen yang disebut service.

c. Broadcast Receiver

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Misalnya baterai lemah, laporan sms terkirim. Broadcast Receiver tidak memiliki user interface (UI), tetapi memiliki activity untuk merespon informasi yang diterima, atau dapat menggunakan Notification Manager untuk memberitahu notifikasi kepada pengguna, seperti nyala lampu led, atau vibrating perangkat dan lain sebagainya.

d. Content Provider

Content Provider membuat kumpulan aplikasi data secara spesifik sehingga bias digunakan oleh aplikasi lain. Data disimpan dalam file sistem. Content Provider juga menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu activity, misalnya ketika menggunakan aplikasi yang membutuhkan peta (Map), atau aplikasi yang membutuhkan akses data kontak, disinilah Content Provider berperan untuk menampilkan data yang dibutuhkan tersebut.

2.3 Java



Gambar 2.4. java

Android adalah aplikasi yang dikembangkan dengan bahasa pemrograman java, sehingga sebelum kita memulai pemrograman android, komputer atau laptop kita harus terinstal program java. Java yang dapat dipergunakan adalah versi 1.5 atau versi 1.6 atau versi di atasnya untuk dapat mengompilasi aplikasi android.

2.4 Corel Draw



Gambar 2.5. Corel Draw

Corel Draw adalah sebuah program komputer yang melakukan editing pada garis vektor. Program ini dibuat oleh Corel, sebuah perusahaan software

yang berkantor pusat di Ottawa, Kanada. Corel draw memiliki kegunaan untuk mengolah gambar, oleh karena itu banyak digunakan pada pekerjaan dalam bidang publikasi atau percetakan ataupun pekerjaan di bidang lain yang membutuhkan proses visualisasi.

2.5 MIT APP INVERTOR

App Inventor adalah sebuah perangkat pemrograman untuk membuat aplikasi android, yang menyenangkan dari tool ini adalah karena berbasis Visual Block Programming. Disebut Visual Block Programming menggunakan, menyusun dan drag-drops blok, merupakan simbol-simbol perintah dan fungsi event handler tertentu dalam membuat aplikasi, dan secara sederhana kita bisa menyebutnya tanpa menuliskan kode program (coding less).

Framework visual programming ini terkait dengan bahasa pemrograman Scratch dari MIT, yang secara spesifik merupakan implementasi dari Open Block yang didistribusikan oleh MIT Scheller Teacher Education Program yang diambil dari riset yang dilakukan oleh Ricarose Roque.

App Inventor menggunakan Kawa Language Framework dan Kawa's dialect yang di dikembangkan oleh Per Bothner dan di distribusikan sebagai bagian dari GNU Operating System oleh Free Software Foundation sebagai Compiler dan menterjemahkan Visual Block Programming untuk diimplementasikan pada platform Android. Tentu disini anda tinggal menikmati kemudahan yang ada.



Gambar 2.6 Jendela kerja MIT APP INVERTOR

2.6 Memulai Eclipse

2.6.1 Untuk memulai App Inventor :

Untuk menggunakan App Inventor ada beberapa hal yang harus di siapkan terlebih dahulu, yaitu :

- Miliki Account gmail, <http://www.gmail.com>
- Masuk ke <http://beta.appinventor.mit.edu/>
- Download & Install Java 6 (1.6) dari <http://www.java.com>
- Download & Install *AppInventor_Setup_Installer_v_1_2.exe*
<http://beta.appinventor.mit.edu/learn/setup/setupwindows.html>
- Install & Setting driver HP Android apabila ada

Selanjutnya anda akan dibawa ke halaman App Inventor Setup, dimana terdapat dua tahapan dasar pada setup yaitu :

1. Pada platform windows, - *apabila anda menggunakan Linux anda bisa download App Inventor setup untuk Linux :*

- ❑ Setup Komputer, yaitu memiliki komputer PC dgn syarat untuk Windows. Sistem Operasi Windows yang didukung Windows XP, Windows Vista, Windows 7
- ❑ Telah menginstall Browser minimal Google Chrome 4.0 – rekomendasi Apple Safari 5.0 , Microsoft Internet Explorer 7, Mozilla Firefox 3.6
- ❑ Menginstall Java 6– *java 1.6. bisa juga java 1.7 yg bisa download di <http://www.java.com>.*
- ❑ Kemudian install *AppInventor_Setup_Installer_v_1_2.exe*



Gambar 2.7 App Inventor Setup

2. Setup lingkungan Developing, yaitu :

- ❑ Menggunakan Emulator,: pada penggunaan emulator tidak memerlukan setup diawal, karena tinggal diaktifkan dari blok editor.
- ❑ Menggunakan Handset Android. - Kalau tidak ingin mengesetnya kita bisa langsung melakukan developing.

2.6.2 Instalasi MIT App Inventor

Tahap ini dilakukan apabila anda ingin langsung mencoba hasil develop anda ke handset android. Pertama anda harus memiliki kabel data USB, dan menginstall driver handset Android anda. Hampir semua handset android di dukung oleh App Inventor, dan pastikan juga anda telah memiliki memory SD Card yang terpasang.

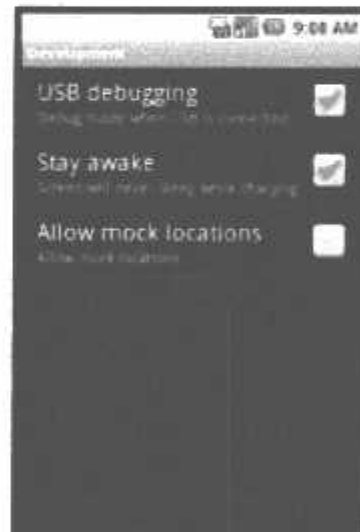
Setelah itu setting pada handset kita :

1. Masuk pada home screen
2. Pilih Setting > applications
3. Pada Unkown sources di ceklist



Gambar 2.8 gambar pada seting smartphone

4. Pilih Development, ceklist pada **USB Debugging** dan **Stay Awake**



Gambar 2.9 tampilan menu setting USB debugging

Setelah itu koneksikan Handset anda dgn kabel USB Data ke komputer – *dalam keadaan screen unlock*, hingga akan muncul dua pesan notifikasi pada atas layar yaitu :

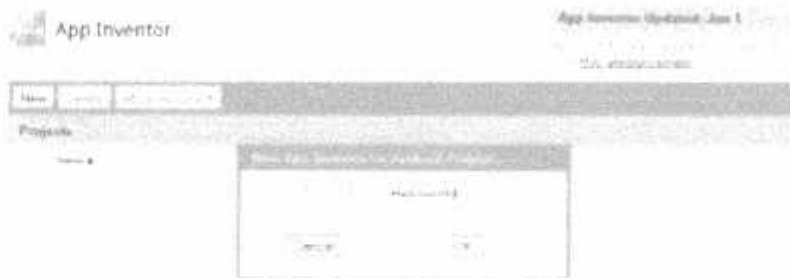
- ☐ USB Connected, yang berarti handset telah terhubung ke komputer.
- ☐ USB Debugging Connected, yang membuat App Inventor di komputer mengontrol handset.

Sampai disini berarti handset android kita telah siap untuk digunakan untuk mencoba aplikasi yang kita buat dengan App Inventor. Setelah semua konfigurasi dan setting selesai, anda bisa langsung masuk ke lingkungan developing dengan melalui <http://beta.appinventor.mit.edu/> - apabila tadi masih login anda langsung akan dibawa ke jendela developing :



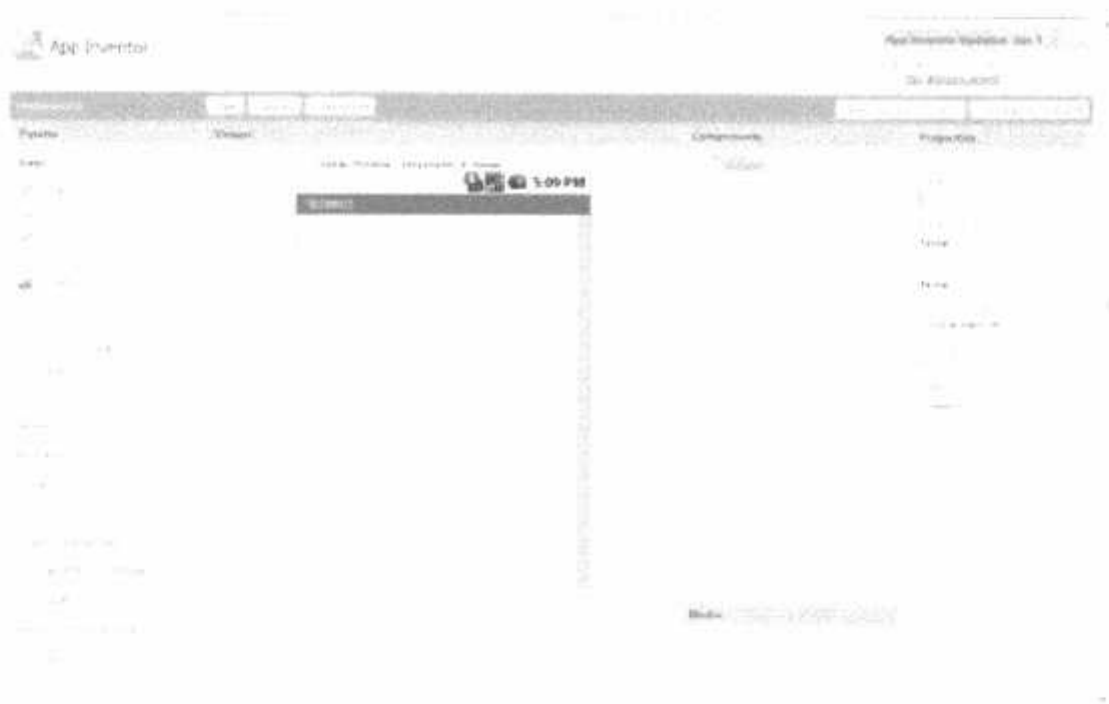
Gambar 2.10 Jendela Developing

5. Klik pada New, dan mulai projek baru misal dengan nama Helloworld, kemudian klik pada tombol OK



Gambar 2.11 Membuat Projek Baru

6. Maka akan terlihat langsung proyek kita



Gambar 2.12 Projek Baru

2.5.3 Lingkungan Kerja App Inventor

Sebelum kita melanjutkan untuk memulai developing, kita mengenal terlebih dahulu lingkungan kerja pada App Inventor yang terdiri dari :

1. **Komponen Desainer** – *Component Designer* yang berjalan pada Browser, digunakan untuk memilih komponen dan mengatur property.



Gambar 2.13 Komponen desainer

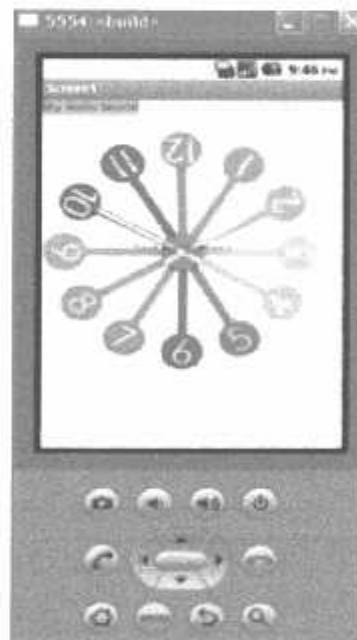
Komponen desainer , terdiri dari :

- a. **Viewer** : untuk menempatkan komponen dan mengaturnya sesuai tampilan yang diinginkan.
 - b. **Pallette** : adalah list dari komponen yang bisa dipakai
 - c. **Component list** : merupakan tempat list komponen dari proyek yang kita pakai
 - d. **Media** : mengambil media audio dan gambar untuk proyek kita
 - e. **Properties** : ketika anda klik komponen pada viewer maka propertiesnya akan terlihat pada panel ini
2. **Blok Editor – Blocks Editor** berjalan diluar browser, dimana digunakan untuk membuat dan mengatur behaviour dari komponen yang kita pilih pada komponen desainer.



Gambar 2.14 Blocks Editor

3. **Emulator** , digunakan untuk menjalankan dan mengetest aplikasi yang kita bangun. Ini sangat bermanfaat apabila kita belum menggunakan handset langsung, karena Emulator Android ini telah terintegrasi dengan baik.



Gambar 2.15 Emulator

Ketiga Lingkungan kerja diatas, akan bisa tampil secara bersamaan dan secara bergantian juga berpindah kita akan menggunakannya dalam membuat aplikasi android dengan App Android.

BAB III

ANALISA DAN PERANCANGAN SISTEM

Bab ini berisi penjelasan mengenai analisis dan perancangan sistem aplikasi. Analisis desain sistem digunakan untuk memberikan gambaran secara umum terhadap aplikasi. Hal ini berguna untuk menunjang perancangan aplikasi yang akan dikembangkan sehingga kebutuhan akan aplikasi tersebut dapat diketahui sebelumnya. Kemudian hasil analisis akan menjadi dasar untuk melakukan perancangan atau desain aplikasi sesuai kebutuhan sistem.

3.1. Analisa Sistem

3.1.1. Deskripsi Sistem

Aplikasi ini memiliki fungsi utama untuk mengetahui lokasi atau letak smartphone Android kita. Dengan menampilkan lokasinya berupa titik koordinat GPS yaitu Latitude, Longitude, dan Altitude.

Pengaktifan GPS antara GPS standard dan A-GPS akan berlangsung secara otomatis pada saat aplikasi ini dijalankan. Jika letak geografis smartphone mendukung pengaktifan GPS standar maka system yang bekerja adalah sistem GPS standar. Tetapi, jika sinyal mikro yang di pancarkan oleh satelit GPS terhalangi (missal kita berada dalam gedung atau, di tengah kota dimana terhalang oleh gedung-gedung tinggi, atau cuaca yang buruk) maka sistem A-GPS yang akan di aktifkan.

Selain itu aplikasi ini juga akan mengirimkan titik koordinat tersebut melalui sms. Dengan sistem auto-reply sms, aplikasi telah disetting agar dapat langsung mengirimkan sms lokasi kepada nomer telpon yang meminta lokasi tersebut.

3.1.2. Analisa Kebutuhan Sistem

Untuk perangkat yang akan digunakan merancang aplikasi ini

adalah sebagai berikut :

1. Laptop Intel Core2Duo 2.10Ghz.
2. RAM DDR2 2 *Gigabyte* (GB).
3. Hardisk 640 *Gigabyte*.
4. VGA card NVIDIA GeForce 310M 1Gb

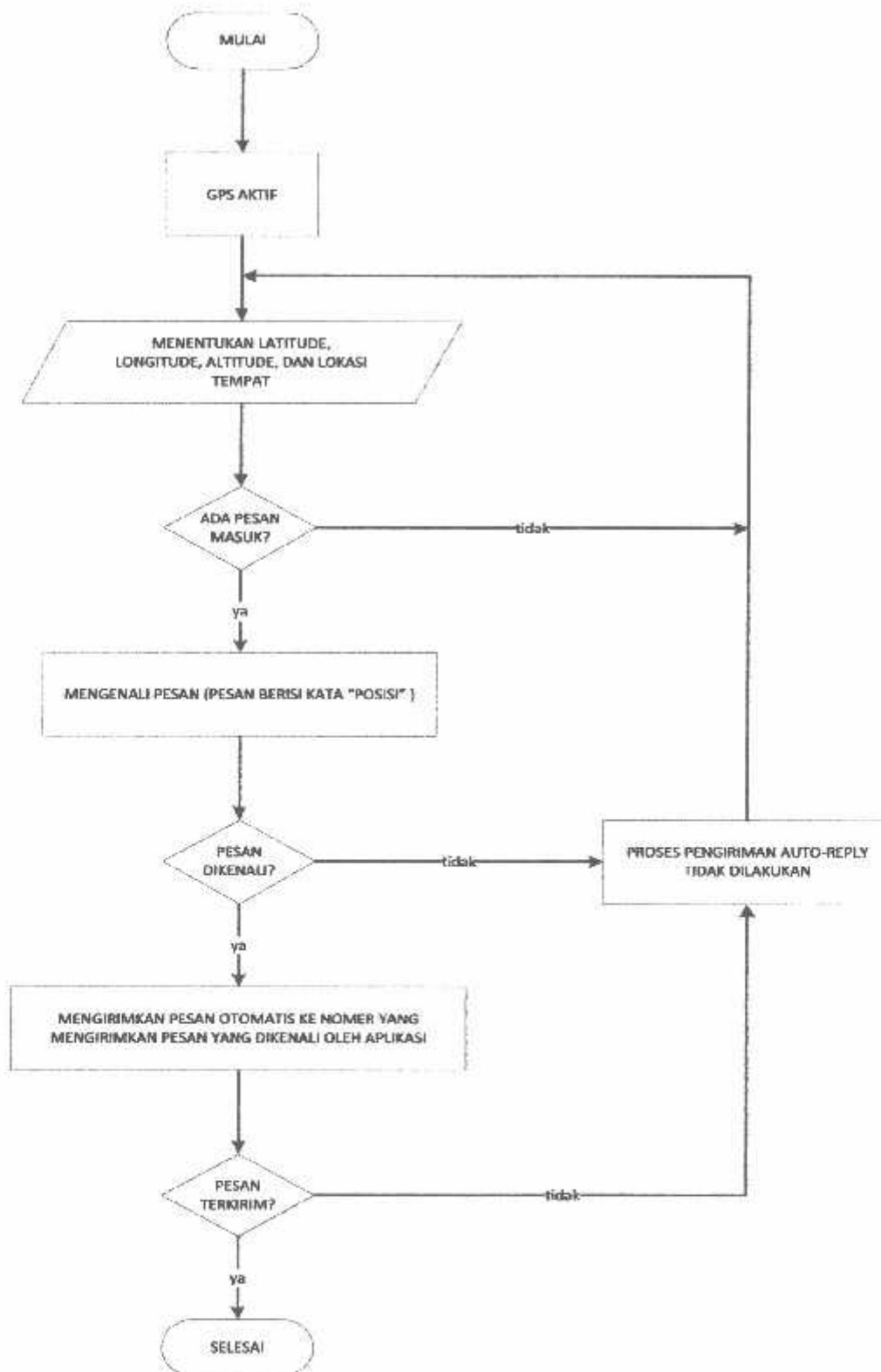
Selain perangkat keras yang digunakan sebagai pembuatan aplikasi, membutuhkan spesifikasi perangkat-perangkat lunak sebagai berikut:

1. Sistem Operasi Windows 7 32bit.
2. MIT App Inventor
3. Java version 7 update 21.
4. Android ADT r21.
5. Android SDK 1.5 (API 3) - 4.2 (API 17).

3.2. Alur Sistem

3.2.1. Flowchart Sistem

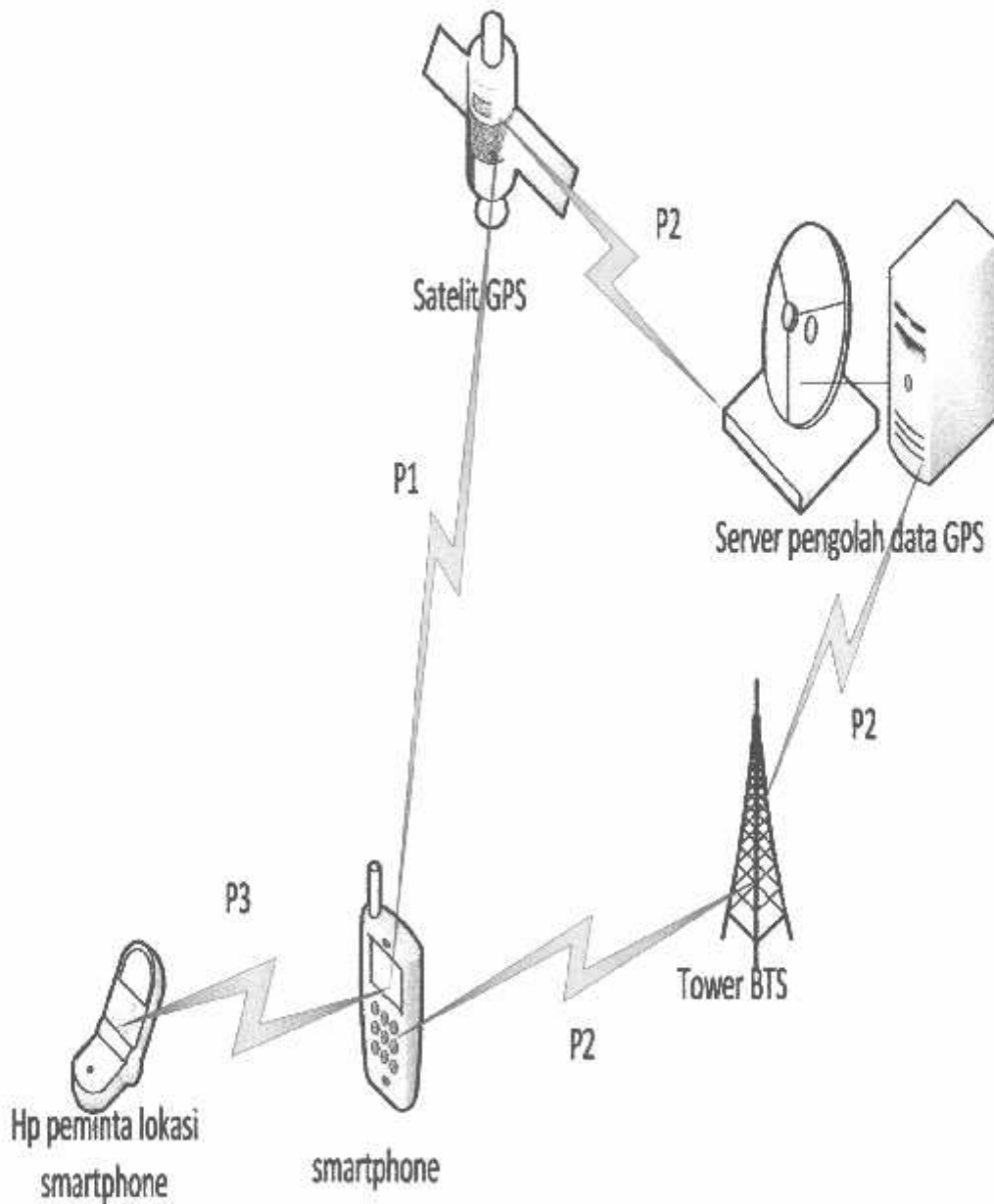
Pada bagian ini akan dijelaskan mengenai desain alur sitem dari aplikasi. Dcsain alur sistem dapat dilihat pada gambar 3.1.



Gambar 3.1. Flowchart Sistem

Dari Flowchart (Gambar 3.1) dapat dijelaskan bahwa pada mulanya aplikasi akan mengaktifkan fitur GPS pada smartphone android, setelah itu

aplikasi tersebut akan mendeteksi lokasinya. Setelah lokasi yang berupa titik koordinat tersedia maka smartphone akan menunggu permintaan berupa sms yang akan diterima smartphone. Sms yang dikenali oleh aplikasi, (berupa sms dengan kata kunci "LOKASI") akan melakukan pengiriman otomatis ke nomer telepon yang mengirim sms (LOKASI) tadi. Sms yang dikirimkan berupa lokasi Latitude, Longitude, dan Altitude. Berikut ini adalah diagram alur aplikasi tersebut baik saat menggunakan GPS atau A-GPS :



Gambar 3.2 Diagram Alur Aplikasi

Brikut ini adalah tabel yang menjelaskan diagram alur diatas:

No	Alur	Keterangan
P1	GPS	System GPS standar. System GPS standar akan berjalan jika sinyal mikro yang di pancarkan oleh satelit, dapat di terima baik oleh hp.
P2	A-GPS	System A-GPS. Akan bekerja jika system GPS tidak berjalan dengan baik. Karena sinyal mikro terhalang oleh gedung, cuaca yang buruk, pepohonan dan lain-lain.
P3	Request (meminta) lokasi dan Reply (membalas) permintaan lokasi	User HP yang ingin mengetahui lokasi smartphone android. Akan mengirimkan sms ke nomer smartphone. Dan smartphone tersebut akan melakukan autoreply sms ke nomer yang meminta lokasinya.

Tabel 3.1 Keterangan Diagram Alur

3.3 Perancangan Aplikasi

Perancangan aplikasi merupakan langkah awal dari pembuatan aplikasi ini. Mulai dari menentukan tampilan dan tataletak tombol hingga script dari setiap komponen yang akan dijalankan.

Beberapa tahapan dalam perancangan aplikas ini adalah sebagai berikut :

1. Pendesainan Tampilan

Pendesainan tampilan aplikasi ini menggunakan software *CorelDraw*, desain yang dibuat menggunakan software ini adalah, tampilan latar belakang aplikasi serta tampilan latar belakang tombol yang ada pada aplikasi ini.

2. Pengkodean Program

Pengkodean aplikasi ini dilkaukan dengan menggunakan software *MIT App Invertor* yang memang dirancang untuk membuat aplikasi android, sehinggann dapat meng-eksekusi bahasa pemrograman android.

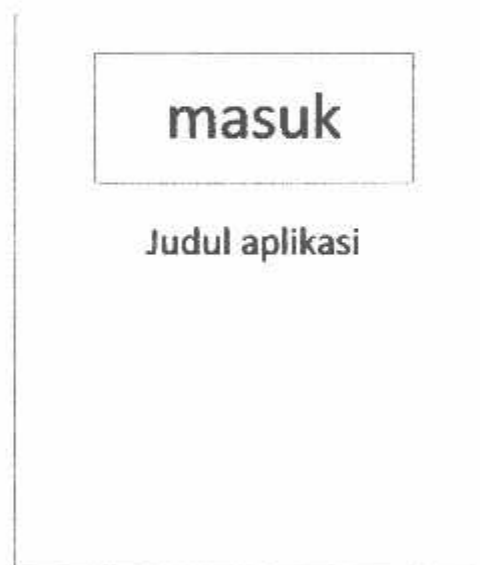
3.4. Desain Alur Menu

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

1. Menu Utama, merupakan tampilan menu awal pada aplikasi ini. Menu Utama tampilan judul aplikasi, sebelum menuju ke layar selanjutnya.
2. Layar untuk menampilkan keterangan yang merupakan pilihan pengaktifan fitur GPS dan mendeteksi titik koordinat..
3. Setelah fitur GPS aktif aplikasi memerlukan waktu beberapa detik untuk menampilkan lokasi titik koordinat, yang berisi nilai Latitude, Longitude, dan Altitud pada layar berikutnya.

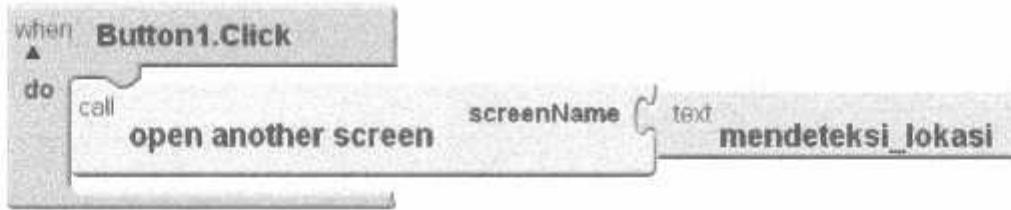
3.5 Desain Antarmuka Aplikasi

3.5.1 Desain Tampilan



Gambar 3.3. Desain tampilan awal aplikasi

Gambar 3.3 merupakan tampilan awal saat membuka aplikasi ini. Hanya terdapat sebuah tombol untuk masuk ke layar selanjutnya. Juga terdapat judul nama aplikasi.



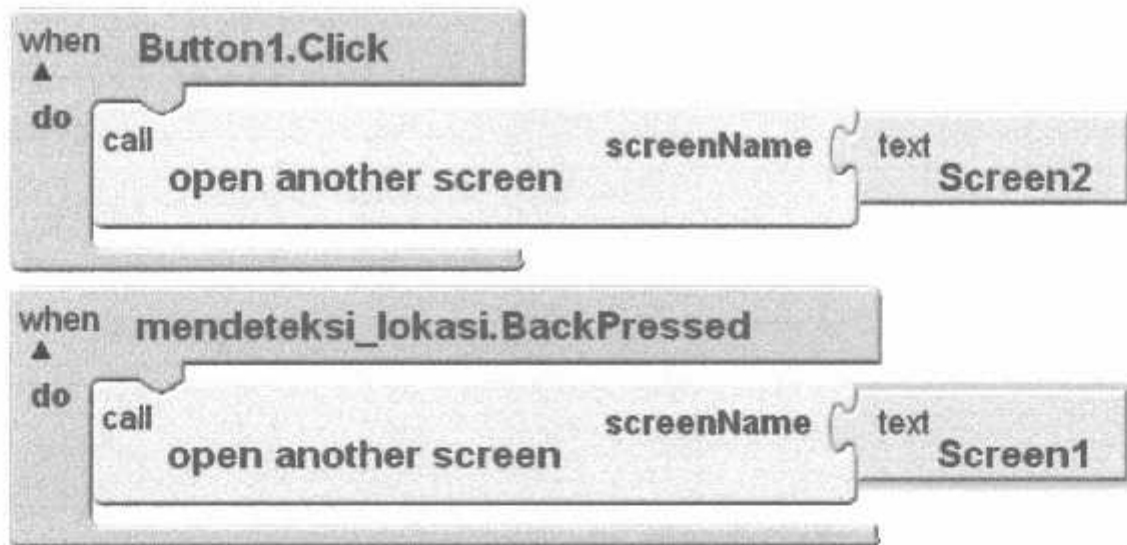
Gambar 3.4 Block Edit tampilan awal

Gambar 3.4 adalah Block Edit pada App inventor. Bagaimana me-link-kan tombol ke layar selanjutnya.



Gambar 3.5. Desain tampilan layar ke-2

Gambar 3.5 merupakan gambar dari tampilan menu pilihan apakah akan melanjutkan ke layar selanjutnya atau tidak. Pada layar ini terdapat sebuah tombol, di mana saat kita menekan tombol tersebut kita akan melakukan pengaktifan otomatis pada fitur GPS yang terdapat pada smartphone kita. Juga merupakan perintah ketika kita akan memasuki layar selanjutnya dan menunggu perintah selanjutnya untuk memliskuqn perintah auto-reply SMS.



Gambar 3.6 Block Edit Layar Ke-2

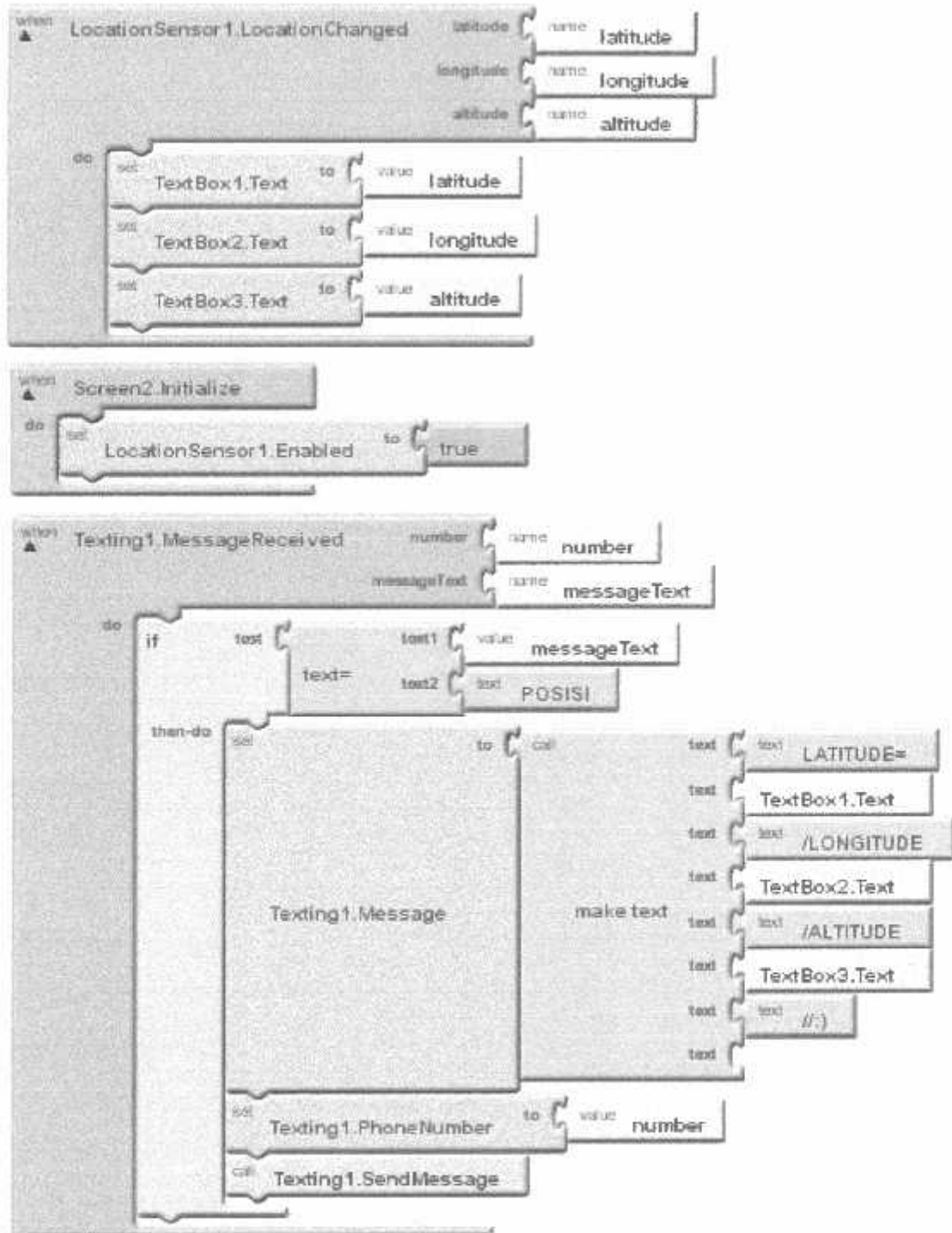
Gambar 3.6 merupakan block edit MIT app Inventor. Terdapat 2 perintah yang pertama ketika kita menekan tombol Mendeteksi Lokasi kita akan masuk ke layar yang merupakan inti aplikasi. Dan yang kedua ada lah perintah saat kita akan menekan tombol back pada smartphone android, kita akan kembali ke tampilan awal.

latitude	<input type="text"/>
longitude	<input type="text"/>
altitude	<input type="text"/>

Gambar 3.7. Desain Layar Titik Koordinat

Gambar 3.13 merupakan gambar dari tampilan utama. Di layar ini smartphone kita berada dalam mode stand-by. Dimana aplikasi telah siap untuk menerima perintah saat ada SMS dengan kata kunci LOKASI masuk ke smartphone tersebut. Di layar ini akan menampilkan titik koordinat lokasi kita

berupa koordinat Latitude, Longitude, dan Altitude. Karena bersifat monitoring, nilai pada masing-masing titik koordinat akan selalu berubah-ubah jika samrtphon kita berpindah lokasi. saat aplikasi minimize di layar ini, aplikasi tetap stand-by untuk menerima SMS.



.gambar 3.8 Block Editor Inti Aplikasi

Gambar 3.8 merupakan Block Edit layar tampilan utama, dimana terdapat perintah auto-reply, monitoring lokasi, dan juga perintah isi SMS yang akan di kirimkan ke hp peminta lokasi. Disini juga merupakan tempat memasukkan perintah untuk mengenali SMS yang hanya berisi kata LOKASI, untuk mengirimkan koordinat lokasi smartphone.

BAB IV

IMPLEMENTASI DAN PENGUJIAN APLIKASI

4.1. Implementasi Sistem

Untuk melakukan pengujian aplikasi maka dibutuhkan software dan hardware sebagai berikut :

1. Smartphone dengan operating system android.
2. HP berbagai jenis yang mendukung fitur SMS untuk melakukan pengiriman dan menerima SMS.
3. Operating system yang dibutuhkan adalah versi 2.3.6 Gingerbread atau operating system versi di atasnya lainnya.

Dalam pengujian aplikasi ini menggunakan smartphone LG L3 E400 yang menggunakan android versi 2.3.6 Gingerbread dan HP Cross tipe GG58CT.

4.2. Pengujian Aplikasi

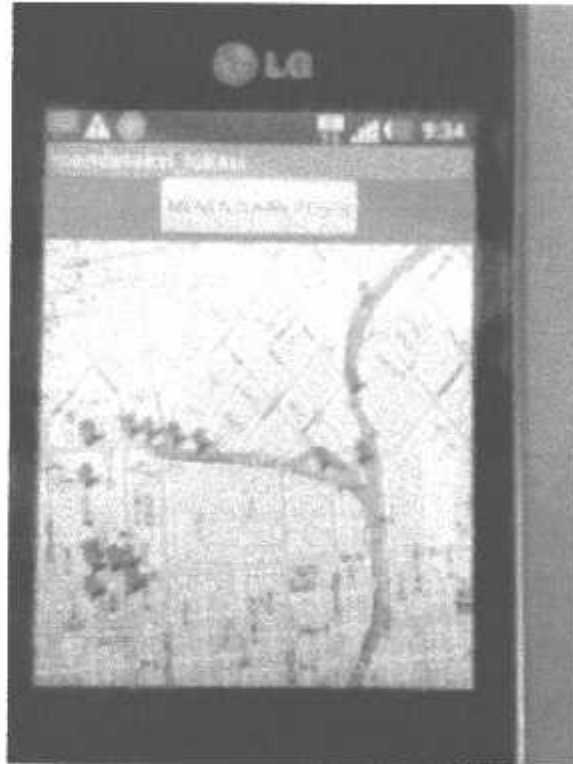
4.2.1. Tampilan Awal



Gambar 4.1 Tampilan awal aplikasi

Tampilan awal aplikasi akan muncul ketika aplikasi pertama kali dijalankan. Proses yang terjadi pada awal aplikasi adalah pembuatan database yang digunakan untuk kamus istilah seperti gambar 4.1. kemudian akan terjadi proses starting aplikasi sebagai pemberitahuan aplikasi segera berjalan.

4.2.2. Tampilan Menu



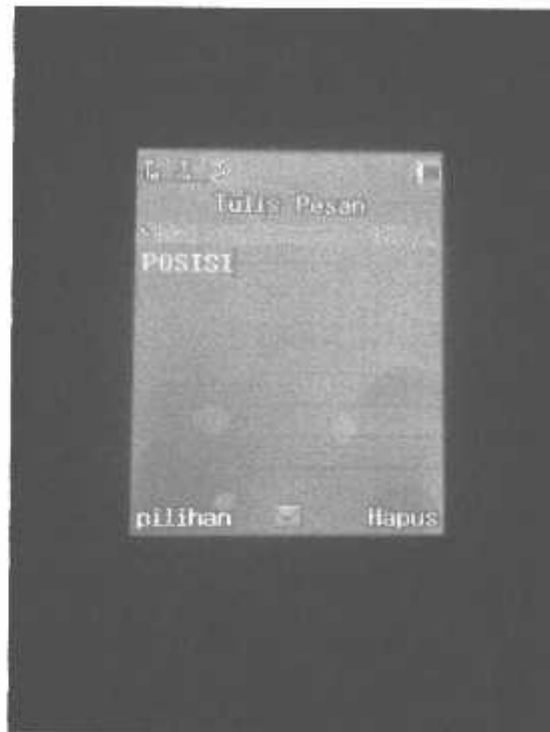
Gambar 4.2. tampilan menu utama

Gambar 4.2 merupakan gambar tampilan konfirmasi untuk mengaktifkan beberapa fitur yang ada di hp android. Diantaranya fitur GPS/A-GPS, fitur GPRS. Dengan menekan tombol MENENTUKAN LOKASI semua fitur yang di butuhkan akan di jalankan.



Gambar 4.3. Tampilan menu utama

Gambar 4.3. tampilan menu utama, setelah GPS/A-GPS diaktifkan, dibutuhkan waktu beberapa detik agar aplikasi dapat mengambil titik koordinat yang sudah terdeteksi oleh GPS/A-GPS. Aplikasi bersistem monitoring, jadi angka yang terdapat pada kotak dialog akan berubah-ubah jika smartphone berpindah tempat (mengikuti lokasi smartphone). Dilayar ini, jika user langsung menekan tombol home (mode minimize), maka aplikasi akan tetap berjalan. Dan akan tetap mendeteksi lokasi tersebut selama fitur GPS/A-GPS dan GPRS tetap berjalan.



Gambar 4.4. Tampilan SMS meminta lokasi

Gambar 4.4 merupakan gambar tampilan dari HP Cross GG580CT mengirim pesan ke nomer smartphon dengan isi pesan LOKASI (kata pesan yang di kenali oleh aplikasi tersebut).



Gambar 4.5. Tampilan sms balasan

Gambar 4.5 balasan yang telah dikirim oleh smartphone android, secara otomatis. Pesan yang dikirim sesuai dengan koordinat lokasi smartphone pada saat menerima SMS. pesan tersebut langsung dikirim oleh nomer yang ada ada di smartphone tersebut. Pesan akan tetap terkirim walaupun aplikasi dalam mode minimize. Membutuhkan waktu sekitar 10 – 75 detik agar pesan LOKASI mendapat balasan berupa titik koordinat lokasi smartphone, tergantung hubungan antar operator dan ketersediaan sinyal yang ada.



(a) Tampilan pada smartphone



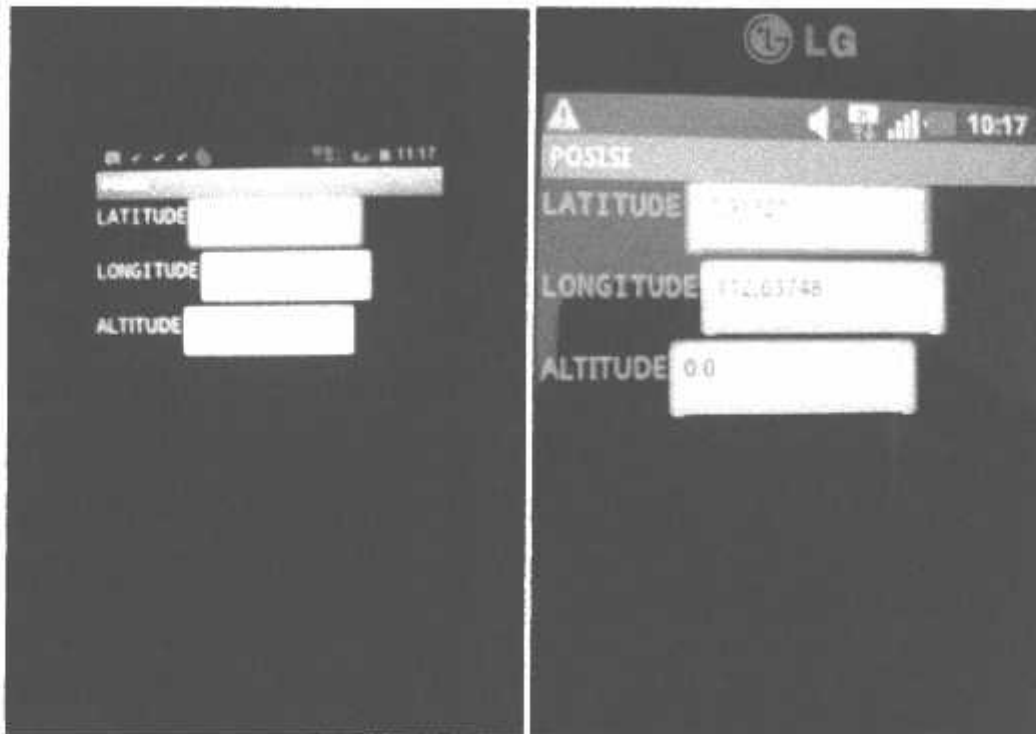
(b) Tampilan SMS Yang dikirim

Gambar 4.6. Tampilan pada hp smartphone dan sms yang dikirimkan

Gambar 4.6 merupakan gambar dari tampilan utama pada smartphone android (a). Dan tampilan SMS di HP Cross GG580CT. Lokasi yang dikirimkan sama persis dan tidak berubah, selama smartphone android belum berubah tempat.

4.3. Membandingkan Aplikasi Terhadap Smartphone Berbeda Operating Sistem

Aplikasi ini pada dasarnya dapat dipergunakan pada Smartphone Android versi dibawah 4.0. akan tetapi versi dibawah 4.0 tidak dapat membaca inputan huruf arab dengan benar.



(a) Tampilan pada Gingerbread

(b) Tampilan pada ICS

Gambar 4.15 Android beda versi

Pada gambar 4.15 baik pada gambar (a) maupun (b), tidak terjadi perbedaan. Karena kedua OS tersebut mendukung untuk menjalankan GPS maupun system auto-reply SMS.

4.4. Tabel Pengujian

Lokasi peminta lokasi	Lokasi smartphone android	Nama tempat yang terdeteksi	Perkiraan kelebihan jarak (km)	Auto-reply SMS berjalan?
Wagir	Kacuk Latitude: -8.020146, Longitude: 112.631105	Kebonsari Latitude: -8.019583, Longitude: 112.619432	0,3 km	ya
Wagir	pasar comboran latitude: -7992332, longitude: 112.621105	Mergosono Latitude: -7.993341, Longitude: 112.633272	0,5 km	ya
wagir	Stasiun kota baru Latitude: -7.977712, Longitude: 112.636952	Klojen Latitude: -7.976416, Longitude: 112.636845	0,5 km	ya
wagir	Pabrik bentoel Latitude: -7.916848 Longitude: 112.654386	Karanglo Latitude: -7.9104, Longitude: 112.64425	1 km	ya
wagir	Kampus 2 ITN Latitude: -7.916173, Longitude: 112.634662	Karanglo Latitude: -7.91045, Longitude: 112.64425	2 km	ya

Table 4.1. Hasil pengujian di beberapa tempat

Dari table 4.1 diatas aplikasi dapat berjalan sebagaimana mestinya. Terdapat ketidak akuratan lokasi yang merujuk pada lokasi BTS yang dikenali oleh smartphone android.

Smartphone yang menjalankan aplikasi	Nama HP/Smartphone peminta lokasi dan OS-nya	Auto-reply SMS berjalan?
Smartphone android LG L3 E400 dengan OS android versi 2.3 (gingerbread)	Cross GG50CT (tidak memiliki OS)	Ya
	Blackberry style 9670 (blackberry OS 6.0)	Ya
	Nokia Lumia 520 (Microsoft Windows Phone 8)	Ya
	Nokia N70 (Symbian OS 8.1)	Ya

Tabel 4.2 Pengujian untuk HP peminta lokasi

Pada Tabel 4.2 adalah pengujian untuk beberapa jenis handphone/smartphone dengan berbeda OS yang mendukung fitur SMS. Auto-reply sms berjalan dengan baik, dan waktu untuk prosesnya memakan waktu 0-2 menit tergantung sinyal dan operator yang digunakan. Dimana antar operator seluler memiliki kelancaran/kecepatan komunikasi SMS yang berbeda-beda.

Ketersediaan pulsa	Kekuatan sinyal	A-GPS	GPS	Auto-reply berjalan?
Ada	Penuh 3G/HSDPA	Ya	Ya	Ya
Ada	Lemah EDGE/GPRS	Ya	Tidak	Ya
Ada	Tidak ada sinyal	Ya	Ya	Tidak / pending (akan terkirim saat mendapatkan sinyal)
Tidak ada	Penuh 3G/HSDPA	Ya	Ya	Tidak

Table 4.3 pengujian untuk tiap kondisi saat akan melakukan proses Auto-reply sms

Dari table 4.3 dapat disimpulkan bahwa, aplikasi akan berjalan jika memenuhi syarat-syarat untuk terjadinya auto-reply sms. Seperti ketersediaan pulsa, kekuatan sinyal, dan fitur A-GPS/GPS yang ada di smartphone android. Di beberapa kondisi saat kekuatan sinyal lemah/tidak ada sinyal, auto-reply tidak akan berjalan, atau di beberapa kali percobaan yang penulis lakukan akan terjadi pending atau selang waktu sampai smartphone mendapatkan sinyal yang cukup untuk melakukan auto-reply sms.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah melakukan segala perencanaan dan pembuatan aplikasi serta hasil pengujian, maka dalam laporan skripsi ini dapat disimpulkan bahwa :

1. Dengan menggunakan aplikasi ini dapat mengetahui lokasi smartphone android.
2. Aplikasi ini menggunakan fitur GPS/A-GPS yang tertanam di Smartphone Android.
3. Aplikasi tetap akan bisa digunakan sebagaimana mestinya dengan syarat tetap berjalan seperti, smartphone lokasi aktif, komunikasi data (GPRS) aktif, nomor telepon yang ada dalam smartphone diketahui oleh peminta lokasi.
4. Saat aplikasi ini dijalankan, aplikasi akan terus mengupdate lokasi dan lokasi akan terus berubah-ubah sesuai lokasi smartphone.
5. Ukuran file ketika diexport menjadi file .APK adalah sekitar 2,03 Mb.
6. Dalam pengujian aplikasi, karena menggunakan sistem A-GPS sehingga tingkat keakuratannya tidak seakurat GPS satndar. Dengan factor kesalahan sebesar 0-2km, tergantung menara BTS yang terdekat.

5.2. Saran

Saran yang dapat disampaikan untuk pengembangan aplikasi ini adalah :

1. Dapat mengetahui keberadaan smartphone asal aplikasi tetap dijalankan dan komunikasi data tetap berjalan serta fitur GPS/A-GPS active.
2. Dapat mengetahui lokasi orang yang ingin kita ketahui keberadaannya selama syarat berjalannya aplikasi ini berjalan.
3. Dapat belajar membuat aplikasi android dengan mudah tanpa koding, dengan menggunakan MIT App Inventor.
4. Mengetahui lebih jauh fitur GPS/A-GPS yang tertanam di smartphone android

DATAR PUSTAKA

1. Mulyadi. 2011. *Android App Inventor*. Yogyakarta: Multimedia Center Publisng.
2. Safaat H, 2011, Nazarudin. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
3. Leo Adi, (2012)*Rancang Bangun Game Edukasi Bahasa Inggris Menggunakan App Inventor*. Institut Teknologi Surabaya
4. Wildan H, (2011) *Pembangunan Sistem Pelacakan Dan Penelusuran Device Mobile Berbasis GPS Pada Smartphone Android*. Amikom Yogyakarta
5. Mulyadi. 2010. *Membuat Aplikasi untuk Android*. Yogyakarta: Multimedia Center Publishing
6. <http://appinventor.mit.edu> Diakses 24 April 2013
7. <http://appinventor.mit.edu/explore/tutorials.html> Diakses 26 April 2013

```

package appinventor.ai_aggina180818.location_detector;

import com.google.appinventor.components.runtime.Component;
import gnu.expr.ModuleBody;
import gnu.expr.ModuleMethod;
import gnu.mapping.CallContext;
import gnu.mapping.Symbol;
import gnu.mapping.Values;
import gnu.mapping.WrongType;

public class mendeteksi_lokasi$frame
    extends ModuleBody
{
    mendeteksi_lokasi $main;

    public Object apply0(ModuleMethod paramModuleMethod)
    {
        switch (paramModuleMethod.selector)
        {
            default:
                return super.apply0(paramModuleMethod);
            case 14:
                return mendeteksi_lokasi.lambda2();
            case 15:
                this.$main.$define();
                return Values.empty;
            case 16:
                return mendeteksi_lokasi.lambda3();
            case 17:
                return this.$main.mendeteksi_lokasi$BackPressed();
            case 18:
                return mendeteksi_lokasi.lambda4();
            case 19:
                return mendeteksi_lokasi.lambda5();

```

```

    }
    return this.$main.Button1$Click();
}

public Object apply1(ModuleMethod paramModuleMethod, Object paramObject)
{
    mendeteksi_lokasi localmendeteksi_lokasi2;
    switch (paramModuleMethod.selector)
    {
        case 2:
        case 4:
        case 6:
        case 7:
        case 8:
        case 9:
        default:
            return super.apply1(paramModuleMethod, paramObject);
        case 1:
            this.$main.androidLogForm(paramObject);
            return Values.empty;
        case 3:
            localmendeteksi_lokasi2 = this.$main;
    }
    try
    {
        Symbol localSymbol2 = (Symbol)paramObject;
        return localmendeteksi_lokasi2.lookupInFormEnvironment(localSymbol2);
    }
    catch (ClassCastException localClassCastException2)
    {
        mendeteksi_lokasi localmendeteksi_lokasi1;
        throw new WrongType(localClassCastException2, "lookup-in-form-environment", 1,
paramObject);
    }
}

```

```

localmendeteksi_lokasi1 = this.$main;
try
{
    Symbol localSymbol1 = (Symbol)paramObject;
    if (localmendeteksi_lokasi1.isBoundInFormEnvironment(localSymbol1)) {
        return Boolean.TRUE;
    }
    return Boolean.FALSE;
}
catch (ClassCastException localClassCastException1)
{
    throw new WrongType(localClassCastException1, "is-bound-in-form-environment",
1, paramObject);
}
this.$main.addToFormDoAfterCreation(paramObject);
return Values.empty;
this.$main.processException(paramObject);
return Values.empty;
}

```

```

public Object apply2(ModuleMethod paramModuleMethod, Object paramObject1,
Object paramObject2)
{
    mendeteksi_lokasi localmendeteksi_lokasi3;
    switch (paramModuleMethod.selector)
    {
        case 4:
        case 5:
        case 8:
        case 10:
        case 11:
        case 12:
        default:
            return super.apply2(paramModuleMethod, paramObject1, paramObject2);
    }
}

```

```

case 2:
    localmendeteksi_lokasi3 = this.$main;
}
try
{
    Symbol localSymbol3 = (Symbol)paramObject1;
    localmendeteksi_lokasi3.addToFormEnvironment(localSymbol3, paramObject2);
    return Values.empty;
}
catch (ClassCastException localClassCastException3)
{
    mendeteksi_lokasi localmendeteksi_lokasi2;
    throw new WrongType(localClassCastException3, "add-to-form-environment", 1,
paramObject1);
}
localmendeteksi_lokasi2 = this.$main;
try
{
    Symbol localSymbol2 = (Symbol)paramObject1;
    return localmendeteksi_lokasi2.lookupInFormEnvironment(localSymbol2,
paramObject2);
}
catch (ClassCastException localClassCastException2)
{
    mendeteksi_lokasi localmendeteksi_lokasi1;
    throw new WrongType(localClassCastException2, "lookup-in-form-environment", 1,
paramObject1);
}
localmendeteksi_lokasi1 = this.$main;
try
{
    Symbol localSymbol1 = (Symbol)paramObject1;
    localmendeteksi_lokasi1.addToGlobalVarEnvironment(localSymbol1,
paramObject2);

```

```

        return Values.empty;
    }
    catch (ClassCastException localClassCastException1)
    {
        throw new WrongType(localClassCastException1, "add-to-global-var-environment",
1, paramObject1);
    }
    this.$main.addToEvents(paramObject1, paramObject2);
    return Values.empty;
    this.$main.addToGlobalVars(paramObject1, paramObject2);
    return Values.empty;
    return this.$main.lookupHandler(paramObject1, paramObject2);
}

/* Error */
public Object apply4(ModuleMethod paramModuleMethod, Object paramObject1,
Object paramObject2, Object paramObject3, Object paramObject4)
{
    // Byte code:
    // 0: aload_1
    // 1: getfield 18      gnu/expr/ModuleMethod:selector      I
    // 4: lookupswitch    default:+28->32, 8:+40->44, 12:+57->61
    // 33: aload_1
    // 34: aload_2
    // 35: aload_3
    // 36: aload 4
    // 38: aload 5
    // 40: invokespecial 129    gnu/expr/ModuleBody:apply4
        (Lgnu/expr/ModuleMethod;Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Obj
ct;Ljava/lang/Object;)Ljava/lang/Object;
    // 43: areturn
    // 44: aload_0

```

```

// 45: getfield 28
    appinventor/ai_aggina180818/location_detector/mendeteksi_lokasi$frame:$main
    Lappinventor/ai_aggina180818/location_detector/mendeteksi_lokasi;
// 48: aload_2
// 49: aload_3
// 50: aload 4
// 52: aload 5
// 54: invokevirtual 133
    appinventor/ai_aggina180818/location_detector/mendeteksi_lokasi:addToCompo
nents (Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;Ljava/lang/Object;)V
// 57: getstatic 37    gnu/mapping/Values:empty    Lgnu/mapping/Values;
// 60: areturn
// 61: aload_0
// 62: getfield 28
    appinventor/ai_aggina180818/location_detector/mendeteksi_lokasi$frame:$main
    Lappinventor/ai_aggina180818/location_detector/mendeteksi_lokasi;
// 65: astore 6
// 67: aload_2
// 68: checkcast 135 com/google/appinventor/components/runtime/Component
// 71: astore 8
// 73: aload_3
// 74: checkcast 137 java/lang/String
// 77: astore 10
// 79: aload 4
// 81: checkcast 137 java/lang/String
// 84: astore 12
// 86: aload 5
// 88: checkcast 139 [Ljava/lang/Object;
// 91: astore 14
// 93: aload 6
// 95: aload 8
// 97: aload 10
// 99: aload 12
// 101: aload 14

```

```

// 103: invokevirtual 143
    appinventor/ai_aggina180818/location_detector/mendeteksi_lokasi:dispatchEven
t
    (Lcom/google/appinventor/components/runtime/Component;Ljava/lang/String;LJ
ava/lang/String;[Ljava/lang/Object;)Z
// 106: ifeq +7 -> 113
// 109: getstatic 78  java/lang/Boolean:TRUE      Ljava/lang/Boolean;
// 112: areturn
// 113: getstatic 81  java/lang/Boolean:FALSE      Ljava/lang/Boolean;
// 116: areturn
// 117: astore 7
// 119: new 89      gnu/mapping/WrongType
// 122: dup
// 123: aload 7
// 125: ldc 144
// 127: iconst_1
// 128: aload_2
// 129: invokespecial 94  gnu/mapping/WrongType:<init>
    (Ljava/lang/ClassCastException;Ljava/lang/String;ILjava/lang/Object;)V
// 132: athrow
// 133: astore 9
// 135: new 89      gnu/mapping/WrongType
// 138: dup
// 139: aload 9
// 141: ldc 144
// 143: iconst_2
// 144: aload_3
// 145: invokespecial 94  gnu/mapping/WrongType:<init>
    (Ljava/lang/ClassCastException;Ljava/lang/String;ILjava/lang/Object;)V
// 148: athrow
// 149: astore 11
// 151: new 89      gnu/mapping/WrongType
// 154: dup
// 155: aload 11

```

```

// 157: ldc 144
// 159: iconst_3
// 160: aload 4
// 162: invokespecial 94    gnu/mapping/WrongType:<init>
//      (Ljava/lang/ClassCastException;Ljava/lang/String;ILjava/lang/Object;)V
// 165: athrow
// 166: astore 13
// 168: new 89    gnu/mapping/WrongType
// 171: dup
// 172: aload 13
// 174: ldc 144
// 176: iconst_4
// 177: aload 5
// 179: invokespecial 94    gnu/mapping/WrongType:<init>
//      (Ljava/lang/ClassCastException;Ljava/lang/String;ILjava/lang/Object;)V
// 182: athrow
// Local variable table:
// start    length  slot   name  signature
// 0 183    0      this   frame
// 0 183    1      paramModuleMethod  ModuleMethod
// 0 183    2      paramObject1  Object
// 0 183    3      paramObject2  Object
// 0 183    4      paramObject3  Object
// 0 183    5      paramObject4  Object
// 65     29     6      localmendeteksi_lokasi  mendeteksi_lokasi
// 117    7      7      localClassCastException1  ClassCastException
// 71     25     8      localComponent      Component
// 133    7      9      localClassCastException2  ClassCastException
// 77     21    10     str1    String
// 149    7      11     localClassCastException3  ClassCastException
// 84     16    12     str2    String
// 166    7      13     localClassCastException4  ClassCastException
// 91     11    14     arrayOfObject  Object[]
// Exception table:

```

```

// from    to    target type
// 67      73    117  java/lang/ClassCastException
// 73      79    133  java/lang/ClassCastException
// 79      86    149  java/lang/ClassCastException
// 86      93    166  java/lang/ClassCastException
}

```

```

public int match0(ModuleMethod paramModuleMethod, CallContext
paramCallContext)

```

```

{
    switch (paramModuleMethod.selector)
    {
    default:
        return super.match0(paramModuleMethod, paramCallContext);
    case 20:
        paramCallContext.proc = paramModuleMethod;
        paramCallContext.pc = 0;
        return 0;
    case 19:
        paramCallContext.proc = paramModuleMethod;
        paramCallContext.pc = 0;
        return 0;
    case 18:
        paramCallContext.proc = paramModuleMethod;
        paramCallContext.pc = 0;
        return 0;
    case 17:
        paramCallContext.proc = paramModuleMethod;
        paramCallContext.pc = 0;
        return 0;
    case 16:
        paramCallContext.proc = paramModuleMethod;
        paramCallContext.pc = 0;
        return 0;
    }
}

```

```

case 15:
    paramCallContext.proc = paramModuleMethod;
    paramCallContext.pc = 0;
    return 0;
}
paramCallContext.proc = paramModuleMethod;
paramCallContext.pc = 0;
return 0;
}

public int match1(ModuleMethod paramModuleMethod, Object paramObject,
CallContext paramCallContext)
{
    switch (paramModuleMethod.selector)
    {
        case 2:
        case 4:
        case 6:
        case 7:
        case 8:
        case 9:
        default:
            return super.match1(paramModuleMethod, paramObject, paramCallContext);
        case 11:
            if (!(paramObject instanceof mendeteksi_lokasi)) {
                return -786431;
            }
            paramCallContext.value1 = paramObject;
            paramCallContext.proc = paramModuleMethod;
            paramCallContext.pc = 1;
            return 0;
        case 10:
            paramCallContext.value1 = paramObject;
            paramCallContext.proc = paramModuleMethod;

```

```

    paramCallContext.pc = 1;
    return 0;
case 5:
    if (!(paramObject instanceof Symbol)) {
        return -786431;
    }
    paramCallContext.value1 = paramObject;
    paramCallContext.proc = paramModuleMethod;
    paramCallContext.pc = 1;
    return 0;
case 3:
    if (!(paramObject instanceof Symbol)) {
        return -786431;
    }
    paramCallContext.value1 = paramObject;
    paramCallContext.proc = paramModuleMethod;
    paramCallContext.pc = 1;
    return 0;
}
paramCallContext.value1 = paramObject;
paramCallContext.proc = paramModuleMethod;
paramCallContext.pc = 1;
return 0;
}

public int match2(ModuleMethod paramModuleMethod, Object paramObject1, Object
paramObject2, CallContext paramCallContext)
{
    switch (paramModuleMethod.selector)
    {
    case 4:
    case 5:
    case 8:
    case 10:

```

```

case 11:
case 12:
default:
    return super.match2(paramModuleMethod, paramObject1, paramObject2,
paramCallContext);
case 13:
    paramCallContext.value1 = paramObject1;
    paramCallContext.value2 = paramObject2;
    paramCallContext.proc = paramModuleMethod;
    paramCallContext.pc = 2;
    return 0;
case 9:
    paramCallContext.value1 = paramObject1;
    paramCallContext.value2 = paramObject2;
    paramCallContext.proc = paramModuleMethod;
    paramCallContext.pc = 2;
    return 0;
case 7:
    paramCallContext.value1 = paramObject1;
    paramCallContext.value2 = paramObject2;
    paramCallContext.proc = paramModuleMethod;
    paramCallContext.pc = 2;
    return 0;
case 6:
    if (!(paramObject1 instanceof Symbol)) {
        return -786431;
    }
    paramCallContext.value1 = paramObject1;
    paramCallContext.value2 = paramObject2;
    paramCallContext.proc = paramModuleMethod;
    paramCallContext.pc = 2;
    return 0;
case 3:
    if (!(paramObject1 instanceof Symbol)) {

```

```

        return -786431;
    }
    paramCallContext.value1 = paramObject1;
    paramCallContext.value2 = paramObject2;
    paramCallContext.proc = paramModuleMethod;
    paramCallContext.pc = 2;
    return 0;
}
if (!(paramObject1 instanceof Symbol)) {
    return -786431;
}
paramCallContext.value1 = paramObject1;
paramCallContext.value2 = paramObject2;
paramCallContext.proc = paramModuleMethod;
paramCallContext.pc = 2;
return 0;
}

```

```

public int match4(ModuleMethod paramModuleMethod, Object paramObject1, Object
paramObject2, Object paramObject3, Object paramObject4, CallContext
paramCallContext)
{
    switch (paramModuleMethod.selector)
    {
        default:
            return super.match4(paramModuleMethod, paramObject1, paramObject2,
paramObject3, paramObject4, paramCallContext);
        case 12:
            if (!(paramObject1 instanceof mendeteksi_lokasi)) {
                return -786431;
            }
            paramCallContext.value1 = paramObject1;
            if (!(paramObject2 instanceof Component)) {
                return -786430;
            }

```

```

    }
    paramCallContext.value2 = paramObject2;
    if (!(paramObject3 instanceof String)) {
        return -786429;
    }
    paramCallContext.value3 = paramObject3;
    if (!(paramObject4 instanceof String)) {
        return -786428;
    }
    paramCallContext.value4 = paramObject4;
    paramCallContext.proc = paramModuleMethod;
    paramCallContext.pc = 4;
    return 0;
}
paramCallContext.value1 = paramObject1;
paramCallContext.value2 = paramObject2;
paramCallContext.value3 = paramObject3;
paramCallContext.value4 = paramObject4;
paramCallContext.proc = paramModuleMethod;
paramCallContext.pc = 4;
return 0;
}
}

```



INSTITUT TEKNOLOGI NASIONAL
JL.RAYA KARANGLO KM.2
MALANG

FORMULIR PERBAIKAN SKRIPSI

Nama : **AGUNG PRANOTO**
Nim : **07.12.536**
Jurusan : **T.Elektro S-1**
Konsentrasi : **T.Komputer dan Informatika**
MasaBimbingan : **Semester Genap Tahun Akademik 2012 - 2013**
JudulSkrpsi : **APLIKASI PENCARIAN LOKASI SMARTPHONE ANDROID
BERDASARKAN TITIK KOORDINAT GPS MELALUI AUTO-
REPLY SMS**

Tanggal	Penguji	Uraian	Paraf
18 Agustus 2013	Penguji I	1. Latar belakang diperbaiki 2. Kata "posisi" diganti kata "lokasi" 3. Bab IV diperbaiki sesuai saran, dibuat tabel untuk percobaan sehingga di dapat kesimpulan	
	Penguji II	1. Tambahkan auto-start app pada saat hp mati	

Diperiksa dan Disetujui ;

Dosen Peguji I

Irmalia Suryani Faradisa, ST, MT
NIP. P . 1030400365

Dosen Penguji II

Bima Aulia Firmandani, ST

Dosen Pembimbing I

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

Dosen Pembimbing II

Yuli Wahyuni, ST, MT
NIP.P. 1031200456



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Agung Purnoto
NIM :
Perbaikan meliputi :

- Latar belakang diperbaiki
- Lokasi = posisi or semua diganti lokasi
- bab IV nya diperbaiki sesuai saran.
dibuat tabel & urut percobaannya supaya
dapat kesimpulan.

Malang



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA

NIM

Perbaikan melalui

07.12.536

tambahkan auto start App pada
beat hp mah /

Malang,

20-08-2023

U
6/12

()



FORMULIR BIMBINGAN SKRIPSI

Nim : **Agung Pranoto**
Nama : **0712536**
Masa Bimbingan : **Semester Ganjil 2012 – 2013**
Judul : **APLIKASI PENCARIAN LOKASI SMARTPHONE ANDROID
BERDASARKAN TITIK KOORDINAT GPS MELALUI
AUTO-REPLY SMS**

Tanggal	Uraian	Paraf Pembimbing

Malang, 2013
Dosen Pembimbing

M. Ibrahim Ashari, ST, MT

NIP. P.1030100358



FORMULIR BIMBINGAN SKRIPSI

Nim : **Agung Pranoto**
Nama : **0712536**
Masa Bimbingan : **Semester Ganjil 2012 – 2013**
Judul : **APLIKASI PENCAIRAN LOKASI SMARTPHONE ANDROID
BERDASARKAN TITIK KOORDINAT GPS MELALUI
AUTO-REPLY SMS**

Tanggal	Uraian	Paraf Pembimbing

Malang, 2013
Dosen Pembimbing

Yuli Wahyuni, ST, MT
NIP.P. 1031200456



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Raya Karanglo Km. 2 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Agung Pranoto
Nim : 07.12.536
Prodi : Teknik Informatika S-1
Judul : APLIKASI PENCARIAN LOKASI SMARTPHONE ANDROID BERDASARKAN
TITIK KOORDINAT GPS MELALUI AUTO-REPLY SMS
Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Selasa
Tanggal : 20 Agustus 2013
Nilai : B+

Panitia Ujian Skripsi :

Ketua Majelis Penguji

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

Anggota Penguji :

Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP. Y.1028700172

Penguji II

Bima Aulia Firmandani, ST