

**APLIKASI PEREKAMAN KECEPATAN DAN RUTE ANGKUTAN KOTA
BERBASIS SMARTPHONE ANDROID**

SKRIPSI



**Disusun Oleh :
FAZRUL RAFSANJANI
09.12.515**



**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

LEMBAR PERSETUJUAN

**APLIKASI PEREKAMAN KECEPATAN DAN RUTE ANGKUTAN KOTA
BERBASIS SMARTPHONE ANDROID**

SKRIPSI

*Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan
guna mencapai gelar Sarjana Teknik*

Disusun oleh :

FAZRUL RAFSANJANI

NIM : 09.12.515

**Mengetahui,
Ketua Program Studi Teknik Elektro S-1**

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Eng. Arjuanto S, ST, MT
NIP.P.1030800417

Irmalia Suryani Faradisa, ST, MT
NIP.P.103000365

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama : Fazrul Rafsanjani
NIM : 0912515
Program Studi : Teknik Elektro S-1
Konsentrasi : Teknik Komputer 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, 23 September 2013

Membuat Pernyataan,



Fazrul Rafsanjani
NIM :0912515

APLIKASI PEREKAMAN KECEPATAN DAN RUTE ANGKUTAN KOTA BERBASIS SMARTPHONE ANDROID

Fazrul Rafsanjani, Nim: 09.12.515

mail: fazrafsan@gmail.com

Dosen pembimbing: Dr. Eng Aryuanto Soetedjo ST, MT

Irmalia Suryani Faradisa ST, MT

Abstrak

Angkutan umum menjadi salah satu transportasi penting bagi masyarakat Flores Timur, tetapi hal ini tidak didukung dengan adanya kesadaran pengemudi angkutan umum untuk mematuhi rute serta batas kecepatan yang telah ditentukan sehingga berdampak pada rute dan jadwal angkutan yang tidak konsisten sehingga menyulitkan para penumpang dan pemilik angkutan untuk memantau rekam jejak rute angkutan yang dikelolanya. Oleh karena itu untuk mempermudah hal ini diperlukan aplikasi perekaman kecepatan dan rute kendaraan bermotor.

Maka untuk membuat aplikasi ini, penulis app eclipse serta memanfaatkan aplikasi GPS yang terdapat pada Smartphone Android sebagai alat untuk mengetahui posisi kendaraan dan untuk mengetahui kecepatan kendaraan itu sendiri. Dan jika aplikasi mencatat data rute atau kecepatan melebihi batas yang telah ditentukan, maka aplikasi akan secara otomatis akan mengirimkan data yang telah di olah menjadi pesan singkat ke nomor telepon pemilik angkutan kota.

Dari beberapa pengujian yang telah dilakukan, di dapat lah nilai error pada aplikasi perekaman kecepatan dan rute angkutan kota ini adalah 2,25 meter pada pengujian jarak, serta 0,6% nilai error pada pengujian kecepatan

KATA PENGANTAR

Puji Syukur Kehadiran Allah SWT atas berkat dan rahmat-Nya, sehingga kami selaku penyusun dapat menyelesaikan Laporan Skripsi ini yang berjudul **“APLIKASI PEREKAMAN KECEPATAN DAN RUTE ANGKUTAN KOTA BERBASIS SMARTPHONE ANDROID”** dapat terselesaikan.

Adapun maksud dan tujuan dari penulisan laporan ini merupakan salah satu syarat untuk dapat menyelesaikan studi dan mendapatkan gelar Sarjana Jurusan Teknik Elektro S-1, Konsentrasi Teknik Informatika & Komputer ITN Malang.

Sebagai pihak penyusun penulis menyadari tanpa adanya kemauan dan usaha serta bantuan dari berbagai pihak, maka laporan ini tidak dapat diselesaikan dengan baik. Oleh sebab itu, penyusun mengucapkan terima kasih kepada yang terhormat :

1. Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. M. Ibrahim Ashari, ST, MT selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang.
4. Dr. Eng. Aryoanto Suetedjo, ST, MT selaku Dosen Pembimbing satu Tugas Skripsi.
5. Ibu Irmalia Suryani Faradisa, ST, MT selaku Dosen Pembimbing dua Tugas Skripsi.
6. Sahabat-sahabat dan rekan-rekan yang tidak kami sebutkan satu-persatu, kami ucapkan banyak terima kasih atas bantuannya dalam proses pembuatan Skripsi yang telah saya kerjakan, begitu juga dengan penyelesaian laporan ini.

Usaha ini telah kami lakukan semaksimal mungkin, namun jika ada kekurangan dan kejanggalan dalam penyusunan, kami mohon saran dan kritik yang sifatnya membangun. Begitu juga sangat kami perlukan untuk menambah kesempurnaan laporan ini dan dapat bermanfaat bagi rekan-rekan mahasiswa pada khususnya dan pembaca pada umumnya.

Malang, Juli 2013

Penyusun

DAFTAR ISI

| | |
|---|------|
| LEMBAR PERSETUJUAN | i |
| ABSTRAK | ii |
| KATA PENGANTAR | iii |
| DAFTAR ISI | iv |
| DAFTAR GAMBAR | vii |
| DAFTAR TABEL | viii |
| | |
| BAB I PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah | 2 |
| 1.3. Batasan Masalah | 2 |
| 1.4. Tujuan Pembuatan Skripsi | 2 |
| 1.5. Manfaat Pembuatan Skripsi | 2 |
| 1.6. Metodologi Penelitian | 2 |
| 1.7. Sistematika Penulisan | 3 |
| | |
| BAB II LANDASAN TEORI | 4 |
| 2.1. Angkutan Umum | 4 |
| 2.1.1 Pengertian Angkutan Umum..... | 4 |
| 2.2.2 Pemanfaatan Angkutan Umum..... | 4 |
| 2.2.3 Jenis Angkutan Darat Di Indonesia | 4 |
| 2.2. Perkembangan Android | 6 |
| 2.3 Software Development Kit (SDK) Eclipse..... | 9 |
| 2.3.1 Sejarah | 10 |
| 2.3.2 Arsitektur | 10 |
| 2.3.3 Histori (versi-versi) Eclipse | 11 |
| 2.4 GPS | 12 |
| 2.4.1 Sejarah Perkembangan GPS..... | 12 |

| | |
|---|-----------|
| 2.4.2 Spesifikasi GPS..... | 13 |
| 2.4.3 Implementasi..... | 16 |
| 2.4.4 Aplikasi..... | 16 |
| 2.5 SMS..... | 17 |
| | |
| BAB III DESAIN DAN IMPLEMENTASI SISTEM..... | 20 |
| 3.1.Deskripsi Sistem..... | 20 |
| 3.2 Analisa sistem..... | 20 |
| 3.3 Tahap Perancangan..... | 21 |
| 3.3.1 Rancangan Use Case Diagram..... | 22 |
| 3.3.2 Rancangan Diagram Alir..... | 21 |
| 3.3.3 Rancangan tampilan Awal..... | 23 |
| 3.3.4 Rancangan Tampilan Menu Input Nomor Tujuan..... | 24 |
| 3.3.5 Rancangan Tampilan Hasil Perekaman Rute Dan Kecepatan | 25 |
| | |
| BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM..... | 26 |
| 4.1. Implementasi Aplikasi Perekaman Kecepatan Dan Rute Angkutan Kota..... | 26 |
| 4.2. Pengujian Aplikasi Perekaman Kecepatan Dan Rute Angkutan Kota..... | 26 |
| 4.3. Tampilan Aplikasi Perekaman Kecepatan Dan Rute Angkutan Kota..... | 26 |
| 4.3.1 Icon Aplikasi Pada Ponsel..... | 26 |
| 4.3.2 Tampilan Menu Input Plat Nomor..... | 28 |
| 4.3.3 Tampilan Menu Input Nomor Tujuan..... | 29 |
| 4.3.4 Tampilan Button Start GPS..... | 30 |
| 4.3.5 Tampilan Hasil Perekaman Pertama..... | 31 |
| 4.3.6 Tampilan Hasil Perekaman Kedua..... | 32 |
| 4.3.7 Tampilan Hasil Perekaman Ketiga..... | 33 |
| 4.3.8 Tampilan Hasil Perekaman Keempat..... | 34 |

| | |
|--|----|
| 4.3.9 Tampilan Hasil Perekaman Kelima | 35 |
| 4.3.10 Tampilan Hasil Perekaman Keenam..... | 36 |
| 4.3.11 Tampilan Button Stop GPS..... | 37 |
| 4.3.12 Tampilan Hasil Pengiriman Data Pada Nomor Telepon Tujuan | 38 |
| 4.4 Tampilan Rute Angkutan Kota Flores Timur Menggunakan Google Earth..... | 39 |
| 4.5. Hasil Pengujian Keakurasian Posisi Dan Kecepatan | 42 |
| 4.5.1. Hasil Uji Coba Keakurasian Posisi Menggunakan 3 Aplikasi GPS Yang Berbeda..... | 42 |
| 4.5.2 Hasil Uji Coba Keakurasian Kecepatan Menggunakan Aplikasi GPS Angkutan Kota | 45 |
| | |
| BAB V PENUTUP | 50 |
| 5.1. Kesimpulan | 50 |
| 5.2 Saran | 50 |
| | |
| DAFTAR PUSTAKA | 51 |

DAFTAR GAMBAR

| | |
|--|----|
| 3.1. Use Case Diagram | 21 |
| 3.2. Diagram Alir Kecepatan | 22 |
| 3.3. Diagram Alir Rute..... | 23 |
| 3.4. Rancangan Tampilan Awal..... | 24 |
| 3.5 Rancangan Input Nomor Tujuan..... | 24 |
| 3.6 Rancangan Hasil | 25 |
| 4.1. Tampilan Icon Aplikasi..... | 27 |
| 4.2. Tampilan Menu Input Plat Nomor..... | 28 |
| 4.3. Tampilan Menu Nomor Telepon Tujuan..... | 29 |
| 4.4 Tampilan Start GPS..... | 30 |
| 4.5 Tampilan Hasil Perekaman Pertama..... | 31 |
| 4.6 Tampilan Hasil Perekaman ke Dua..... | 32 |
| 4.7 Tampilan Hasil Perekaman ke Tiga..... | 33 |
| 4.8 Tampilan Hasil Perekaman ke Empat..... | 34 |
| 4.9 Tampilan Hasil Perekaman ke Lima..... | 35 |
| 4.9 Tampilan Hasil Perekaman ke Enam..... | 36 |
| 4.10 Tampilan Stop GPS..... | 37 |
| 4.11 Tampilan Laporan Start GPS Dan Penerimaan Data..... | 38 |
| 4.12 Tampilan Penerimaan Data Dan Laporan Stop GPS..... | 38 |
| 4.13 Tampilan Rute Angkutan Kota Flores Timur Menggunakan Google Earth Dari Landmark 1 sampai 17..... | 39 |
| 4.14 Tampilan Rute Angkutan Kota Flores Timur Menggunakan Google Earth Dari Landmark 17 sampai 40..... | 40 |
| 4.15 Hasil Percobaan Pertama Menggunakan 3 Aplikasi GPS Yang Berbeda | 43 |
| 4.16 Hasil Percobaan Kedua Menggunakan 3 Aplikasi GPS Yang Berbeda | 44 |
| 4.17 Hasil Uji Coba Keakurasian Kecepatan Pada Aplikasi | |

| | |
|--|----|
| GPS Angkutan Kota | 46 |
| 4.18 Tampilan Speedometer Pada Posisi Diam | 47 |
| 4.19 Tampilan Speedometer Pada Posisi Bergerak | 48 |
| 4.20 Hasil Konversi Kecepatan | 49 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 4.1. Hasil Pemetaan Dan Perhitungan Jarak Menggunakan Google Earth | 40 |
| Tabel 4.2 Hasil Percobaan Pertama Keakurasian Posisi..... | 43 |
| Tabel 4.3 Hasil Percobaan Kedua Keakurasian Posisi..... | 45 |

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Angkutan umum menjadi salah satu transportasi penting bagi masyarakat Flores Timur, tetapi hal ini tidak didukung dengan adanya kesadaran pengemudi angkutan umum untuk mematuhi rute serta batas kecepatan yang telah ditentukan sehingga berdampak pada rute dan jadwal angkutan yang tidak konsisten sehingga menyulitkan para penumpang dan pemilik angkutan untuk memantau rekam jejak rute angkutan yang dikelolanya. Oleh karena itu untuk mempermudah hal ini diperlukan aplikasi perekaman kecepatan dan rute kendaraan bermotor.

Sistem perekaman ini adalah suatu aplikasi yang dapat memudahkan pemilik kendaraan untuk mengetahui rute yang telah dilalui dan kecepatan kendaraan tersebut secara real-time melalui rekaman pada smartphone android.

Pada Skripsi ini penulis membuat suatu aplikasi mobile yang berfungsi untuk merekam rute dan kecepatan angkutan kota menggunakan sistem operasi Android. Aplikasi ini dapat digunakan oleh pemilik kendaraan bermotor yang menggunakan telepon seluler dengan sistem operasi Android. Dan dalam aplikasi ini dilengkapi pula dengan sistem SMS gateway, dimana pemilik kendaraan dapat menerima laporan melalui SMS yang dikirim langsung oleh smartphone Android yang ada di kendaraan tersebut.

Hal ini tentunya menjadi keunggulan sendiri karena pengguna tidak perlu mengawasi kendarannya di jalan raya. Pengguna tinggal menggunakan aplikasi ini untuk mengetahui informasi kendaraanya tersebut. Berdasarkan latar belakang tersebut maka penulis membuat skripsi dengan judul "APLIKASI PEREKAMAN KECEPATAN DAN RUTE ANGKUTAN KOTA BERBASIS SMARTPHONE ANDROID".

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka masalah yang dapat dibahas adalah mengenai :
Bagaimana cara membuat aplikasi perekaman kecepatan dan rute angkutan kota berbasis smartphone android?

1.3 Batasan Masalah

Permasalahan yang akan dibahas dalam penulisan ini dibatasi pada:

1. Program aplikasi ini hanya berfungsi pada Sistem Operasi Android minimal 2,3.
2. Pengembangan menggunakan Apps eclipse.
3. Rute angkutan yang dibahas adalah rute angkutan kota Kabupaten Flores Timur.
4. Kecepatan maksimal angkutan kota adalah 60 km/jam.
5. Toleransi untuk jarak dari rute adalah 500 meter.

1.4 Tujuan Pembuatan Skripsi

Tujuan pembuatan program ini adalah untuk membuat aplikasi perekaman kecepatan dan rute angkutan kota yang akan merekam informasi jalur yang ditempuh dan kecepatan kendaraan tersebut.

1.5 Manfaat Pembuatan Skripsi

Manfaat dari pembuatan skripsi ini dengan judul “apliaksi perekaman rute dan kecepatan angkutan kota berbasis android” adalah sebagai berikut:

1. Dapat membantu mengurangi tingkat kecelakaan lalu lintas
2. Dapat meminimalisir pengeluaran untuk perawatan angkutan kota

1.6 Metodologi Penelitian

Adapun metode penelitian yang digunakan 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 dihasilkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan aplikasi dimana nantinya akan digunakan sebagai acuan perancangan aplikasi.

3. Perancangan dan implementasi

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 dan diimplementasikan ke dalam aplikasi.

4. Eksperimen dan evaluasi

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

1.7 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut:

Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah, Metode Penelitian, dan Sistematika Penulisan.

Bab II : Tinjauan Pustaka

Berisi tentang Landasan Teori mengenai permasalahan yang berhubungan dengan perancangan aplikasi yang dilakukan.

Bab III : Perancangan dan Analisa Aplikasi

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

Bab IV: Pembuatan dan Pengujian Sistem

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 pengembangan penulisan selanjutnya.

BAB II

LANDASAN TEORI

2.1. Angkutan Umum

2.1.1. Pengertian Angkutan Umum

Angkutan umum adalah salah satu media transportasi yang digunakan masyarakat secara bersama sama dengan membayar tarif. Angkutan umum merupakan lawan kata dari kendaraan pribadi. Sejalan dengan peningkatan pendapatan masyarakat, kini banyak masyarakat yang mampu membeli kendaraan pribadi. Banyak alasan untuk memiliki kendaraan pribadi, antara lain demi kenyamanan dan privasi, namun kepemilikan kendaraan pribadi yang tertalu banyak juga akan menimbulkan masalah baru.

2.1.2. Pemanfaatan Angkutan Umum

Berbicara tentang angkutan umum, sebenarnya bukanlah hal baru. Dari zaman dahulu manusiapun telah mengenal angkutan umum. Dari yang berupa angkutan umum dengan menggunakan kereta kuda, hingga sekarang zaman modern yang menggunakan pesawat terbang. Dari semua jenis angkutan umum yang digunakan, sejatinya hanya satu tujuan utamanya, yaitu memanfaatkan waktu secara efisien.

2.1.3. Jenis – Jenis Angkutan Darat di Indonesia

a. Bemo

Ternyata Bemo punya singkatan yaitu “becak motor” dan merupakan kendaraan bermotor roda tiga yang biasanya digunakan sebagai angkutan umum di Indonesia. Meskipun nama Bemo dikaitkan dengan becak, namun bemo sebenarnya adalah model bis ukuran supermini, karena mempunyai trayek dan tarif tetap. Beda dengan becak yang lebih mirip taksi bertenaga nasi. Memang bemo mulanya

beroperasi seperti taksi, namun kemudian dibentuk trayek tertentu, dan akhirnya dikhususkan ke trayek pinggiran yang tidak disentuh oleh bus kota.

Bemo mulai dipergunakan di Indonesia pada awal tahun 1962, pertama-tama di Jakarta, kemudian Bogor, Bandung, Surabaya, Surakarta, Malang, Padang, Denpasar, yang dimaksudkan untuk menggantikan becak. Namun rencana ini tidak berhasil karena kehadiran bemo tidak didukung oleh rencana yang matang. Bemo cepat populer karena kendaraan ini lebih murah, berdaya angkut besar (bisa menampung 7 orang), mampu menjangkau jalan-jalan yang sempit, bisa menempuh jarak jauh dan lebih cepat daripada becak yang lebih mahal karena daya angkutnya cuma 2 orang, pelan dan tidak bisa jauh. Di Jakarta, bemo mulai dipinggirkan pada 1971, disusul oleh Surabaya dan Malang pada tahun yang sama. Pada 1979, Pemerintah Daerah Surakarta mengambil langkah yang sama.

b. Helicak

Helicak juga mempunyai singkatan, dan lagi-lagi berhubungan dengan becak, yaitu helikopter becak, karena bentuknya mirip dengan helikopter dan becak. Helicak adalah kendaraan angkutan masyarakat seperti taxi yang banyak ditemukan di Jakarta, Surabaya, dan Yogyakarta pada tahun 1970-an.

Helicak pertama muncul tahun 1971, Diadopsi pada skuter Lambretta yang didatangkan dari Italia. Gubernur Jakarta waktu itu, Ali Sadikin merencanakannya sebagai pengganti becak yang dianggap tidak manusiawi.

c. Bajaj

Bajaj awalnya sama dengan helicak, untuk menggantikan becak. Jika helicak produk Italia, bajaj adalah produk India dibawah lisensi Italia. Desain bajaj masih lebih manusiawi dari pada helicak, dengan nilai ekonomis yang lebih baik. Menjadikan bajaj cepat populer. Bajaj masih eksis sampai kini dan upaya membuat punah angkutan ini susah. Sehingga kemudian lebih mudah mengalihkannya ke generasi bajaj baru type RE 4 stroke yang menggunakan mesin 4 tak dengan sistem 2 bahan bakar, premium dan gas, berwarna biru, berbeda dengan bajaj legendaris 2 tak yang berwarna oranye.

d. Becak

Becak (dari bahasa hokkien: be chia = kereta kuda) adalah suatu model transportasi beroda tiga yang umum ditemukan di indonesia dan juga disebagian asia. Kapasitas normal becak adalah dua orang penumpang dan seorang pengemudi. Tidak jelas kapan becak mulai dikenal di indonesia, namun diperkirakan legenda becak di indonesia dimulai sekitar tahun 1930-an.

e. Delman

Delman adalah kendaraan transportasi tradisional yang beroda dua atau empat yang menggunakan kuda sebagai penggantinya. Nama kendaraan ini berasal dari nama penemunya, yaitu charles theodore deeleman. Seorang litografer dan insinyur di masa hindia belanda. Orang belanda sendiri menyebut kendaraan ini dengan nama dos-a-dos (punggung pada punggung, arti harafiah bahasa prancis), yaitu sejenis kereta yang posisi duduk penumpangnya saling menunggangi.

2.2. Perkembangan Android

Android adalah salah satu sistem operasi untuk telepon seluler yang berbasis Linux. Pada awalnya dikembangkan oleh Android Inc. (sebuah perusahaan telepon seluler yang berada di California, AS), pada bulan Juli 2000. Rumor yang berkembang adalah bahwa Google akan memasuki pasar telepon seluler, dikarenakan kerjasama yang dilakukan antara Google dengan Android inc.

Sebuah tim yang memiliki tugas untuk mengembangkan program perangkat seluler yang didukung oleh kernel Linux akhirnya dibentuk, hal ini menjadikan rumor yang beredar semakin kuat bahwa memang Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

Untuk mempermudah usaha Google untuk mengembangkan produk Android, maka Google membeli Android Inc dan serta dibentuklah “Open Handset Alliance” yang tergabung atas 34 perusahaan software, hardware dan telekomunikasi seperti : Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Nvidia dan lainnya.

Sekitar September 2007, Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar

yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat mobile yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru. Telepon seluler pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008.

Berikut adalah sejarah dan fitur sistem operasi Android semua versi:

1. Android Versi 1.0 (Apple pie)

Android 1.0 dirilis pada tanggal 23 September 2008 yang memiliki kode nama Apple pie serta ukuran layar 320×480 HVGA.

2. Android Versi 1.1 (Banana bread)

Android 1.1 dirilis pada tanggal 9 Februari 2009 yang memiliki kode nama Banana bread serta ukuran layar 320×480 HVA.

3. Android 1.5 (Cupcake)

Android 1.5 dirilis pada tanggal 30 April 2009 yang memiliki kode nama Cupcake. Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur yakni kemampuan merekam dan menonton video, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon seluler, dukungan Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

4. Android 1.6 (Donut)

Android 1.6 dirilis pada tanggal 15 September 2009 yang memiliki kode nama Donut. Versi ini perbaikan dari Android Cupcake dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk

memilih foto yang akan dihapus; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel); pengadaan resolusi VWGA.

5. Android versi 2.0/2.1 (Eclair)

Android versi 2.0/2.1 dirilis pada tanggal 3 Desember 2009 yang memiliki kode nama Eclair. perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

6. Android versi 2.2 (Froyo)

Android versi 2.2 dirilis pada tanggal 20 Mei 2010 yang memiliki kode nama Froyo. Froyo akan memperkenalkan peningkatan kecepatan dengan JIT optimalisasi dan Chrome V8 mesin JavaScript, dan menambahkan Wi-Fi hotspot dan dukungan Adobe Flash.

7. Android versi 2.3 (Gingerbread)

Android versi 2.3 dirilis pada tanggal 6 Desember 2010 yang memiliki kode nama Gingerbread. Android Gingerbread adalah android yang mempermudah pengguna antarmuka , meningkatkan keyboard lunak dan fitur copy / paste, dan menambahkan dukungan untuk Near Field Communication. pada versi ini Android telah melakukan update sebanyak 5 kali yang dirilis pada tanggal yang berbeda-beda serta penambahan difitur-fitur baru, mereka masing-masing adalah Android 2.3.3, Android 2.3.4, Android, 2.3.5, Android 2.3.6, dan terakhir Android 2.3.7

8. Android versi 3.0 (Honeycomb)

Android versi 3.0 dirilis pada tanggal 15 Juli 2011 yang memiliki kode nama Honeycomb. Di versi ini Android akhirnya jalan pada sebuah Tablet, dan tablet pertama yang menggunakan sistem operasi Honeycomb adalah Motorola Xoom.

9. Android versi 4.0 (Ice Cream Sandwich)

Android versi 4.0 dirilis pada tanggal 19 Oktober 2011 yang memiliki kode nama Ice Cream Sandwich. Di versi Android menyempurnakan seluruh GUI yang dibangun dan penambahan fitur seperti Facial recognition (Face Unlock), UI use Hardware acceleration, Better voice recognition (dictating/Voice typing), Web

browser, allows up to 16 tabs, Updated launcher (customizable), Android Beam app to exchange data through NFC, Resizeable widgets

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.

Android memiliki berbagai keunggulan sebagai software yang memakai basis kode komputer yang bisa didistribusikan secara terbuka (open source) sehingga pengguna bisa membuat aplikasi baru di dalamnya. Android memiliki aplikasi native Google yang terintegrasi seperti pushmail Gmail, Google Maps, dan Google Calendar.

Para penggemar open source kemudian membangun komunitas yang membangun dan berbagi Android berbasis firmware dengan sejumlah penyesuaian dan fitur-fitur tambahan, seperti FLAC lossless audio dan kemampuan untuk menyimpan download aplikasi pada microSD card. Mereka sering memperbaharui paket-paket firmware dan menggabungkan elemen-elemen fungsi Android yang belum resmi diluncurkan dalam suatu carrier-sanction firmware.

Berikut sejumlah vendor yang mengeluarkan telepon seluler berbasis Android :

1. HTC
2. Samsung
3. LG
4. Motorola
5. Sony
6. Acer
7. Dell

2.3. Software Development Kit (SDK) Eclipse

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. Multit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain 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, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan open source. Open source 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 membuat komponen yang disebut plug-in.

2.3.1. Sejarah

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak pengembangan IBM Visual Age for Java 4.0. Produk Eclipse ini diluncurkan oleh IBM pada tanggal 5 November 2001. IBM menginvestasikan US\$ 40 juta untuk pengembangannya. Sejak 5 November 2001, konsorsium Eclipse Foundation mengambil alih pengembangan Eclipse lebih lanjut.

2.3.2. Arsitektur

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah dipasang (diinstal). Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP).

Berikut ini adalah komponen yang membentuk RCP:

- a. Core platform
 - b. OSGi
 - c. SWT (Standard Widget Toolkit)
 - d. JFace
 - e. Eclipse Workbench
-

Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE adalah:

- a) terbuka (open),
- b) mudah diperluas (extensible) untuk apa saja, dan
- c) tidak untuk sesuatu yang spesifik.

Eclipse tidak saja untuk mengembangkan program Java, tetapi juga untuk berbagai macam keperluan. Perluasan apapun cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ maka telah terdapat plug-in CDT (C/C++ Development Tools) yang dapat dipasang di Eclipse untuk Eclipse menjadi perangkat untuk pengembangan C/C++.

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. Salah satu situs yang menawarkan plug-in yang gratis adalah Eclipse downloads by project.

2.3.3. Histori (versi-versi) Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah deployment dan maintenance untuk sistem enterprise, serta untuk kenyamanan. Peluncuran simultan dijadwalkan pada bulan Juni setiap tahunnya

2.4. GPS

GPS adalah sistem radio navigasi dan penentuan posisi menggunakan satelit yang dimiliki dan dikelola oleh Amerika Serikat. Sistem ini dapat digunakan oleh banyak orang sekaligus dalam segala cuaca, serta didesain untuk memberikan posisi dan kecepatan tiga dimensi yang teliti dan juga informasi mengenai waktu secara kontinyu di seluruh dunia. GPS telah banyak digunakan di Indonesia, antara lain untuk eksplorasi minyak, pertambangan, geologi, kelautan, dan dapat diintegrasikan dengan SIG misalnya untuk tracking benda bergerak (mobil, pesawat, satelit, dll). Secara komersial alat ini selain dapat membantu pengguna dalam menentukan lokasinya di permukaan bumi, juga dapat merekomendasikan lintasan dari lokasi saat ini hingga tujuan perjalanan, merekam lintasan yang pernah dilalui dan memberikan informasi lokasi fasilitas-fasilitas penting terdekat seperti ATM, Bank, supermarket dan lain-lain.

2.4.1. Sejarah Perkembangan GPS

Sejarah GPS dimulai dari awal tahun 1960-an saat Departemen Pertahanan (Dephan) Amerika Serikat merasa perlu memiliki sistem navigasi yang akurat, dapat berfungsi secara global, dalam segala cuaca, dan tersedia setiap saat. Berbagai pendekatan dan teknologi diuji coba sampai akhirnya pada akhir tahun 1973 Dephan AS menyetujui pelaksanaan uji coba satelit Navstar yang menjadi generasi pertama dari satelit GPS. Hingga tahun 1983, masa pemerintahan Presiden Ronald Reagan mengizinkan penggunaan GPS untuk pesawat sipil setelah terjadi insiden penembakan pesawat Korean Airlines, penerbangan 007 yang dianggap “nyasar” melintasi perbatasan Uni Soviet. Sejak saat itu, GPS mulai disiapkan untuk dipergunakan oleh kalangan sipil secara internasional, terutama untuk kalangan penerbangan dan kelautan.

Lonjakan pesat industri GPS pertama terjadi di tahun 1991 saat terjadinya Perang Teluk. Pada saat itu, Pentagon memesan 10.000 unit dan 3.000 unit perangkat GPS nonmilitar dari Trimble Navigation dan Magellan Systems. Pada perkembangan

selanjutnya, perangkat GPS terus dikembangkan semakin baik, andal, dan terjangkau harganya.

2.4.2 Spesifikasi GPS

a. Sinyal GPS

Satelit GPS mengirim dua sinyal transmisi gelombang radio dengan emisi "Code-Phase" dan "Carrier-Phase" untuk menghitung jarak Satellite dan GPS Receiver agar lebih akurat, dengan frekuensi L1(1,57542 GHz) GPS transmisi Signal diperuntukan pengguna sipil dan L2.(1227.60 MHz) US GPS transmisi Sinyal untuk keperluan militer dengan spesifikasi keakuratan serta Error Correction lebih baik. Sinyal satelite GPS Navstar memancar menyorot permukaan bumi sesuai dengan karakter signal Microwave pada band sekitar 1.2- 1,5 GHZ, menembus awan, kaca dan plastic namun tidak akan bisa menembus benda padat/keras seperti bangunan atau gunung.

b. Kemampuan GPS

Beberapa kemampuan GPS antara lain dapat memberikan informasi tentang posisi, kecepatan, dan waktu secara cepat, akurat, murah, dimana saja di bumi ini tanpa tergantung cuaca. Hal yang perlu dicatat bahwa GPS adalah satu-satunya sistem navigasi ataupun sistem penentuan posisi dalam beberapa abad ini yang memiliki kemampuan handal seperti itu. Ketelitian dari GPS dapat mencapai beberapa mm untuk ketelitian posisinya, beberapa cm/s untuk ketelitian kecepatannya dan beberapa nanodetik untuk ketelitian waktunya. Ketelitian posisi yang diperoleh akan tergantung pada beberapa faktor yaitu metode penentuan posisi, geometri satelit, tingkat ketelitian data, dan metode pengolahan datanya.

c. Produk yang diberikan GPS

Secara umum produk dari GPS adalah posisi, kecepatan, dan waktu. Selain itu ada beberapa produk lainnya seperti percepatan, azimuth, parameter attitude, TEC (Total Electron Content), WVC (Water Vapour Content), Polar motion parameters, serta beberapa produk yang perlu dikombinasikan dengan informasi eksternal dari

sistem lain, produknya antara lain tinggi ortometrik, undulasi geoid, dan defleksi vertikal.

d. Segmen Penyusun Sistem GPS

Secara umum ada tiga segmen dalam sistem GPS yaitu segmen sistem kontrol, segmen satelit, dan segmen pengguna. Satelit GPS dapat dianalogikan sebagai stasiun radio angkasa, yang dilengkapi dengan antena-antena untuk mengirim dan menerima sinyal –sinyal gelombang. Sinyal-sinyal ini selanjutnya diterima oleh receiver GPS di/dekat permukaan bumi, dan digunakan untuk menentukan informasi posisi, kecepatan, maupun waktu. Selain itu satelit GPS juga dilengkapi dengan peralatan untuk mengontrol attitude satelit. Satelit-satelit GPS dapat dibagi atas beberapa generasi yaitu ; blok I, blok II, blok IIA, blok IIR dan blok IIF. Hingga april 1999 ada 8 satelit blok II, 18 satelit blok II A dan 1 satelit blok II R yang operasional.

Secara umum segmen sistem kontrol berfungsi mengontrol dan memantau operasional satelit dan memastikan bahwa satelit berfungsi sebagaimana mestinya. Segmen pengguna terdiri dari para pengguna satelit GPS di manapun berada. Dalam hal ini alat penerima sinyal GPS (GPS receiver) diperlukan untuk menerima dan memproses sinyal -sinyal dari satelit GPS untuk digunakan dalam penentuan posisi, kecepatan dan waktu. Komponen utama dari suatu receiver GPS secara umum adalah antena dengan pre-amplifier, bagian RF dengan pengidentifikasi sinyal dan pemroses sinyal, pemroses mikro untuk pengontrolan receiver, data sampling dan pemroses data (solusi navigasi), osilator presisi , catu daya, unit perintah dan tampilan, dan memori serta perekam data.

e. Tipe alat (Receiver) GPS

Ada 3 macam tipe alat GPS, dengan masing-masing memberikan tingkat ketelitian (posisi) yang berbeda-beda. Tipe alat GPS pertama adalah tipe Navigasi (Handheld, Handy GPS). Tipe navigasi harganya cukup murah, sekitar 1 – 4 juta rupiah, namun ketelitian posisi yang diberikan saat ini baru dapat mencapai 3 sampai 6 meter. Tipe alat yang kedua adalah tipe geodetik single frekuensi (tipe pemetaan), yang biasa digunakan dalam survey dan pemetaan yang membutuhkan ketelitian posisi sekitar sentimeter sampai dengan beberapa desimeter. Tipe terakhir adalah tipe

Geodetik dual frekuensi yang dapat memberikan ketelitian posisi hingga mencapai milimeter. Tipe ini biasa digunakan untuk aplikasi precise positioning seperti pembangunan jaring titik kontrol, survey deformasi, dan geodinamika. Harga receiver tipe geodetik cukup mahal, mencapai ratusan juta rupiah untuk 1 unitnya.

f. Sinyal dan Bias pada GPS

GPS memancarkan dua sinyal yaitu frekuensi L1 (1575.42 MHz) dan L2 (1227.60 MHz). Sinyal L1 dimodulasikan dengan dua sinyal pseudo-random yaitu kode P (Protected) dan kode C/A (coarse/aquisition). Sinyal L2 hanya membawa kode P. Setiap satelit mentransmisikan kode yang unik sehingga penerima (receiver GPS) dapat mengidentifikasi sinyal dari setiap satelit. Pada saat fitur "Anti-Spoofing" diaktifkan, maka kode P akan dienkripsi dan selanjutnya dikenal sebagai kode P(Y) atau kode Y.

Ketika sinyal melalui lapisan atmosfer, maka sinyal tersebut akan terganggu oleh konten dari atmosfer tersebut. Besarnya gangguan di sebut bias. Bias sinyal yang ada utamanya terdiri dari 2 macam yaitu bias ionosfer dan bias troposfer. Bias ini harus diperhitungkan (dimodelkan atau diestimasi atau melakukan teknik differencing untuk metode diferensial dengan jarak baseline yang tidak terlalu panjang) untuk mendapatkan solusi akhir koordinat dengan ketelitian yang baik. Apabila bias diabaikan maka dapat memberikan kesalahan posisi sampai dengan orde meter.

g. Error Source pada GPS

Pada sistem GPS terdapat beberapa kesalahan komponen sistem yang akan mempengaruhi ketelitian hasil posisi yang diperoleh. Kesalahan-kesalahan tersebut contohnya kesalahan orbit satelit, kesalahan jam satelit, kesalahan jam receiver, kesalahan pusat fase antena, dan multipath. Hal-hal lainnya juga ada yang mengiringi kesalahan sistem seperti efek imaging, dan noise. Kesalahan ini dapat dieliminir salah satunya dengan menggunakan teknik differencing data.

h. Ketelitian Posisi yang diperoleh dari Sistem GPS

Untuk aplikasi sipil, GPS memberikan nilai ketelitian posisi dalam spektrum yang cukup luas, mulai dari meter sampai dengan milimeter. Sebelum Mei 2000 (SA on) ketelitian posisi GPS metode absolut dengan data pseudorange mencapai 30 – 100 meter. Kemudian setelah SA off ketelitian membaik menjadi 3 – 6 meter. Sementara itu Teknik DGPS memberikan ketelitian 1-2 meter, dan teknik RTK memberikan ketelitian 1-5 sentimeter. Untuk posisi dengan ketelitian milimeter diberikan oleh teknik survei GPS dengan peralatan GPS tipe geodetik dual frekuensi dan strategi pengolahan data tertentu.

2.4.3. Implementasi

1. Menghitung jarak dan arah dari lokasi tempat kita berada.
2. Satu unit GPS dapat menyimpan dalam memory lokasi di mana kita berada saat ini.
3. Setiap lokasi dapat diberi nama atau nomor dan tanggal dan waktu.
4. Mengingat lokasi yang pernah kita simpan.
5. Mengarahkan kita dari satu lokasi ke lokasi lain dengan simbol berupa grafik.
6. Menyimpan rute perjalanan kita dan mengantar kita kembali dengan rute yang sama.
7. Berfungsi sebagai kompas yang dapat menuntun kita ke arah yang tepat.
8. Dapat digunakan sebagai penunjuk arah di kapal dan di mobil.
9. Beberapa GPS dapat menunjukkan peta jalan-jalan utama, sungai-sungai.
10. Beberapa GPS juga dapat menampilkan kekuatan baterai, posisi satelit, kekuatan sinyal.

2.4.4. Aplikasi

Aplikasi GPS sangat beragam dan tidak terbatas pada hal-hal yang berhubungan dengan penentuan posisi saja. Di udara, GPS digunakan sebagai salah satu alternatif peralatan navigasi pesawat terbang. Dibandingkan dengan peralatan navigasi lain, penerima GPS paling mudah digunakan karena langsung memberikan

posisi pesawat sehingga sangat cepat menjadi populer. Dengan menggunakan beberapa penerima GPS, orientasi kemiringan pesawat juga bisa dihitung, GPS juga favorit digunakan untuk membimbing pesawat tanpa awak dan rudal-rudal jarak jauh.

Di laut, kapal-kapal juga senang menggunakan GPS karena alasan kemudahan penggunaannya. IMO (International Maritime Organization) bahkan menganjurkan pemakaian AIS (Automatic Identification System), yaitu alat penerima GPS yang secara periodik mengirimkan posisi kapal. GPS juga digunakan untuk mempelajari kebiasaan migrasi satwa laut.

Penerima GPS yang tersedia dalam berbagai bentuk dan ukuran membuat penggunaannya di darat juga beragam. Mulai dari penerima GPS handheld untuk perjalanan lintas alam seharga sekitar Rp 1 juta sampai penerima GPS untuk memantau perjalanan truk-truk kontainer dan kereta api. GPS juga digunakan membuat peta dan membantu bermain golf. Jam satelit GPS yang sangat presisi juga banyak dimanfaatkan, di antaranya sinkronisasi antar BTS/menara pada jaringan telepon seluler.

Beberapa tahun belakangan GPS bahkan dimanfaatkan juga di angkasa luar untuk mendapatkan posisi satelit lainnya. Akan tetapi, aplikasi yang paling kreatif menurut penulis adalah menggunakan GPS sebagai radar.

2.5. SMS

Pengertian SMS. Short Message Service (SMS) adalah suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat nirkabel, yaitu perangkat komunikasi telepon selular, dalam hal ini perangkat nirkabel yang digunakan adalah telepon selular. Salah satu kelebihan dari SMS adalah biaya yang murah.

Selain itu SMS merupakan metode store dan forward sehingga keuntungan yang didapat adalah pada saat telepon selular penerima tidak dapat dijangkau, dalam arti tidak aktif atau diluar service area, penerima tetap dapat menerima SMS-nya apabila telepon selular tersebut sudah aktif kembali. SMS menyediakan mekanisme untuk mengirimkan pesan singkat dari dan menuju media-media wireless dengan

menggunakan sebuah Short Messaging Service Center (SMSC), yang bertindak sebagai sistem yang berfungsi menyimpan dan mengirimkan kembali pesan-pesan singkat.

Jaringan wireless menyediakan mekanisme untuk menemukan station yang dituju dan mengirimkan pesan singkat antara SMSC dengan wireless station. SMS mendukung banyak mekanisme input sehingga memungkinkan adanya interkoneksi dengan berbagai sumber dan tujuan pengiriman pesan yang berbeda.

Pengertian SMS Menurut beberapa ahli, Short Message Service (SMS) adalah sebuah layanan yang dilaksanakan dengan sebuah telepon Selular untuk mengirim atau menerima pesan-pesan pendek. Pada mulanya SMS dirancang sebagai bagian daripada GSM, tetapi sekarang sudah didapatkan pada jaringan bergerak lainnya termasuk jaringan Universal Mobile Telecommunications System (UMTS).

Sebuah pesan SMS maksimal terdiri dari 140 bytes, dengan kata lain sebuah pesan bisa memuat 140 karakter 8-bit, 160 karakter 7-bit atau 70 karakter 16-bit untuk Bahasa Jepang, Bahasa Korea dan Bahasa Mandarin yang memakai Hanzi (Aksara Kanji/Hanja). Selain 140 bytes ini ada data-data lain yang termasuk. Adapula beberapa metode untuk mengirim pesan yang lebih dari 140 bytes, tetapi seorang pengguna harus membayar lebih dari sekali. Misalnya pesan yang dikirimkan terdiri dari 167 karakter, maka pesan ini akan dipecah menjadi 2 buah SMS (1 buah SMS dengan 160 karakter dan 1 SMS dengan 7 karakter).

Kedua SMS ini akan dikirimkan sebagai 2 SMS terpisah dan di sisi penerima akan digabungkan menjadi satu SMS lagi. Elemen yang dapat mengirim maupun menerima pesan pendek dinamakan External Short Message Entities (ESME). ESME dapat berupa aplikasi software di dalam mobile handset, faksimili, remote internet server, dan lain-lain. ESME juga dapat berupa server yang menghubungkan Short Message Service Center (SMSC) secara langsung atau via gateway. Dengan teknologi GSM/GPRS, operator jaringan telepon dapat dengan mudahnya melakukan pertukaran pesan dari jaringan yang berbeda. Pemetaan sinyal dilakukan diantara dua jaringan telepon. dalam pemetaan dua jaringan ini, SMSC dari pembuat ESME mengolah Home Location Register (HLR).

Jaringan ini bertujuan untuk mendapatkan informasi mengenai penerima dan mengirimnya langsung. Dalam contoh ini, SMSC penerima pesan tidak berpengaruh. Proses pengiriman SMS antar sesama teknologi jaringan Untuk pengiriman pesan diantara teknologi jaringan yang berbeda (seperti GSM/GPRS dan CDMA), dilakukan dengan menyambungkan dua gateway jaringan telepon dengan menggunakan protokol pertukaran.

Proses pengiriman SMS antar teknologi jaringan yang berbeda Dalam pengiriman antara dua teknologi jaringan yang berbeda terdapat beberapa tahap. Pertama, pesan di buat dan dikirimkan oleh ESME ke SMSC pengirim. Selanjutnya SMSC pengirim meneruskan pesan melalui SMSC penerima dan SMSC penerima mengirimkan pesan ke ESME penerima. Jika status report diminta oleh pengirim pesan, maka SMSC penerima membuat status report dan mengirimkannya ke ESME pengirim.

BAB III

DESAIN DAN IMPLEMENTASI SISTEM

3.1. Deskripsi Sitem

Deskripsi sistem adalah gambaran tentang sistem yang akan dibangun untuk mengetahui kecepatan serta rute angkutan kota menggunakan aplikasi di smart phone android. Apikasi ini dibangun sebagai sistem perekaman kecepatan dan rute angkutan kota. Dimana system akan merekam pergerakan setiap radius minimal 500 meter, namun akurasi waktu perekman juga tergantung kondisi jaringan daerah tersebut.

Disini sistem akan merekam pergerakan kendaraan namun hanya akan mengirimkan laporan berupa sms ke user hanya jika kendaraan tersebut keluar dari rute yang telah ditentukan atau melebihi kecepatan maksimal yang telah di tentukan.

3.2. Analisa Sistem

Proses awal yang dilakukan adalah menentukan lintang dan bujur suatu daerah, menentukan batas kecepatan angkutan serta rute angkutan kota tersebut. Hal ini di lakukan mengingat aplikasi di smart phone android ini di dukung oleh system GPS. Aplikasi ini di desain untuk merekam kecepatan angkutan yang melebihi kecepatan maksimal yang telah ditentukan serta angkutan kota yang keluar dari rute yang telah ditentukan.

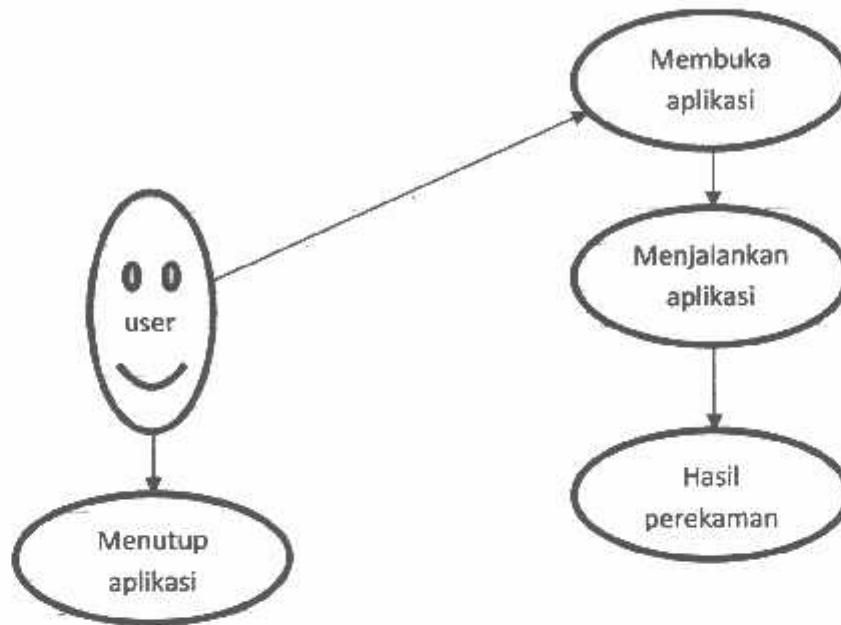
Untuk mengetahui batas rute angkutan ditentukan dengan cara menentukan lintang dan bujur, sedangkan untuk menentukan batas kecepatan angkutan dihitung menggunakan menggunakan rumus dasar kecepatan rata – rata: $V = \frac{S}{T}$

Keterangan : V= Kecepatan
S= Jarak
T= Waktu

3.3. Tahap Perancangan

Pada tahap ini di lakukan perancangan aplikasi yang terdiri dari rancangan use case diagram, rancangan diagram alir, rancangan tampilan awal serta rancangan tampilan hasil perekaman.

3.3.1. Rancangan Use Case Diagram

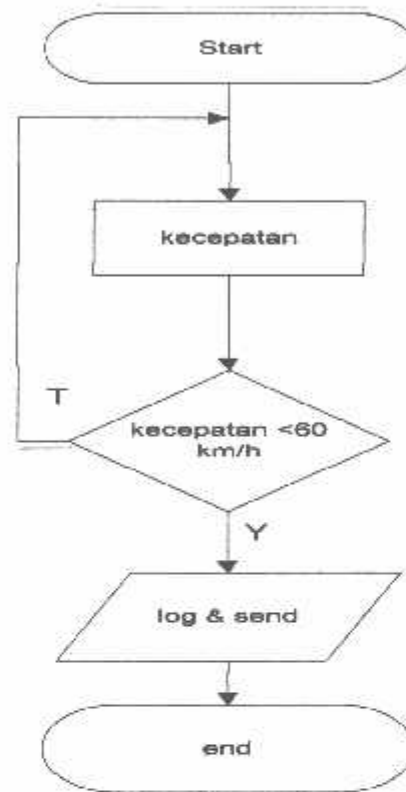


Gambar 3.1 Use case diagram

Use case diagram mendefenisikan fitur – fitur yang terdapat dalam system. Gambar 3.1 menunjukkan adanya interaksi antara aktor dengan system. Aktor yang berperan adalah user yang menjalankan aplikasi, sedangkan yang melakukan perekaman adalah smartphone android lalu dikirimkan secara otomatis ke pengguna aplikasi.

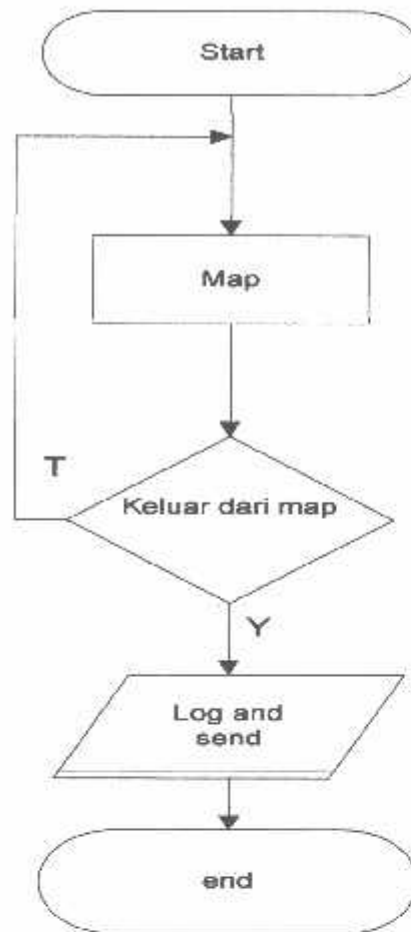
3.3.2. Rancangan Diagram Alir

Diagram Alir (Flowchart) adalah gambaran secara grafik yang terdiri dari simbol-simbol dari algoritma-algoritma dalam suatu program, yang menyatakan arah dari alur program. Diagram Alir (Flowchart) digunakan untuk membantu menganalisis untuk memecahkan masalah dalam program yang di buat. Berikut adalah rancangan diagram alir yang di buat untuk menjelaskan cara kerja aplikasi perekaman kecepatan dan rute angkutan kota.



Gambar 3.2 Diagram Alir Kecepatan

Dari gambar 3.2 Diagram alir kecepatan, dapat dilihat dengan jelas cara kerja aplikasi yaitu user sebagai pengaktif aplikasi di smartphone selanjutnya sistem mulai melakukan perekaman kecepatan. Disaat kecepatan angkutan kota melebihi batas kecepatan yang telah di tentukan, maka system akan secara otomatis mengirimkan laporan kecepatan kepada user.

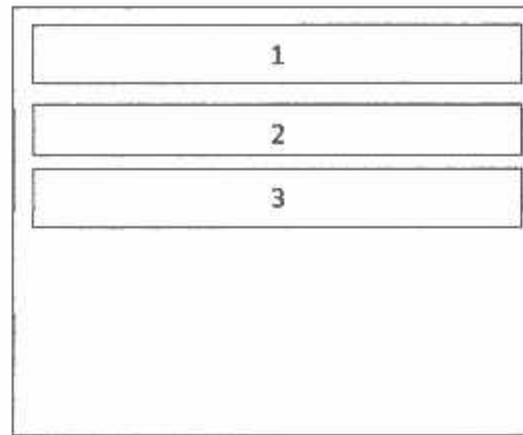


Gambar 3.3 Diagram Alir Rute

Dari gambar 3.3 Diagram alir rute, dapat dilihat dengan jelas cara kerja aplikasi yaitu user sebagai pengaktif aplikasi di smartphone selanjutnya sistem merekam pergerakan kendaraan sesuai dengan rute yang telah ditentukan. Disaat pergerakan kendaraan melewati rute yang telah di tentukan, maka system akan secara otomatis mengirimkan laporan jarak kepada user.

3.3.3. Rancangan Tampilan Awal

Rancangan Tampilan Awal dibuat untuk memberikan gambaran tampilan ketika aplikasi di buka, Adapun rancangan tampilan awal dapat di lihat pada gambar 3.4.



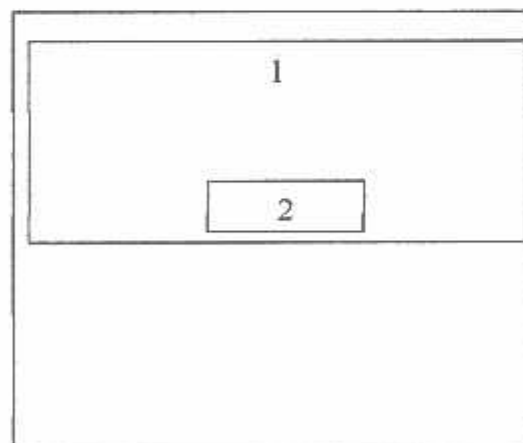
Gambar 3.4 Rancangan Tampilan Awal

Keterangan :

1. Header
2. Menu input "*plat nomor*"
3. Button "*ok*"

3.3.4. Rancangan Tampilan Menu Input Nomor Tujuan

Rancangan Tampilan input nomor tujuan dibuat untuk memberikan gambaran tampilan ketika user diminta untuk mengintupkan nomor telepon tujuan pengiriman data. Adapun rancangannya dapat di lihat pada gambar 3.4.



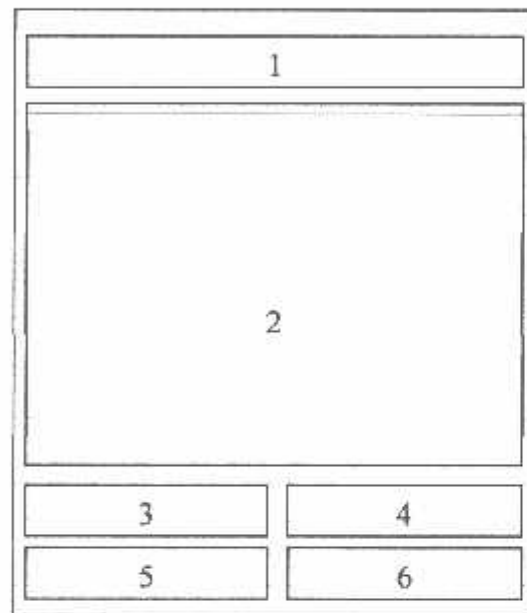
Gambar 3.5 Rancangan Input Nomor Tujuan

Keterangan:

1. Menu Input "*nomor telepon tujuan*"
2. Button "*ok*"

3.3.5. Rancangan Tampilan Hasil Perekaman Rute Dan Kecepatan

Rancangan Tampilan hasil perekaman dan rute bisa dilihat pada tampilan berikut, Adapun rancangan tampilan menu utama dapat di lihat pada gambar 3.5.



Gambar 3.6 Rancangan Hasil

Keterangan :

1. header
2. tampilan perekaman
3. Button "*start/stop GPS*"
4. Button "*last location*"
5. Button "*select log*"
6. Button "*phone number*"

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Aplikasi Perekaman Kecepatan Dan Rute Angkutan Kota

Pembuatan aplikasi perekaman kecepatan dan rute angkutan kota Flores Timur ini dilakukan dengan menerapkan hasil desain yang telah dibuat ke dalam bahasa pemrograman Java yakni Action script sehingga prosedur-prosedur yang dibuat dapat menghasilkan keluaran seperti yang diharapkan.

4.2 Pengujian “Aplikasi Perekaman Kecepatan Dan Rute Angkutan Kota”

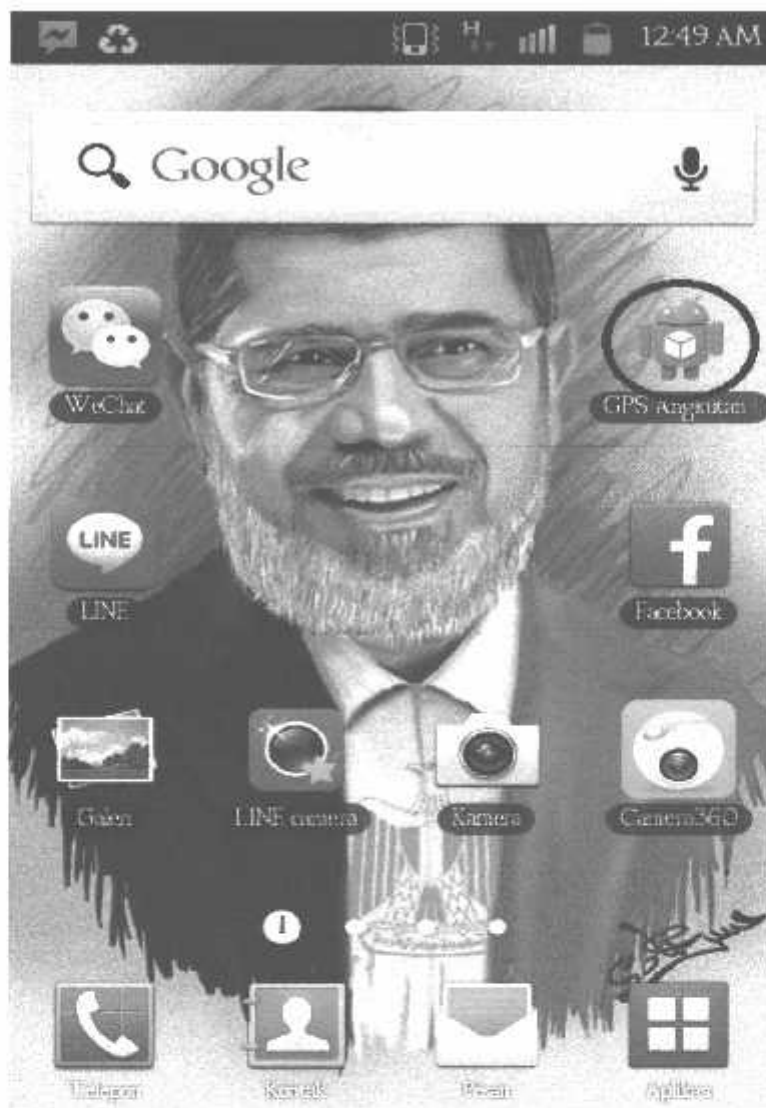
Dalam pembuatan aplikasi perekaman kecepatan dan rute angkutan kota dilakukan juga tahapan pengujian aplikasi untuk mengetahui fitur yang digunakan untuk menjalankan aplikasi perekaman kecepatan dan rute angkutan kota.

4.3 Tampilan “Aplikasi Perekaman Kecepatan Dan Rute Angkutan Kota”

Dalam pengujian Aplikasi Perekaman Kecepatan Dan Rute Angkutan Kota dapat dilihat beberapa tampilan aplikasi yang digunakan dalam pembuatan aplikasi ini.

4.3.1 Icon Aplikasi pada ponsel

Icon yang dilingkari dengan warna hitam adalah contoh tampilan Aplikasi Perekaman Kecepatan Dan Rute Angkutan Kota setelah diinstal pada salah satu ponsel Android.

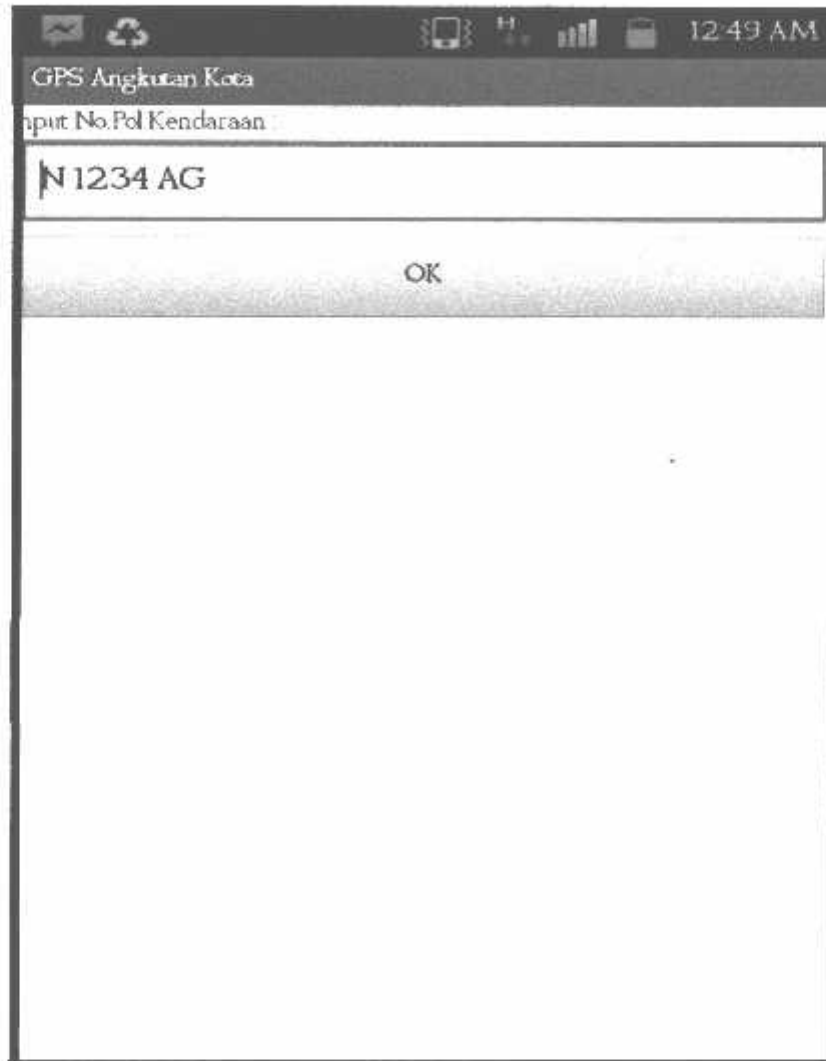


Gambar 4.1. Tampilan Icon Aplikasi

Untuk menjalankan aplikasi perekaman kecepatan dan rute angkutan kota ini, user cukup memilih icon yang ada pada layar smartphone android seperti pada gambar 4.1.

4.3.2. Tampilan menu input plat nomor

Tampilan pada gambar 4.2 merupakan tampilan menu input plat nomor kendaraan, dimana pada bagian ini, user menginput nomor kendaraan yang akan menggunakan aplikasi perekamana kecepatan dan rute angkutan kota.



Gambar 4.2. Tampilan Menu Input Plat Nomor

4.3.3. Tampilan Menu Input Nomor Tujuan

Tampilan pada gambar 4.3 merupakan tampilan dimana user menginput nomor telepon tujuan yang nantinya aplikasi akan mengirimkan laporan kecepatan dan rute secara otomatis ke nomor telepon yang telah di inputkan.



Gambar 4.3. Tampilan Menu Nomor Telepon Tujuan

4.3.4. Tampilan Button Start GPS

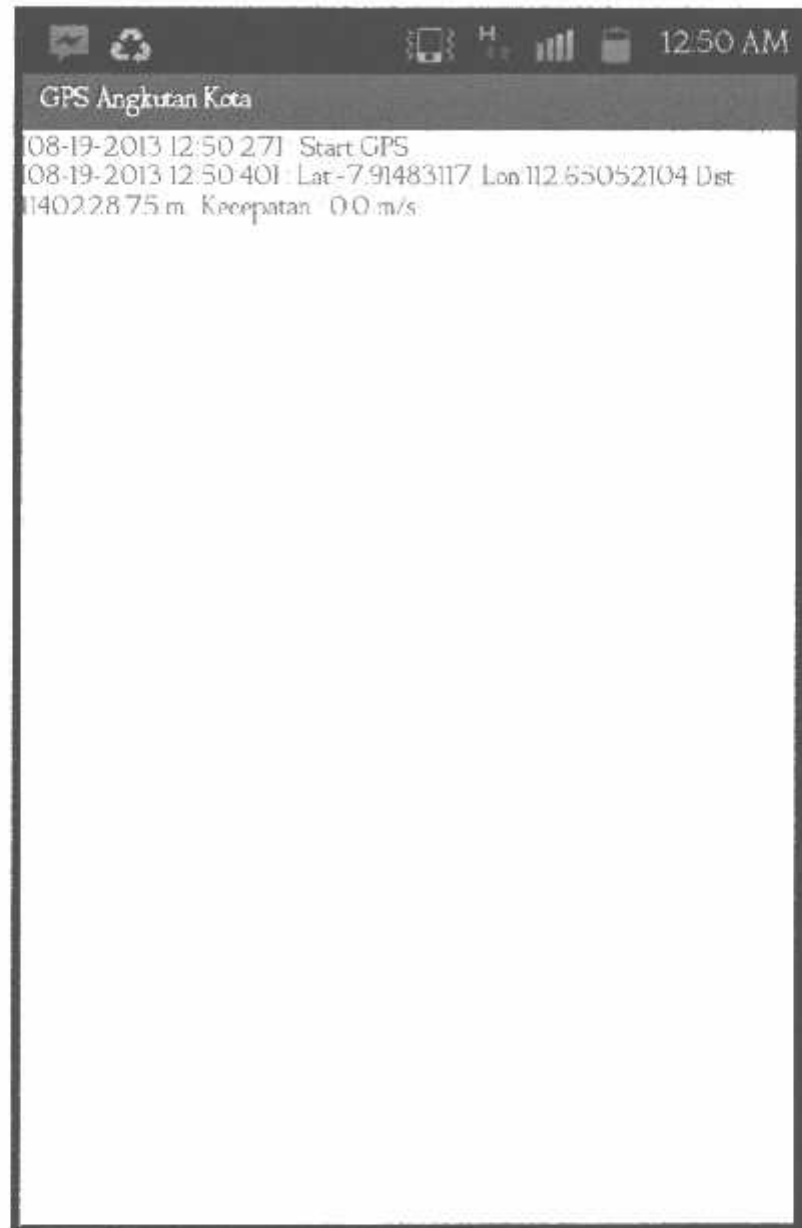
Tampilan pada gambar 4.4 merupakan tampilan dimana user mengaktifkan aplikasi dengan memilih button start GPS.



Gambar 4.4 Tampilan Start GPS

4.3.5. Tampilan Hasil Perekaman Pertama

Tampilan pada gambar 4.5 merupakan tampilan hasil perekaman pertama dimana kendaraan berada di luar jalur sejauh 1.140.228,75 Meter = 1.140,228 KM, dengan Kecepatan 0,0 m/s. Pada percobaan ini kondisi kendaraan dalam keadaan diam.



Gambar 4.5 Tampilan Hasil Perekaman Pertama

4.3.6. Tampilan Hasil Perekaman Ke-Dua

Tampilan pada gambar 4.6 merupakan tampilan hasil perekaman kedua dimana kendaraan berada di luar jalur sejauh 1.140.697,0 Meter = 1.140,697 KM dengan Kecepatan 14.9494505 m/s = 53,817 km/h. pada percobaan ini kondisi kendaraan dalam keadaan bergerak.



Gambar 4.6 Tampilan Hasil Perekaman ke Dua

4.3.7. Tampilan Hasil Perekaman Ke-Tiga

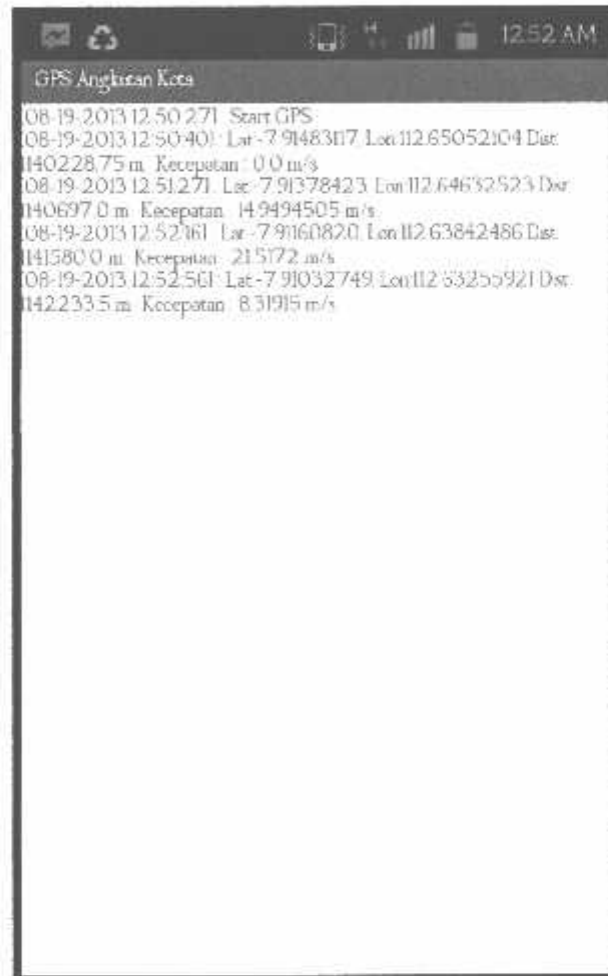
Tampilan pada gambar 4.7 merupakan tampilan hasil perekaman ketiga dimana kendaraan berada di luar jalur sejauh 1.140.580,0 Meter = 1.140,580 KM dengan Kecepatan 21.5175 m/s = 77,462 km/h. pada percobaan ini kondisi kendaraan dalam keadaan bergerak.



Gambar 4.7 Tampilan Hasil Perekaman ke Tiga

4.3.8. Tampilan Hasil Perekaman Ke-Empat

Tampilan pada gambar 4.8 merupakan tampilan hasil perekaman keempat dimana kendaraan berada di luar jalur sejauh 1.142.233,0 Meter = 1.142,233 KM dengan Kecepatan 8.3191 m/s = 29,984916 km/h. pada percobaan ini kondisi kendaraan dalam keadaan bergerak.



Gambar 4.8 Tampilan Hasil Perekaman ke Empat

4.3.9. Tampilan Hasil Perekaman Ke-Lima

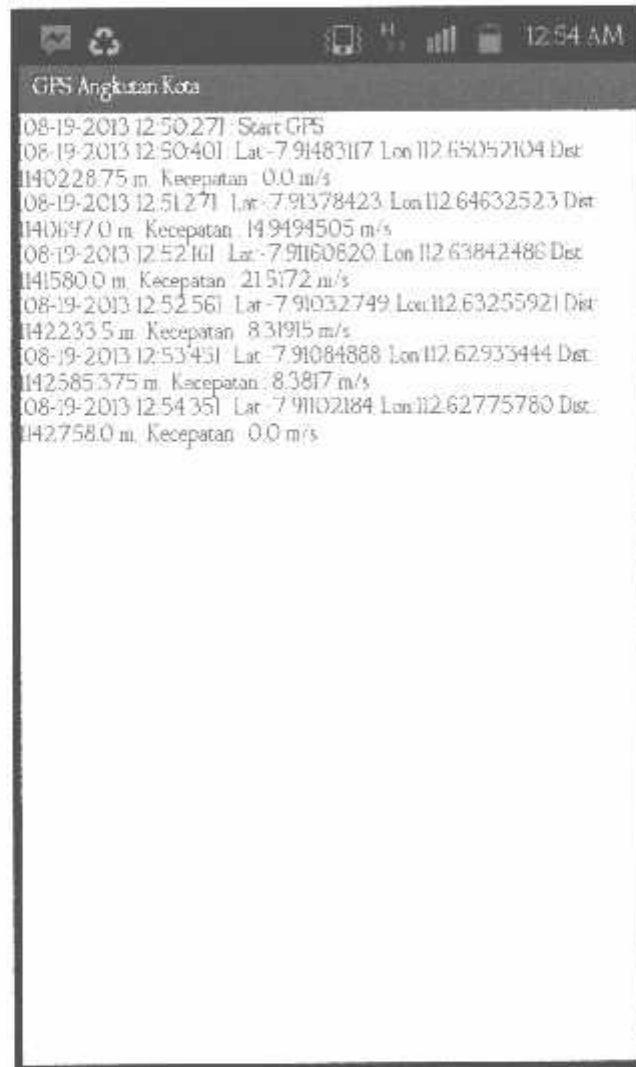
Tampilan pada gambar 4.9 merupakan tampilan hasil perekaman kelima dimana kendaraan berada di luar jalur sejauh 1.142.585,375 Meter = 1.142,585 KM dengan Kecepatan 8.33817 m/s = 30,174096 km/h. pada percobaan ini kondisi kendaraan dalam keadaan bergerak.



Gambar 4.9 Tampilan Hasil Perekaman ke Lima

4.3.10. Tampilan Hasil Perekaman Ke-Enam

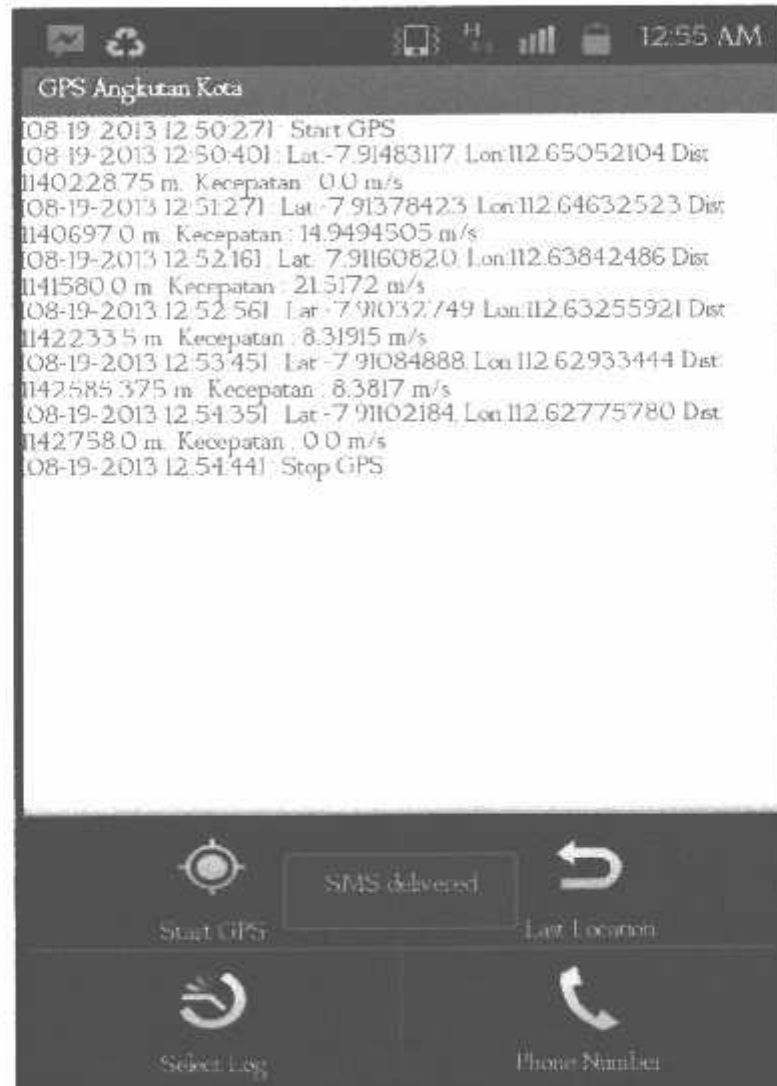
Tampilan pada gambar 4.10 merupakan tampilan hasil perekaman keenam dimana kendaraan berada di luar jalur sejauh 1.142.758,000 Meter = 1.142,758 KM dengan Kecepatan 0,0 m/s = 0,0 km/h. pada percobaan ini kondisi kendaraan dalam keadaan berhenti.



Gambar 4.10 Tampilan Hasil Perekaman ke Enam

4.3.11. Tampilan Button Stop GPS

Tampilan pada gambar 4.10 merupakan tampilan dimana button “stop” di pilih untuk menghentikan aplikasi.



Gambar 4.10 Tampilan Stop GPS

1.3.12 Tampilan Hasil Pengiriman Data Pada Nomor Telepon Tujuan

Tampilan pada gambar 4.11 dan 4.12 ini merupakan tampilan hasil yang dikirimkan oleh aplikasi ke nomor yang telah di inputkan pada tampilan awal sebelum aplikasi GPS di jalankan.



Gambar 4.11 Tampilan Laporan Start GPS Dan Penerimaan Data



Gambar 4.12 Tampilan Penerimaan Data Dan Stop GPS

4.4. Tampilan Rute Angkutan Kota Flores Timur Menggunakan Google Earth

Berikut merupakan tampilan rute angkutan kota flores timur yang di perolemmenggunakan google earth. Pada gambar ini penulis telah menandai rute dengan garis merah dan landmark. Dimana garis merah berfungsi sebagai penunjuk jalan dan Landmark sendiri berfungsi sebagai titik yang digunakan untuk menghitung jarak rute tersebut. Agar mendapatkan hasil tampilan yang maksimal, gambar di bagi menjadi dua bagian.



Gambar 4.13 Tampilan Rute Menggunakan Google Earth Dari Landmark 1 sampai



Gambar 4.14 Tampilan Rute Menggunakan Google Earth Dari Landmark 17 sampai 40

Berikut adalah data hasil pengukuran posisi dan jarak rute angkutan kota Flores Timur. Penentuan posisi dan pengukuran jarak menggunakan Google Earth.

| No | Placemark position | Latitude | longitude | Jarak Antara Posisi (m = meter) |
|----|--------------------|------------|-------------|---------------------------------|
| 1 | position 1 | -8.342283° | 122.988281° | 1 sampai 2= 52,3 m |
| 2 | position 2 | -8.341864° | 122.988053° | 2 sampai 3= 109m |
| 3 | position 3 | -8.341167° | 122.988767° | 3 samapai 4= 143m |
| 4 | position 4 | -8.340370° | 122.989785° | 4 samapai 5= 183m |
| 5 | position 5 | -8.339080° | 122.990819° | 5 sampai 6= 425m |
| 6 | position 6 | -8.338741° | 122.991015° | 6 sampai 7= 116m |
| 7 | position 7 | -8.337729° | 122.991263° | 7 sampai 8= 29m |
| 8 | position 8 | -8.337484° | 122.991357° | 8 sampai 9= 154m |
| 9 | position 9 | -8.336439° | 122.992281° | 9 sampai 10= 397m |
| 10 | position 10 | -8.334963° | 122.995559° | 10 sampai 11= 280m |

| No | Placemark position | Latitude | longitude | Jarak Antara Posisi (m = meter) |
|----|--------------------|------------|-------------|---------------------------------|
| 11 | position 11 | -8.333669° | 122.997742° | 11 sampai 12=83,5m |
| 12 | position 12 | -8.333220° | 122.998353° | 12 sampai 13=219m |
| 13 | position 13 | -8.331888° | 122.999825° | 13 sampai 14= 453m |
| 14 | position 14 | -8.328690° | 123.002396° | 14 sampai 15= 279m |
| 15 | position 15 | -8.326632° | 123.003863° | 15 sampai 16= 90,7m |
| 16 | position 16 | -8.325985° | 123.004376° | 16 sampai 17= 142m |
| 17 | position 17 | -8.325022° | 123.005225° | 17 sampai 18= 375m |
| 18 | position 18 | -8.322490° | 123.007494° | 18 sampai 19= 215m |
| 19 | position 19 | -8.321176° | 123.008935° | 19 sampai 20= 101m |
| 20 | position 20 | -8.320828° | 123.009780° | 20 sampai 21= 106m |
| 21 | Position 21 | -8.320573° | 123.010708° | 21 sampai 22=373m |
| 22 | Position 22 | -8.319041° | 123.013712° | 22 sampai 23= 34,5m |
| 23 | Position 23 | -8.318870° | 123.013977° | 23 sampai 24= 62,8m |
| 24 | Position 24 | -8.318476° | 123.014384° | 24 sampai 25= 57,5m |
| 25 | Position 25 | -8.318085° | 123.014728° | 25 sampai 26= 104m |
| 26 | Position 26 | -8.317277° | 123.015205° | 26 sampai 27= 41,2m |
| 27 | Position 27 | -8.316965° | 123.015410° | 27 sampai 28= 69m |
| 28 | Position 28 | -8.316462° | 123.015784° | 28 sampai 29= 101m |
| 29 | Position 29 | -8.315767° | 123.016378° | 29 sampai 30= 61,7m |
| 30 | Position 30 | -8.315343° | 123.016740° | 30 sampai 31= 119m |
| 31 | Position 31 | -8.314342° | 123.017131° | 31 sampai 32= 243m |
| 32 | Position 32 | -8.312319° | 123.017986° | 32 sampai 33= 203m |
| 33 | Position 33 | -8.310567° | 123.018506° | 33 sampai 34= 69,1m |
| 34 | Position 34 | -8.309694° | 123.018533° | 34 sampai 45= 264m |
| 35 | Position 35 | -8.307348° | 123.018966° | 35 sampai 36= 121m |
| 36 | Position 36 | -8.306255° | 123.018999° | 36 sampai 37= 117m |
| 37 | Position 37 | -8.305234° | 123.018766° | 37 sampai 38= 103m |
| 38 | Position 38 | -8.304302° | 123.018529° | 38 sampai 39= 75,6m |

| No | Placemark position | Latitude | longitude | Jarak Antara Posisi (m = meter) |
|--|--------------------|------------|-------------|---------------------------------|
| 39 | Position 39 | -8.303643° | 123.018553° | 39 sampai 40= 132m |
| 40 | Position 40 | -8.302464° | 123.018754° | 40 sampai 41= 241m |
| 41 | Position 41 | -8.340817° | 122.990375° | 41 sampai 42= 103m |
| 42 | Position 42 | -8.341392° | 122.989642° | 42 sampai 43= 68,8m |
| 43 | Position 43 | -8.341794° | 122.989166° | 43 sampai 1= 111m |
| Total jarak = 5.851,1 meter = 5,851 km | | | | |

Tabel 4.1 Hasil Pemetaan Dan Perhitungan Jarak Menggunakan Google Earth

4.5. Hasil Pengujian Keakurasian Posisi Dan Kecepatan

4.5.1. Hasil Uji Coba Keakurasian Posisi Menggunakan 3 Aplikasi GPS Yang Berbeda

Pada percobaan ini penulis menggunakan 3 aplikasi GPS yang berbeda pada satu buah smartphone android yang di uji pada dua tempat yang berbeda. Adapun aplikasi GPS yang digunakan antara lain:

1. GPS Angkutan Kota (aplikasi skripsi)
2. GPS Logger
3. One Touch Location

Pada pengujian pertama, hasil posisi yang di peroleh masih-masing GPS bebeda dari posisi asli. Percobaan pertama di lakukan di jln. Perusahaan singosari malang.



Gambar 4.15 hasil percobaan pertama menggunakan 3 aplikasi GPS yang berbeda

Data hasil pengujian pertama percobaan keakurasian posisi menggunakan 3 (tiga) aplikasi GPS yang berbeda pada smartphone Android. Posisi asli di tentukan dengan menggunakan Google Earth.

| no | GPS | Lat | Long | Selisih jarak dari posisi asli |
|----|--------------------|-------------|--------------|--------------------------------|
| 1 | Posisi Asli | -7.909823 | 112.635174 | |
| 2 | Gps Angkutan Kota | -7,90982734 | 112.63516522 | 2,25 meter |
| 3 | GPS Logger | -7,90985553 | 112.63508333 | 12,63 meter |
| 4 | One Touch Location | -7.90981 | 112.6352 | 3,03 meter |

Tabel 4.2 hasil percobaan pertama keakurasian posisi

Pada pengujian kedua, hasil posisi yang di peroleh masih-masing GPS berbeda dari posisi asli. Percobaan ke dua di lakukan di jln. Bulu l'angkis singosari malang.







Gambar 4.16 hasil percobaan kedua menggunakan 3 aplikasi GPS yang berbeda

Data hasil pengujian kedua percobaan keakurasian posisi menggunakan 3 (tiga) aplikasi GPS yang berbeda pada smartphone Android. Posisi asli di tentukan dengan menggunakan Google Earth.

| no | GPS | Lat | Long | Selisih jarak dari posisi asli |
|----|--------------------|--------------|--------------|--------------------------------|
| 1 | Posisi asli | -7.909823 | 112.635174 | |
| 2 | Gps Angkutan Kota | -7, 90982734 | 112.63516522 | 13, 47 meter |
| 3 | GPS Logger | -7,90985553 | 112.63508333 | 3, 35 meter |
| 4 | One Touch Location | -7.90981 | 112.6352 | 9, 29 meter |

Tabel 4.3 hasil percobaan kedua keakurasian posisi

Keterangan Gambar:

-  posisi asli,
-  Aplikasi GPS angkutan kota (skripsi)
-  Aplikasi GPS Logger, dan
-  Aplikasi One Touch Location

Dari ke dua hasil percobaan yang di lakukan, dapat disimpulkan bahwa keakurasian hasil data yang di terima tergantung pada kondisi jaringan satelit GPS pada tempat di lakukannya pengambilan data. Perbedaan jarak yang di dapat pada dua pengujian yang di lakukan, di dapatkan nilai error posisi tidak lebih dari 15 meter.

4.5.2 Hasil Uji Coba Keakurasian Kecepatan Menggunakan Aplikasi GPS Angkutan Kota

Pada pengujian keakurasian kecepatan ini, pengujian di lakukan dengan menggunakan kendaraan sepeda motor untuk membandingkan keakurasian kecepatan yang terdapat pada smart phone android dan yang terdapan pada sepeda motor itu sendiri.

Berikut merupakan hasil pengujian keakurasian kecepatan:



Gambar 4.17 hasil uji coba keakurasian kecepatan pada aplikasi GPS angkutan kota

Ini merupakan tampilan hasil pengujian kecepatan pada aplikasi perekaman kecepatan dan rute angkutan kota menggunakan smartphone android.



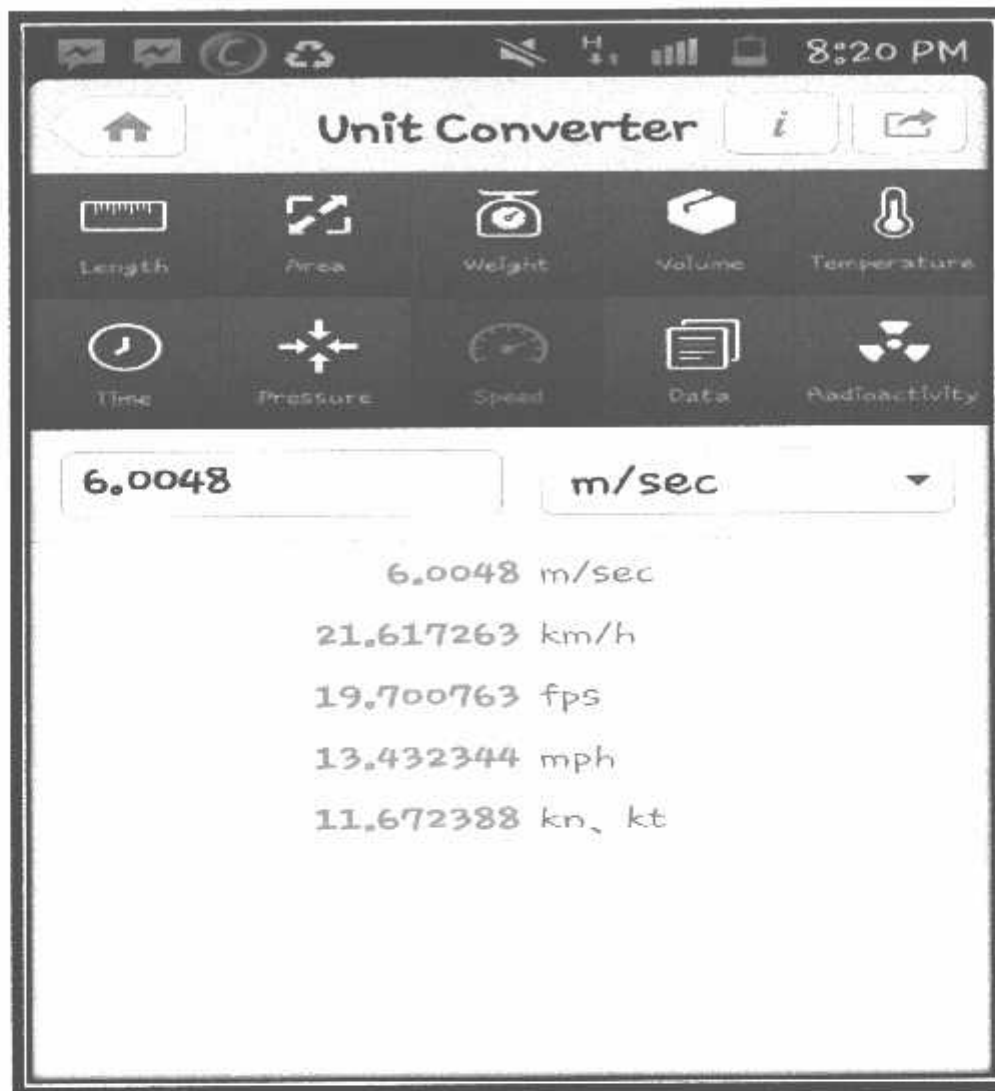
Gambar 4.18 tampilan speedometer pada posisi diam

Ini merupakan tampilan kecepatan pada sepeda motor di mana speedometer menunjukkan angka 0km/h. Pada saat gambar di ambil, sepeda motor dalam keadaan diam.



Gambar 4.19 tampilan speedometer pada posisi bergerak

Ini merupakan tampilan kecepatan pada sepeda motor di mana speedometer menunjukkan angka 21km/h. Pada saat gambar di ambil, sepeda motor dalam keadaan bergerak.



Gambar 4.20 Hasil konversi kecepatan

Ini merupakan tampilan hasil konversi data kecepatan dari hasil perekaman aplikasi pada smartphone android dari m/sec ke km/h. Konversi data kecepatan ini menggunakan software unit converter.

Jadi dari hasil percobaan di atas, dapat di tarik kesimpulan bahwa keakurasian kecepatan pada aplikasi perekaman kecepatan dan rute angkutan kota 99,4% dengan nilai error 0,6% sesuai dengan hasil yang di dapat pada pengujian kecepatan menggunakan sepeda motor.

BAB V

PENUTUP

Dalam bab ini akan disampaikan kesimpulan dan saran yang diperoleh dari hasil pembahasan dari BAB IV adalah sebagai berikut.

5.1. Kesimpulan

Dari hasil uji coba yang dilakukan di bab sebelumnya, diperoleh kesimpulan sebagai berikut :

1. Aplikasi akan mengirimkan data secara otomatis kepada nomor pengguna yang telah di inputkan disaat aplikasi merekam kecepatan yang melebihi batas kecepatan maksimal ataupun melewati rute yang telah di tentukan.
2. Dari beberapa percobaan yang di lakukan, dapat disimpulkan bahwa keakurasian hasil data yang di terima tergantung pada kondisi jaringan satelit GPS pada tempat aplikasi itu di jalankan.
3. Dari beberapa pengujian yang telah di lakukan, nilai error aplikasi pada pengujian jarak ialah 2,25 meter, dan nilai error kecepatan adalah 0,6%.
4. Dalam program aplikasi ini mampu memberikan solusi bagi para pemilik angkutan agar dapat lebih mudah dalam pemantauan kendaraannya dan diharapkan dapat mengurangi pengeluaran untuk perawatan angkutan.

5.2 Saran

Pada kenyataannya Program aplikasi yang di buat masih sederhana dan perlu pengembangan lebih lanjut pada program aplikasi perakaman rute dan kecepatan angkutan kota.

DAFTAR PUSTAKA

- [1] El-Rabbany, A. 2006. **Introduction to GPS (The Global Positioning System)**. Edisi ke-2 Canada: Artech House Publishers.
- [2] Broida, P. 2004. **How to Do Everything with Your GPS**. London: McGraw-Hill/Osborne.
- [3] Istiyanto, Efendy. 2004. **Rancangan Dan Implementasi Prototipe Sistem Kendali Jarak Jauh Berbasis AT89C52 Dan Layanan SMS GSM**. Yogyakarta: Gajah Mada University
- [4] Ansori¹, Pramadihanto, Ramadijanti, 2005. **Deteksi Kendaraan Bergerak Secara Real Time**. Surabaya: PENS-ITS.
- [5] Yousman, Yeyep. 2008. **Google Earth**. Yogyakarta.
- [6] Rozidi, R.I. 2004. **Membuat Sendiri SMS Gateway (ESME) Berbasis Protokol SMPP**, Yogyakarta.
- [7] Abidin, Hasanuddin Z. 2000. **Penentuan Posisi dengan GPS dan Aplikasinya**. Jakarta.
- [8] Budiawan, 2008. **Mobile Tracking GPS (GLOBAL POSITIONING SYSTEM) Melalui Media SMS (SHORT MESSAGE SERVICE)**. Semarang: Universitas Diponegoro.



LAMPIRAN

```
package com.gps.angkutan.trackker;

import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;

public class ColorSelectDialogClickListener implements
OnClickListener {

    private GpsLog log;
    public static String[] colors = {"Red", "Green", "Blue"};

    public ColorSelectDialogClickListener(GpsLog log){
        this.log = log;
    }

    public void onClick(DialogInterface dialog, int which) {
        // TODO Auto-generated method stub
        switch(which){
            case 0:
                log.setR(255);
                log.setG(0);
                log.setB(0);
                break;
            case 1:
                log.setR(0);
                log.setG(255);
                log.setB(0);
                break;
            case 2:
                log.setR(0);
                log.setG(0);
                log.setB(255);
                break;
        }
    }
}
```

```

package com.gps.angkutan.trackker;

import java.util.*;
import java.util.regex.*;
import java.io.*;
import android.app.AlertDialog;
import android.base.org.sun.java.Utills;
import android.os.Environment;
import android.util.Log;

public class GpsLog {

    private List<LogPoint> points;
    private GpsTracker context;
    private File dataDir;
    private int r, g, b;
    Utills u = new Utills();
    String fLog;
    public GpsLog(GpsTracker context){
        this.context = context;
        this.r = 255;
        this.g = 0;
        this.b = 0;

        this.points = new ArrayList<LogPoint>();
        if (this.hasSdCard()){
            this.dataDir = new File(this.getDir());
            fLog = (this.dataDir + "/db.rute");

            context.trace("data dir : "+ dataDir.getPath());
        }
        else{
            context.trace("SD Card not Found");
            new AlertDialog.Builder(context).setMessage("this
app uses SD Card").setPositiveButton("OK", null).show();
        }
        try{
            this.loadLog();
        }
        catch(Exception e){
            Log.e("GpsTracker", e.getMessage());
        }
    }
}

```

```

public String getfLog() {
    return fLog;
}

public String getDir(){
    if (!this.hasSdCard()) return "";
    dataDir = new
File(Environment.getExternalStorageDirectory(),
context.getPackageName()+"/log");
    if(!dataDir.exists()) dataDir.mkdirs();
    return dataDir.getPath();
}

public boolean hasSdCard(){
    return
Environment.getExternalStorageState().equals(Environment.MEDIA_M
OUNTED);
}

public List<LogPoint> getPoints(){
    return this.points;
}

public int size(){
    return this.points.size();
}

public LogPoint getPoint(int index){
    return this.points.get(index);
}

public void add(double lat, double lon){
    LogPoint p = new LogPoint((int) (lat * 1E6), (int) (lon
* 1E6));
    p.setR(r);
    p.setG(g);
    p.setB(b);
    context.trace("add points : " + p.toLog());
    try{
        saveLog(p);
    }
    catch(Exception e){
        Log.e("GpsTracker", e.getMessage());
    }
}

```

```

public LogPoint lastPoint(){
    return this.points.get(this.points.size()-1);
}

private void saveLog(LogPoint p) throws Exception{
    if (dataDir == null) return;
    File f = new File(this.dataDir, p.getYear()+"-
"+p.getMonth()+"-"+p.getDay());

    try{
        OutputStream os = new FileOutputStream(f, true); //
append mode
        os.write(p.toLog().getBytes());
        os.close();
    }
    catch(Exception e){
        throw e;
    }
    context.trace("saveLog : " + f.getPath());
}

public boolean loadLog() throws Exception{
    Calendar cal = Calendar.getInstance();
    int year = cal.get(Calendar.YEAR);
    int month = cal.get(Calendar.MONTH) + 1;
    int day = cal.get(Calendar.DAY_OF_MONTH);
    try{
        return this.loadLog(year, month, day);
    }
    catch(Exception e){
        throw e;
    }
}

public boolean loadLog(String fileName) throws Exception{
    try{
        File f = new File(this.dataDir, fileName);
        return this.loadLog(f);
    }
    catch(Exception e){
        throw e;
    }
}

public boolean loadLog(int year, int month, int day) throws
Exception{

```

```

File f = new File(this.dataDir, year+"-"+month+"-"+day);
try{
    return this.loadLog(f);
}
catch(Exception e){
    throw e;
}
}

public boolean loadLog(File f) throws Exception{
    if(!f.exists()) return false;
    context.trace("load Log - " + f.getPath());
    try{
        InputStream is = new FileInputStream(f);
        byte[] bytes = new byte[is.available()];
        is.read(bytes);
        is.close();
        String[] lines = new String(bytes).split("\n");
        List<LogPoint> points = new ArrayList<LogPoint>();
        for(String line : lines){
            LogPoint p = LogPoint.parse(line);
            points.add(p);
            context.trace(p.toLog());
        }
        if (points.size() > 0){
            this.points = points;
            return true;
        }
        return false;
    }
    catch(Exception e){
        throw e;
    }
}

public String[] fileNames(){
    ArrayList<String> files = new ArrayList<String>();
    for(String file : this.dataDir.list()){
        files.add(file);
    }

    Collections.sort(files, new FileNamesComparator());
    Collections.reverse(files);

    String[] result = new String[files.size()];
    for(int i = 0; i < files.size(); i++){
        result[i] = files.get(i);
    }
}

```

```

    }

    return result;
}

public class FileNamesComparator implements Comparator {
    Pattern pattern_ymd;

    public FileNamesComparator() {
        this.pattern_ymd = Pattern.compile("^([1-9][0-9]*)-
([1-9][0-9]*)-([1-9][0-9]*)$");
    }

    public int compare(Object obj_a, Object obj_b) {
        if
(!pattern_ymd.matcher(obj_a.toString()).matches()) return 1;
        else if
(!pattern_ymd.matcher(obj_b.toString()).matches()) return -1;
        String[] tmp_a = obj_a.toString().split("-");
        String[] tmp_b = obj_b.toString().split("-");
        int[] a = new int[3];
        int[] b = new int[3];
        for (int i = 0; i < 3; i++) {
            a[i] = Integer.parseInt(tmp_a[i]);
            b[i] = Integer.parseInt(tmp_b[i]);
        }
        if (a[0] > b[0]) return 1;
        if (a[0] < b[0]) return -1;
        if (a[1] > b[1]) return 1;
        if (a[1] < b[1]) return -1;
        if (a[2] > b[2]) return 1;
        if (a[2] < b[2]) return -1;
        return 0;
    }
}

public int getR() {
    return r;
}

public void setR(int r) {
    this.r = r;
}

public int getG() {
    return g;
}

```

```
public void setG(int g) {  
    this.g = g;  
}
```

```
public int getB() {  
    return b;  
}
```

```
public void setB(int b) {  
    this.b = b;  
}
```

```
}
```

```
package com.gps.angkutan.trackker;

import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Message;

import com.google.android.maps.*;

import android.annotation.TargetApi;
import android.app.*;
import android.base.org.sun.java.Utils;
import android.content.BroadcastReceiver;
import android.content.ContentResolver;
import android.content.ContentValues;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.location.*;
import android.telephony.SmsManager;
import android.util.FloatMath;
import android.util.Log;
import android.view.*;
import android.widget.*;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.regex.*;

@TargetApi(Build.VERSION_CODES.DONUT)
public class GpsTracker extends MapActivity implements
LocationListener {

    private TextView textViewMessage;
    private LocationManager lm;

    private final int zoom_default = 18;
    private boolean location_enabled, log_enabled;
    private LogOverlay logOverlay;
    private GpsLog log;
    private boolean paused;
```

```
Utils u = new Utils();
private String ID;
private String phoneNumber="";

private static class MenuId {
    private static final int START_GPS = 1;
    private static final int LAST_LOCATION = 2;
    private static final int SET_ZOOM = 3;
    private static final int PHONE_TOGGLE = 4;
    private static final int LOG_TOGGLE = 5;
    private static final int SELECT_LOGFILE = 6;
    private static final int SELECT_COLOR = 7;
    private static final int SEND_LOGFILE = 8;
}

public GpsTracker() {
    this.location_enabled = false;
    this.log_enabled = true;
    this.paused = false;
}

@Override
public void onPause(){
    super.onPause();
    trace("onPause");
    this.paused = true;
}

@Override
public void onResume(){
    super.onResume();
    trace("onResume");
    this.paused = false;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    this.log = new GpsLog(this);
    this.logOverlay = new LogOverlay(log);
}
```

```

        this.logOverlay.setMaxLines(500);
        Intent i = getIntent();
        ID = i.getStringExtra("ID") ;
        //sendSMS(phoneNumber, "No. Pol " + ID + " Start
Tracker");
        //addMessageToSend(phoneNumber, "No. Pol " + ID + "
Start Tracker");
        if (phoneNumber=="") {
            setPhone();
            sendSMS(phoneNumber, "No. Pol " + ID + " Start
Tracker");
        }
        this.textViewMessage = (TextView)
findViewById(R.id.txtData);
        lm = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

    }
    private double gps2m(float lat_a, float lng_a, float lat_b,
float lng_b) {
        float pk = (float) (180/3.14169);

        float a1 = lat_a / pk;
        float a2 = lng_a / pk;
        float b1 = lat_b / pk;
        float b2 = lng_b / pk;

        float t1 =
FloatMath.cos(a1)*FloatMath.cos(a2)*FloatMath.cos(b1)*FloatMath.
cos(b2);
        float t2 =
FloatMath.cos(a1)*FloatMath.sin(a2)*FloatMath.cos(b1)*FloatMath.
sin(b2);
        float t3 = FloatMath.sin(a1)*FloatMath.sin(b1);
        double tt = Math.acos(t1 + t2 + t3);

        return 6366000*tt;
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        boolean supRetVal = super.onCreateOptionsMenu(menu);
        menu.add(0, MenuId.START_GPS, 0, "Start
GPS").setIcon(android.R.drawable.ic_menu_mylocation);
        menu.add(0, MenuId.LAST_LOCATION, 0, "Last
Location").setIcon(android.R.drawable.ic_menu_revert);
    }

```

```

        menu.add(0, MenuId.SELECT_LOGFILE, 0, "Select
Log").setIcon(android.R.drawable.ic_menu_recent_history);
        menu.add(0, MenuId.PHONE_TOGGLE, 0, "Phone
Number").setIcon(android.R.drawable.ic_menu_call);
        return superRetVal;
    }
    protected String timex() {
        String ret = "";
        SimpleDateFormat format = new SimpleDateFormat("[MM-
dd-yyyy hh:mm:ss]");
        Calendar c = Calendar.getInstance();
        c.setTime(new Date());

        System.out.println(format.format(c.getTime()));
        ret = (format.format(c.getTime()));

        return ret;
    }
    private static final String TELEPHON_NUMBER_FIELD_NAME =
"address";
    private static final String MESSAGE_BODY_FIELD_NAME =
"body";
    private static final Uri SENT_MSGS_CONTET_PROVIDER =
Uri.parse("content://sms/sent");

    private void addMessageToSent(String telNumber, String
messageBody) {
        ContentValues sentSms = new ContentValues();
        sentSms.put(TELEPHON_NUMBER_FIELD_NAME, telNumber);
        sentSms.put(MESSAGE_BODY_FIELD_NAME, messageBody);

        ContentResolver contentResolver = getContentResolver();
        contentResolver.insert(SENT_MSGS_CONTET_PROVIDER,
sentSms);
    }

    private void sendSMS(String phoneNumber, String message )
    {
        if (phoneNumber=="") return;
        String SENT = "SMS_SENT";
        String DELIVERED = "SMS_DELIVERED";

        Intent emailIntent = new Intent(Intent.ACTION_SEND);

```

```

PendingIntent sentPI = PendingIntent.getBroadcast(this,
0,
    new Intent(SENT), 0);

PendingIntent deliveredPI =
PendingIntent.getBroadcast(this, 0,
    new Intent(DELIVERED), 0);

//---when the SMS has been sent---
registerReceiver(new BroadcastReceiver(){
    @Override
    public void onReceive(Context arg0, Intent arg1) {
        switch (getResultCode())
        {
            case Activity.RESULT_OK:
                Toast.makeText(getBaseContext(), "SMS
sent",
                    Toast.LENGTH_SHORT).show();
                break;
            case
SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                Toast.makeText(getBaseContext(),
"Generic failure",
                    Toast.LENGTH_SHORT).show();
                break;
            case SmsManager.RESULT_ERROR_NO_SERVICE:
                Toast.makeText(getBaseContext(), "No
service",
                    Toast.LENGTH_SHORT).show();
                break;
            case SmsManager.RESULT_ERROR_NULL_PDU:
                Toast.makeText(getBaseContext(), "Null
PDU",
                    Toast.LENGTH_SHORT).show();
                break;
            case SmsManager.RESULT_ERROR_RADIO_OFF:
                Toast.makeText(getBaseContext(), "Radio
off",
                    Toast.LENGTH_SHORT).show();
                break;
        }
    }
}, new IntentFilter(SENT));

//---when the SMS has been delivered---
registerReceiver(new BroadcastReceiver(){
    @Override

```

```

        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode())
            {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS
delivered",
                                Toast.LENGTH_SHORT).show();
                    break;
                case Activity.RESULT_CANCELED:
                    Toast.makeText(getBaseContext(), "SMS
not delivered",
                                Toast.LENGTH_SHORT).show();
                    break;
            }
        }, new IntentFilter(DELIVERED));
        SmsManager sms = SmsManager.getDefault();
        sms.sendTextMessage(phoneNumber, null, message, sentPI,
deliveredPI);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case MenuId.START_GPS:
                trace("Menu - Start GPS");
                if (!this.location_enabled) {

                    lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 30*1000,
                    100, this); // 30(sec), 100(meter)

                    message("Start GPS");
                    this.location_enabled = true;
                    item.setTitle("Stop GPS");
                    if (phoneNumber=="") setPhone();
                    sendSMS(phoneNumber, "No. Pol " + ID + " Start
GPS");
                } else {
                    lm.removeUpdates(this);

                    message("Stop GPS");
                    this.location_enabled = false;
                    item.setTitle("Start GPS");
                    sendSMS(phoneNumber, "No. Pol " + ID + " Stop
GPS");
                }
                break;
            }
    }

```

```

case MenuId.LAST_LOCATION:
    trace("Menu - Last Location");
    if(log.size() > 0){
        LogPoint p = log.lastPoint();
        double lat = p.getLatitudeE6()/1E6;
        double lon = p.getLongitudeE6()/1E6;
        message("last lat:" + Double.toString(lat) + ",
lon:" + Double.toString(lon));
        this.setPosition(lat, lon, this.zoom_default);
    }
    else{
        new AlertDialog.Builder(this).setMessage("No
Logs. Please Start GPS").setPositiveButton("OK", null).show();
    }
    break;
case MenuId.SET_ZOOM:
    trace("Menu - Set Zoom");

    break;
case MenuId.PHONE_TOGGLE:
    trace("Menu - PHONE Toggle");
    setPhone();
    break;
case MenuId.LOG_TOGGLE:
    trace("Menu - Log Toggle : " + !log_enabled);
    if (this.log_enabled != true) {

        item.setTitle("Hide Log");
        message("logs: " +
Integer.toString(log.size()));
        log_enabled = true;
    } else {

        item.setTitle("Show Log");
        log_enabled = false;
    }

    break;
case MenuId.SELECT_LOGFILE:
    trace("Menu - Select Log");
    if(log.fileNames().length < 1){
        new AlertDialog.Builder(this).setMessage("No
Logs").setPositiveButton("OK", null).show();
    }
    else{
        new AlertDialog.Builder(this).setTitle("Select
Log")

```

```

        .setItems(log.fileNames(), new
LoadLogFileSelectDialogClickListener(this, log))
        .show();
    }
    break;
case MenuId.SELECT_COLOR:
    trace("Menu - Select Color");
    new AlertDialog.Builder(this).setTitle("Select Log
Color")
        .setItems(ColorSelectDialogClickListener.colors,
new ColorSelectDialogClickListener(log))
        .show();
    break;
case MenuId.SEND_LOGFILE:
    trace("Menu - Send Log File");
    new AlertDialog.Builder(this).setTitle("Select Log")
        .setItems(log.fileNames(), new
SendLogFileSelectDialogClickListener(this, log))
        .show();
    break;
}
return true;
}

private void setPhone() {
    // TODO Auto-generated method stub
    final EditText input = new EditText(this);
    input.setText(phoneNumber);
    AlertDialog alertDialog = new
AlertDialog.Builder(this).create();
    alertDialog.setTitle("Phone Number");
    alertDialog.setMessage("Enter phone number :");
    alertDialog.setView(input, 10, 0, 10, 0); // 10 spacing,
left and right
    alertDialog.setButton("Ok", new
DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int
which) {

            // TODO Auto-generated method stub
            phoneNumber = input.getText().toString();
        }
    });

    alertDialog.show();
}

```

```

    public void setPosition(double lat, double lon, int zoom) {
        GeoPoint p = new GeoPoint((int) (lat * 1E6), (int) (lon
* 1E6));

    }

    public void setPosition(double lat, double lon) {

    }

    public void message(String mes) {
        trace("message - " + mes);
        mes =
Pattern.compile("(\\d+\\.\\d{8})\\d+").matcher(mes).replaceAll("$1");
        this.textViewMessage.append(String.format("timex() +" :
%s \n", mes));
    }

    public void trace(String message){
        Log.v("GpsTracker", message);
    }

    public static double distFrom(double lt, double lg, double
d, double e) {
        double earthRadius = 3958.75;
        double dLat = Math.toRadians(d-lt);
        double dLng = Math.toRadians(e-lg);
        double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
            Math.cos(Math.toRadians(lt)) *
Math.cos(Math.toRadians(d)) *
            Math.sin(dLng/2) * Math.sin(dLng/2);
        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
        double dist = earthRadius * c;

        int meterConversion = 1609;

        return new Double(dist * meterConversion).floatValue();
    }

```

```

//public double CalculationByDistance(GeoPoint StartP,
GeoPoint EndP) {
    public double CalculationByDistance(double startLatitude,
double startLongitude, double endLatitude, double endLongitude
) {
        /*double lat1 = StartP.getLatitudeE6()/1E6;
double lat2 = EndP.getLatitudeE6()/1E6;
double lon1 = StartP.getLongitudeE6()/1E6;
double lon2 = EndP.getLongitudeE6()/1E6;
double dLat = Math.toRadians(lat2-lat1);
double dLon = Math.toRadians(lon2-lon1);
double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
        Math.cos(Math.toRadians(lat1)) *
Math.cos(Math.toRadians(lat2)) *
        Math.sin(dLon/2) * Math.sin(dLon/2);
double c = 2 * Math.asin(Math.sqrt(a));
return 6.371 * c; */

        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];
    }
@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    super.onBackPressed();
    sendSMS(phoneNumber, "No. Pol " + ID + " End Tracker");
}
public void onLocationChanged(Location location) {
    double lat = location.getLatitude();
    double lon = location.getLongitude();
    log.add(lat, lon);

    String [] strDB = u.files.toArray(log.getfLog(), "|");
    double MinDistance = 1E18;
    for (int i = 0; i < strDB.length; i++) {

```

```

        String string = strDB[i].trim();
        if (string.trim().length() > 0) {
            String str1 =
string.split(";")[0].trim().toString() ;
            String str2 =
string.split(";")[1].trim().toString() ;
            double lt =      (lat  );
            double lg =      (lon  );
            System.out.println("String : " +
string);
            if (!str1.equals("") &&
!str2.equals("")) {
                lt =      (Double.parseDouble(
str2));
                lg =      (Double.parseDouble(
str1));
                //double dis = distFrom(lt, lg,
(lat * 1E6), (lon * 1E6)) - 619107.8125;
                double dis =
CalculationByDistance(lt, lg, lat, lon);
                System.err.println(dis);
                if (dis < MinDistace ) {
                    MinDistace = dis;
                }
            }
        }
    }

    if(!paused){
        //if (MinDistace > 100 )
        {
            message("Lat:" + Double.toString(lat) + ", Lon:"
+ Double.toString(lon) + " Dist: " + MinDistace+ " m, Kecepatan
: " + location.getSpeed() + " m/s" );
        }
        if (MinDistace > 500 ) {
            //addMessageToSent(phoneNumber, "No. Pol " + ID +
" Jarak: " + MinDistace +" meter, Kecepatan : " +
location.getSpeed());
            sendSMS(phoneNumber, "No. Pol " + ID + " Jarak: "
+ MinDistace +" meter, Kecepatan : " + location.getSpeed() + "
m/s" );
        }
    }
}

```

```
        this.setPosition(lat, lon);
    }
}

public void onProviderDisabled(String provider) {
}

public void onProviderEnabled(String provider) {
}

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

@Override
protected boolean isRouteDisplayed() {
    return false;
}
}
```

```

package com.gps.angkutan.trackker;

import com.gps.angkutan.Trackker.R.id;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class InputActivity extends Activity {
    EditText editPlat ;
    Button buttonOK;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.input);
        buttonOK = (Button) findViewById(id.btnOk);
        editPlat = (EditText) findViewById(id.txtNopol);
        editPlat.setText("N 1234 AG");
        buttonOK.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                // TODO Auto-generated method stub
                if
(editPlat.getText().toString().trim().length() !=0){
                    showNext();
                }else {
                    t("Inputkan Nomor Polisi..");
                }
            }
        });
    }
    public void t(String s) {
        Toast.makeText(getApplicationContext(), s,
Toast.LENGTH_LONG).show();
    }
    protected void showNext() {
        Intent intents = new Intent( this, GpsTracker.class);

```

```
        intents.putExtra("ID",  
editPlat.getText().toString().trim() );
```

```
        startActivity(intents);
```

```
    }  
}
```

```

package com.gps.angkutan.trackker;

import com.google.android.maps.GeoPoint;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.util.Log;

public class LoadLogFileSelectDialogClickListener implements
OnClickListener {

    private GpsLog log;
    private GpsTracker gpsTracker;

    public LoadLogFileSelectDialogClickListener(GpsTracker
gpsTracker, GpsLog log){
        this.log = log;
        this.gpsTracker = gpsTracker;
    }

    public void onClick(DialogInterface dialog, int which) {
        try{
            String name = log.fileNames()[which];
            log.loadLog(name);
            gpsTracker.message("load : " + name + " - " +
log.size() + " logs");
            GeoPoint p = log.getPoint(log.size()-1);
            double lat = ((double)p.getLatitudeE6()) / 1E6;
            double lon = ((double)p.getLongitudeE6()) / 1E6;
            gpsTracker.setPosition(lat, lon);
        }
        catch(Exception e){
            Log.e("GpsTracker", e.getMessage());
        }
    }

}

```

```

package com.gps.angkutan.trackker;

import java.util.*;
import android.graphics.*;
import android.util.Log;
import com.google.android.maps.*;

public class LogOverlay extends Overlay {

    private GpsLog log;
    private int maxLines = 500;

    public LogOverlay(GpsLog log) {
        this.log = log;
    }

    public void setMaxLines(int num){
        if (num < 1) return;
        this.maxLines = num;
    }

    @Override
    public void draw(Canvas canvas, MapView view, boolean
shadow) {
        List<LogPoint> points = log.getPoints();
        int size = points.size();
        if (size < 2) return;

        GeoPoint center = view.getMapCenter();
        int top = center.getLatitudeE6() -
view.getLatitudeSpan()*2;
        int bottom = center.getLatitudeE6() +
view.getLatitudeSpan()*2;
        int left = center.getLongitudeE6() -
view.getLongitudeSpan()*2;
        int right = center.getLongitudeE6() +
view.getLongitudeSpan()*2;

        boolean[] visibles = new boolean[size];
        int visibles_num = 0;
        for(int i = 0; i < size; i++){
            GeoPoint p = points.get(i);
            if( top < p.getLatitudeE6() && p.getLatitudeE6() <
bottom &&
                left < p.getLongitudeE6() && p.getLongitudeE6()
< right ){

```

```

        visibles[i] = true;
        visibles_num++;
    }
    else{
        visibles[i] = false;
    }
}

//
ASac A»Aci;AAYAnac"AlI;A.Afi;AASÁ-Á;Afaesae"i;ki;ASac A»
AlI;A,,cAfaesae'AlI;A.Afaesi;Afi;A@GeoPointAfaesae"AlE'Á"Alac
SA'ALE'E+AfaesAçAlE'E'AlE'ae"
    boolean[] borders = new boolean[size];
    for (int i = 0; i < size; i++) {
        if (!visibles[i]) {
            if ((i > 0 && visibles[i - 1]) ||
                (i < size - 1 && visibles[i + 1])) {
                borders[i] = true;
            }
        }
    }

    Point pa = new Point();
    Point pb = new Point();
    LogPoint la, lb;
    int count = 0;
    int visible_ratio = visibles_num / this.maxLines + 1;
    if(visible_ratio > 1) Log.v("GpsTracker", "visibles : "
+ visibles_num + ", max : " + this.maxLines);
    for (int i = 0; i < size - 1; i++) {
        if(visibles[i] && visibles[i+1]){
            la = points.get(i);
            if(count++ % visible_ratio != 0) i++;
            if(i >= size-1) break;
            if(visibles[i+1] || borders[i+1]){
                lb = points.get(i+1);
                view.getProjection().toPixels(la, pa);
                view.getProjection().toPixels(lb, pb);
                canvas.drawLine(pa.x, pa.y, pb.x, pb.y,
la.getPaint());
            }
        }
        else if ((visibles[i] && borders[i+1]) ||
(borders[i] && visibles[i+1])) {
            la = points.get(i);
            lb = points.get(i+1);
            view.getProjection().toPixels(la, pa);

```

```
        view.getProjection().toPixels(lb, pb);
        canvas.drawLine(pa.x, pa.y, pb.x, pb.y,
la.getPaint());
    }
}

public int size() {
    return log.getPoints().size();
}
}
```

```

package com.gps.angkutan.trackker;

import java.util.Calendar;
import android.graphics.Paint;
import android.graphics.Paint.Style;
import android.util.Log;

import com.google.android.maps.GeoPoint;

public class LogPoint extends GeoPoint {

    private Paint paint;
    private int year, month, day, hour, min, sec;
    private int r, g, b;

    public LogPoint(int latitudeE6, int longitudeE6) {
        super(latitudeE6, longitudeE6);
        this.r = 255;
        this.g = 0;
        this.b = 0;

        this.paint = new Paint();
        this.paint.setStrokeWidth(2);
        this.paint.setDither(true);
        this.paint.setStyle(Style.FILL);
        this.paint.setAntiAlias(true);
        this.paint.setStrokeJoin(Paint.Join.ROUND);
        this.paint.setStrokeCap(Paint.Cap.ROUND);

        Calendar cal = Calendar.getInstance();
        this.year = cal.get(Calendar.YEAR);
        this.month = cal.get(Calendar.MONTH) + 1;
        this.day = cal.get(Calendar.DAY_OF_MONTH);
        this.hour = cal.get(Calendar.HOUR_OF_DAY);
        this.min = cal.get(Calendar.MINUTE);
        this.sec = cal.get(Calendar.SECOND);
    }

    public String toLog(){
        double lat = ((double)this.getLatitudeE6())/1E6;
        double lon = ((double)this.getLongitudeE6())/1E6;
        String str = new String(lat+", " +
            lon+", " +
            hour+": "+min+": "+sec +", " +
            r+": "+g+": "+b +
            "\n");

        return str;
    }
}

```

```

    }

    public static LogPoint parse(String log) throws Exception{
        String[] items = log.split("[ ]*,[ ]*");
        if(items.length < 2) throw new Exception("log format
error");
        else{
            LogPoint p;
            double lat = Double.parseDouble(items[0]);
            double lon = Double.parseDouble(items[1]);
            p = new LogPoint((int) (lat * 1E6), (int) (lon *
1E6));

            if(items.length > 2){
                String[] time = items[2].split(":");
                p.setHour(Integer.parseInt(time[0]));
                p.setMin(Integer.parseInt(time[1]));
                p.setSec(Integer.parseInt(time[2]));
            }
            if(items.length > 3 ){
                String[] color = items[3].split(":");
                p.setR(Integer.parseInt(color[0]));
                p.setG(Integer.parseInt(color[1]));
                p.setB(Integer.parseInt(color[2]));
            }
            return p;
        }
    }

    public Paint getPaint(){
        this.paint.setARGB(255, r, g, b);
        return paint;
    }

    public int getYear() {
        return year;
    }

    public int getMonth() {
        return month;
    }

    public int getDay() {
        return day;
    }

    public int getHour() {
        return hour;
    }

```

```
}

public int getMin() {
    return min;
}

public int getSec() {
    return sec;
}

public void setYear(int year) {
    this.year = year;
}

public void setMonth(int month) {
    this.month = month;
}

public void setDay(int day) {
    this.day = day;
}

public void setHour(int hour) {
    this.hour = hour;
}

public void setMin(int min) {
    this.min = min;
}

public void setSec(int sec) {
    this.sec = sec;
}

public int getR() {
    return r;
}

public void setR(int r) {
    this.r = r;
}

public int getG() {
    return g;
}

public void setG(int g) {
```

```
        this.g = g;
    }

    public int getB() {
        return b;
    }

    public void setB(int b) {
        this.b = b;
    }

}
```

```

package com.gps.angkutan.trackker;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.DialogInterface.OnClickListener;
import android.net.Uri;
import android.util.Log;

public class SendLogFileSelectDialogClickListener implements
OnClickListener {

    private GpsLog log;
    private GpsTracker gpsTracker;

    public SendLogFileSelectDialogClickListener(GpsTracker
gpsTracker, GpsLog log){
        this.log = log;
        this.gpsTracker = gpsTracker;
    }

    public void onClick(DialogInterface dialog, int which) {
        try{
            String name = log.fileNames()[which];
            if(!log.hasSdCard()) return;
            String fullName = log.getDir() + "/" + name;

            Intent it = new Intent();
            it.putExtra(Intent.EXTRA_SUBJECT, "GpsTracker Log: "
+ name);
            it.setAction(Intent.ACTION_SEND);
            it.putExtra(Intent.EXTRA_STREAM, Uri.parse("file://"
+ fullName));
            it.setType("text/plain");
            gpsTracker.startActivity(it);
        }
        catch(Exception e){
            Log.e("GpsTracker", e.getMessage());
        }
    }
}

```

```

package com.gps.angkutan.trackker;

import java.io.BufferedReader;
import java.io.File;
import java.io.InputStream;
import java.io.InputStreamReader;

import com.google.android.maps.GeoPoint;

import android.app.Activity;
import android.base.org.sun.java.Utils;
import android.content.Context;
import android.content.Intent;

import android.location.Location;
import android.os.Bundle;
import android.os.Environment;
import android.widget.ProgressBar;

public class SplashScreenAct extends Activity{

    ProgressBar pbMain;
    File dataDir;
    Utils u = new Utils();
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);
        setTitle("Selamat Datang..");

        dataDir = new
File(Environment.getExternalStorageDirectory(),
getApplicationContext().getPackageName()+"/log");
        String fLog = (this.dataDir +"/db.rute");
        u.files.Reset(fLog);
        String strTmp = ReadFromfile("rute",
getApplicationContext());
        System.out.println("Rute : " + strTmp);
        u.files.Append(fLog, strTmp);

        GeoPoint StartP = new GeoPoint( (int) (-
8.342005074897214 * 1E6) , (int) (122.9885859963612 * 1E6) );
        GeoPoint EndP = new GeoPoint( (int) (-
8.342010960773527 * 1E6), (int) (122.9885893221303 * 1E6) );

```

```

Location locationA = new Location("point A");

locationA.setLatitude(-8.342281089031602);
locationA.setLongitude(122.9882800823799 );

Location locationB = new Location("point B");

locationB.setLatitude(-8.341865397594729);
locationB.setLongitude(122.9880544100169);
float [] dist = new float[1];
    Location.distanceBetween(locationA.getLatitude(),
locationA.getLongitude()
locationB.getLatitude(),locationB.getLongitude(), dist);
    System.out.println("Test : " + dist[0]);

final int welcomeScreenDisplay =100;
pbMain = (ProgressBar) findViewById(R.id.pbMain );
pbMain.setMax(welcomeScreenDisplay);
pbMain.setProgress(0);

Thread threads = new Thread() {

    int wait = 0;

    @Override
    public void run() {
        try {
            super.run();

            while (wait < welcomeScreenDisplay) {
                sleep(10);
                wait += 1;
                pbMain.setProgress(wait);
            }
        } catch (Exception e) {
            System.out.println("Error:" + e);
        } finally {

            newIntens();
        }
    }
};
threads.start();
}

```

```

    public String ReadFromfile(String fileName, Context
context) {
    StringBuilder ReturnString = new StringBuilder();
    InputStream fIn = null;
    InputStreamReader isr = null;
    BufferedReader input = null;
    try {
        fIn = context.getResources().getAssets()
            .open(fileName,
context.MODE_WORLD_READABLE);
        isr = new InputStreamReader(fIn);
        input = new BufferedReader(isr);
        String line = "";
        while ((line = input.readLine()) != null) {
            ReturnString.append(line);
        }
    } catch (Exception e) {
        e.getMessage();
    } finally {
        try {
            if (isr != null)
                isr.close();
            if (fIn != null)
                fIn.close();
            if (input != null)
                input.close();
        } catch (Exception e2) {
            e2.getMessage();
        }
    }
    return ReturnString.toString();
}

    public double CalculationByDistance(GeoPoint StartP,
GeoPoint EndP) {
        double lat1 = StartP.getLatitudeE6()/1E6;
        double lat2 = EndP.getLatitudeE6()/1E6;
        double lon1 = StartP.getLongitudeE6()/1E6;
        double lon2 = EndP.getLongitudeE6()/1E6;
        double dLat = Math.toRadians(lat2-lat1);
        double dLon = Math.toRadians(lon2-lon1);
        double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
            Math.cos(Math.toRadians(lat1)) *
Math.cos(Math.toRadians(lat2)) *
            Math.sin(dLon/2) * Math.sin(dLon/2);

```

```
        double c = 2 * Math.asin(Math.sqrt(a));  
        return 6.371 * c;  
    }
```

```
private void newIntens() {  
    startActivity(new Intent( this,InputActivity.class ));  
    finish();  
}
```

```
}
```

```
}
```



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : Fazrul Rafsanjani
NIM : 09.12.515
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer
MASA BIMBINGAN: SEMESTER GENAP 2012/2013
JUDUL : *APLIKASI PEREKAMAN KECEPATAN DAN RUTE ANGKUTAN
KOTA BERBASIS SMARTPHONE ANDROID*

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa
Tanggal : 20 Agustus 2013
Dengan Nilai : 76,75 (B+) \checkmark

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

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

Sekretaris Majelis Penguji

Dr. Eng. Arvanto S, ST, MT
NIP.P.1030800417

ANGGOTA PENGUJI

Dosen Penguji I

Ir. Eko Nurcahyo, MT
NIP.P.10128700172

Dosen Penguji II

Yuli Wahyuni ST, MT
NIP.P. 1031200456



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : FAZRUL RAFSANJANI
NIM : 09.12.515
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer
MASA BIMBINGAN: SEMESTER GENAP 2012/2013
JUDUL : APLIKASI PEREKAMAN KECEPATAN DAN RUTE
ANGKUTAN KOTA BERBASIS SMARTPHONE ANDROID

| Tanggal | Uraian | Paraf |
|--------------------------------|---|-------|
| Pengujian I 20- 08 - 2013 | 1. Percobaan untuk jarak dan kecepatan 2. Menentukan nilai error untuk jarak dan kecepatan | |
| Pengujian II 20 - 08 - 2012 | 1. Abstrak 2. Daftar Pustaka 3. Pengujian keakurasian 4. kesimpulan 5. Keterangan tabel 6. Keterangan jarak pada tabel | |

Disetujui,

Dosen Pengujian I

Ir. Eko Nurcahyo, MT
NIP.P.10128700172

Dosen Pengujian II

Yuli Wahyuni ST, MT
NIP.P. 1031200456

Mengetahui,

Dosen Pembimbing-I

Dr. Eng. Arvyanto S, ST, MT
NIP.P.1030800417

Dosen Pembimbing II

Irmalia Suryani Faradisa, ST, MT
NIP.P.103000365



FORMULIR BIMBINGAN SKRIPSI

Nama : FAZRUL RAFSANJANI
Nim : 09.12.515
Masa Bimbingan : Maret 2013 s/d Agustus 2013
Judul Skripsi : " APLIKASI PEREKAMAN KECEPATAN DAN RUTE ANGKUTAN KOTA
BERBASIS SMARTPHONE ANDROID "

| No. | Tanggal | Uraian | Parap Pembimbing |
|-----|---------|---|------------------|
| 1. | | Arc bab I dan II, "178" perbaiki spasi | |
| 2. | | acc. bab III dan IV | |
| 3. | | Acc bab IV | |
| 4. | | Perbaiki bab V (kesimpulan) | |
| 5. | | Acc bab V | |
| 6. | | | |
| 7. | | | |
| 8. | | | |
| 9. | | | |
| 10. | | | |

Malang, Juni 2013
Dosen Pembimbing,

(Dr. Eng. Aryanto Soetedjo, ST, MT)
NIP.P.1030800417

Form.S-4b



FORMULIR BIMBINGAN SKRIPSI

Nama : FAZRUL RAFSANJANI
Nim : 09.12.515
Masa Bimbingan : Maret 2013 s/d Agustus 2013
Judul Skripsi : " APLIKASI PEREKAMAN KECEPATAN DAN RUTE ANGKUTAN KOTA
BERBASIS SMARTPHONE ANDROID "

| No. | Tanggal | Uraian | Parap Pembimbing |
|-----|---------|--|------------------|
| 1. | | Acc bab I, II, revisi flowchart dan rumus bab II | |
| 2. | | Acc bab III | |
| 3. | | Revisi bab IV | |
| 4. | | revisi data tabel dan pengujian | |
| 5. | | revisi pengujian | |
| 6. | | Acc bab IV | |
| 7. | | revisi kesimpulan bab V | |
| 8. | | acc bab V | |
| 9. | | | |
| 10. | | | |

Malang, Juni 2013
Dosen Pembimbing,

(Irmalia Suryani Faradisa, ST, MT)
NIP.P. 103000365

Form.S-4b



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

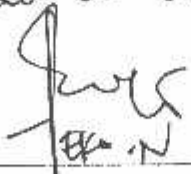
Formulir Perbaikan Ujian Skripsi

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 : FARROL R
NIM : 0912515
Perbaikan meliputi :

- Lakukan perbaikan secara nyata utli jarak dan kecepatan
- Menentukan sendiri, utli jarak & kecepatan

Malang, 20-08-2013

()



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

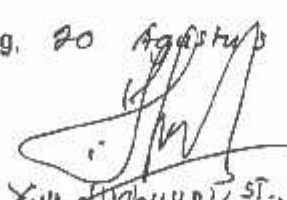
Formulir Perbaikan Ujian Skripsi

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 : *Fazrul Rapsanjani*
NIM : *09.12.515*
Perbaikan meliputi :

- ① *Abstrak.*
- ② *Daftar pustaka.*
- ③ *pengujian keakuratan.*
- ④ *kesimpulan harus berdasarkan hasil pengujian.*
- ✓ ⑤ *keterangan tabel tidak ada.*
- ✓ ⑥ *Tambahan keterangan arah pada tabel*

Malang, 20 Agustus 2013


(*Yuli Wahyuni, ST., MT.*)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Big thank's to ALLAH S.W.T

Skripsi ini khusus goe persembahkan buat abah dan mama, ini hanya hal kecil yang baru bisa goe persembahkan buat menghibur hati kalian yang telah goe susahkan. Yang goe tau banyak sekali hal yang telah kalian korbankan untuk memenuhi kebutuhan goe selama kuliah. Kalian sungguh orang tua yang hebat, sungguh orang tua yang super . Untuk my beloved brother and sister ka jaiz, ka vai, bampi, terima kasih untuk dukungan, bantuan, serta kepercayaannya selama ini. Terima kasih untuk telah menjadi orang tua dan saudara yang hebat, terima kasih untuk telah menjadi keluarga yang dapat di banggakan.

Buat saudara-saudara sepupu yang juga banyak membantu selama kuliah, terima kasih banyak juga. Untuk Ambon, Aii, tengkyu beroh 4 tahun bersama, kalian juga super sekali, NTT butuh kita. Ahahahaha..... Jefri, om gilang, tengkyu juga buat perjuangan bersama dalam menyelesaikan skripsi ini, hiduppp mas Putra. Untuk arek-arek dinoyo gang 17, “*kuliah hera sudah, gelaha baing, inga ortu be'u doi hopang we*”. Untuk Tete, “*kelea udah ade, glaha cukup sudah*”. Untuk onen, “*moe paling oke, klo bisa gelaha we cukup ade*”. Tengkyu juga buat The Crocodile United, pisah dari kalian, berarti skil olah bola pun berkurang. Untuk “rakaters” ITN 2, terima kasih untuk kebersamaan selama ini.

Hadist nabi: *“Pelajarilah ilmu karena sesungguhnya belajar semata-mata bagi Allah itu merupakan kebaikan, dan mempelajari ilmu merupakan tasbih, dan membahasnya merupakan jihad, dan mencarinya merupakan ibadah, dan mengajarkannya merupakan sedekah sedangkan menggunakannya bagi orang yang membutuhkannya merupakan Qurbah(pedekatan diri kepada allah)”*.
