

# Analisis dan Perancangan Prototipe Aplikasi Tracking Bis Universitas Multimedia Nusantara pada Platform Android

Edgar Winata, Johan Setiawan

Program Studi Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia  
edgar.winata@gmail.com, johansetiawanumn@gmail.com

Diterima 1 Mei 2013

Disetujui 31 Mei 2013

**Abstrak**—Penelitian ini bertujuan untuk melakukan analisis dan perancangan prototipe aplikasi *tracking* bis Universitas Multimedia Nusantara pada *platform* Android. Perancangan prototipe aplikasi ini menggunakan beberapa diagram Unified Modeling Language (UML) antara lain use case diagram, class diagram, sequence diagram, dan activity diagram untuk menggambarkan aplikasi yang akan dibuat. Aplikasi yang dibuat dirancang untuk memudahkan pihak universitas atau mahasiswa pengguna bis kampus untuk melihat informasi bis kampus antara lain rute bis kampus, posisi bis kampus, dan jadwal bis kampus.

**Kata kunci**—Android, tracking kendaraan

## I. PENDAHULUAN

### A. Latar Belakang

Universitas Multimedia Nusantara (UMN), sebagai salah satu perguruan tinggi swasta yang sedang berkembang dan memiliki banyak mahasiswa berusaha memenuhi kebutuhan mahasiswa baik dalam sarana dan prasarana yang dapat menunjang kegiatan perkuliahan. Salah satu hal yang dilakukan oleh pihak universitas adalah mengadakan bis universitas yang dapat membantu mahasiswa dalam hal transportasi menuju universitas.

Bis Universitas Multimedia Nusantara mulai beroperasi sejak tanggal 20 Februari 2012. Bis Universitas Multimedia Nusantara ini ada sebagai sarana dan prasarana untuk membantu mahasiswa dalam hal transportasi menuju universitas. Rute dari bis kampus ini adalah Pasar Modern BSD menuju kampus UMN melalui de'Latinos dan RS Assobirin. Beberapa titik keberangkatan dari bis kampus ini adalah Pasar Modern BSD, de'Latinos BSD, Halte BJ Home BSD, dan Kampus UMN. Bis kampus ini mulai berangkat sejak pukul 06.00 dari titik keberangkatan pertama.

Bis kampus ini diadakan dengan melihat kebutuhan mahasiswa akan transportasi yang aman

dan nyaman dalam perjalanan menuju kampus. Penjadwalan keberangkatan bis dilakukan dengan baik agar mahasiswa tidak terlambat dalam mengikuti perkuliahan. Akan tetapi situasi daerah Serpong dan sekitarnya yang padat penduduk menjadi masalah tersendiri dalam kegiatan operasional bis kampus. Kemacetan menjadi suatu kendala sehingga bis tidak dapat mencapai titik keberangkatan dengan tepat waktu sesuai dengan jadwal yang sudah dibuat.

Berdasarkan keadaan di atas, perlu dilakukan hal yang dapat menanggulangi masalah tersebut. Sehingga dibutuhkan suatu alat yang dapat membantu mahasiswa melakukan *tracking* untuk mengetahui informasi dan posisi bis.

### B. Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat sebuah prototipe aplikasi *tracking* terhadap bis kampus pada platform Android agar posisi bis dapat dilihat didalam sebuah map oleh pihak universitas dan mahasiswa yang menggunakan bis kampus sebagai sarana transportasi menuju dan dari universitas

### C. Metode Penelitian

Metode penelitian yang dipergunakan dalam penelitian ini meliputi: observasi, perancangan system menggunakan diagram-diagram UML, perancangan user interface menggunakan pedoman dan prinsip perancangan dari Ben Schneiderman, sedangkan untuk pemrograman menggunakan Java, Android SDK dan Tools Eclipse.

## II. TINJAUAN PUSTAKA

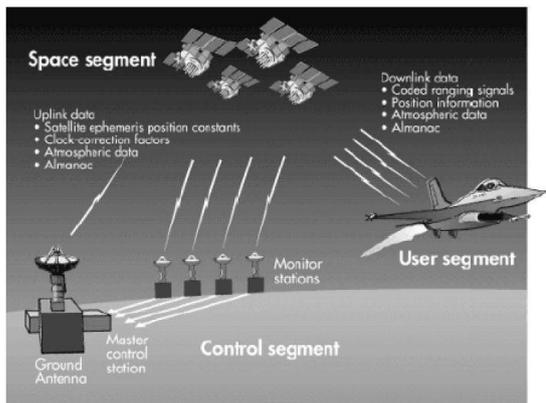
### A. Global Positioning System (GPS)

Masalah navigasi dan posisi letak sangatlah penting dalam berbagai aktifitas misalnya di bidang penerbangan dan kelautan, dan beberapa tahun ini berbagai jenis teknologi telah mencoba untuk memberikan solusi terbaik namun masih terdapat

beberapa kelemahan.

Global Positioning System merupakan sistem navigasi radio yang terbentuk dari gugusan 24 satelit beserta ground station. Global Positioning System menggunakan “*man-made stars*” sebagai acuan untuk memperhitungkan akurasi posisi dengan satuan meter. Bahkan dengan *Global Positioning System* yang berteknologi lebih maju, kita dapat mengukur akurasi kurang dari satu sentimeter (Trimble – All About GPS).

Saat ini, ukuran GPS receiver yang hanya terdiri dari beberapa integrated circuit terbilang kecil dan sangat ekonomis sehingga memudahkan bagi setiap orang untuk membawanya. GPS dapat digunakan pada mobil, kapal, pesawat, komputer atau laptop bahkan sekarang ini GPS merupakan peralatan yang sudah digunakan pada telepon-telepon selular seperti ponsel pintar atau Smartphone.



Gambar 1 Segmen-segmen Sistem GPS (McNamara, 2004, p51)

GPS tersusun atas satelit, stasiun pemancar di bumi, dan alat penerima sinyal. Dalam istilah GPS, satelit termasuk didalam segmen luar angkasa (space). Kumpulan dari 24 satelit GPS (21 operasional dan 3 cadangan) yang mengitari bumi pada ketinggian 12.000 mil diatas permukaan bumi. Satelit mengitari bumi dengan kecepatan 7.000 mil per jam. Sehingga membutuhkan waktu 12 jam bagi satelit untuk dapat mengitari bumi secara keseluruhan, melewati titik yang sama kurang lebih setiap 24 jam. Satelit diposisikan ketika alat penerima sinyal GPS (GPS Receiver) dapat menerima sinyal sekurang-kurangnya dari 6 satelit di seluruh lokasi di Bumi.

Lima langkah kerja GPS adalah sebagai berikut:

1. GPS melakukan triangulasi dari satelit-satelit. Triangulasi merupakan suatu proses mencari koordinat dari sebuah titik dengan cara menghitung panjang sisi segitiga yang

berhadapan dengan titik tersebut.

2. Untuk melakukan triangulasi, GPS receiver mengukur jarak menggunakan waktu tempuh sinyal radio.
3. Untuk mengukur waktu tempuh, GPS membutuhkan perhitungan waktu yang sangat akurat yang bisa didapat dengan cara-cara tertentu.
4. Sepanjang perhitungan jarak, kita harus mengetahui dimana posisi satelit berada di angkasa.
5. Terakhir, kita harus melakukan *error correction* terhadap penundaan sinyal yang terjadi ketika melalui atmosfer.

#### B. Telepon Seluler

Telepon seluler merupakan sebuah perangkat telekomunikasi yang mempunyai kemampuan sama dengan telepon konvensional tetapi dapat dibawa kemana-mana (mobile dan wireless). Telepon seluler bukanlah merupakan hal yang asing bagi masyarakat sekarang ini. Telepon seluler sudah menjadi suatu keharusan bagi masyarakat yang tinggal di kota-kota sebagai pendukung kebutuhan mereka.

Telepon seluler yang memiliki kemampuan menyerupai komputer atau biasa disebut dengan ponsel pintar (*Smartphone*) ini juga mulai banyak digunakan oleh masyarakat. Baik digunakan untuk kebutuhan bisnis maupun hiburan seperti chatting, menonton video, menjelajah situs web, membaca berita terbaru melalui internet, mengakses jejaring sosial, dan berbagai hal lain yang dapat dilakukan oleh ponsel pintar tersebut. Dengan banyaknya fitur yang dapat dilakukan oleh ponsel pintar tersebut. Keberadaan ponsel pintar atau smartphone ini dapat membantu pengguna dalam memperoleh informasi yang dibutuhkan dimanapun mereka berada.

#### C. Android

Android merupakan sistem operasi berbasis Linux yang dapat digunakan pada ponsel pintar dan komputer tablet. Dalam mengembangkan Android pertama kali dibentuklah Open Handset Alliance yang terdiri dari 34 perusahaan hardware, software, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada tahun 2008 Android mulai diperkenalkan kepada masyarakat dunia dan sekarang Android banyak digunakan sebagai sistem operasi ponsel pintar, seperti Samsung, Motorola, LG, HTC, dan Sony Ericson. Bahkan telepon seluler buatan lokal maupun Cina mulai banyak menggunakan sistem operasi Android sehingga ponsel pintar dengan sistem

operasi Android ini tersedia dalam berbagai rentang harga, mulai dari satu hingga diatas lima juta rupiah. Hal ini memungkinkan telepon selular dengan sistem operasi Android makin banyak digunakan oleh semua orang.

Berdasarkan Canals Research pada tahun 2011 dalam artikel “Android at 48% Market Share in Q2”, Android telah menguasai pangsa pasar (market share) dunia sebesar 48% di kuartal ke-2 tahun 2011, sementara Blackberry, IOS, dan Windows Mobile sebagai pesaingnya masing-masing memiliki pangsa pasar sebesar 32%, 19%, dan 1% (McLaughlin, 2011).

Hal menarik lain dari ponsel pintar dengan sistem operasi Android selain didukung dengan banyak fitur yang dapat dilakukan oleh layaknya sebuah ponsel pintar adalah Android merupakan sistem operasi yang menyediakan platform terbuka bagi para pengembang sehingga pengembang dapat menciptakan aplikasi yang dapat digunakan di sebuah ponsel pintar dengan sistem operasi Android.

Gambar 2 dibawah adalah penjelasan arsitektur komponen utama dari sistem operasi Android.



Gambar 2 Arsitektur Android

- **Linux Kernel**, merupakan tumpukan paling bawah pada arsitektur Android. Google menggunakan kernel Linux versi 2.6 untuk membangun sistem Android. Kernel berperan sebagai abstraction layer antara hardware dan keseluruhan software.
- **Android Runtime**, pada Android Runtime berisi Core Libraries dan Dalvik Virtual Machine. Core Libraries mencakup serangkaian inti library Java. Dalvik Virtual Machine berfungsi memberikan kekuatan dan

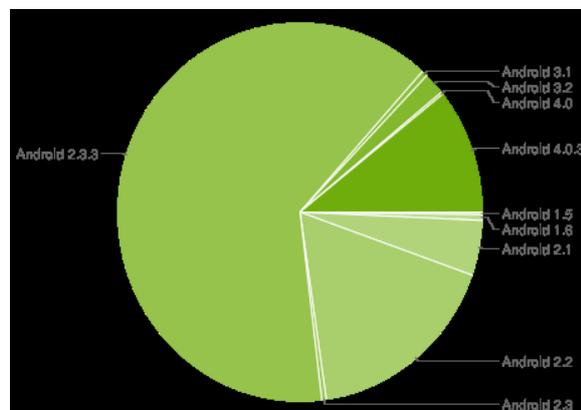
mengoptimalkan sistem operasi Android.

- **Libraries**, memiliki tempat yang sama dengan Android Runtime. Tempat dimana menyimpan set-set library dalam bahasa C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem operasi Android.
- **Application Framework**, pada lapisan ini mencakup program untuk mengatur fungsi-fungsi dasar pada smartphone. Application Framework merupakan serangkaian tools dasar seperti alokasi resource smartphone, aplikasi telepon, pergantian antar proses atau program, dan pelacakan lokasi fisik telepon.
- **Applications**, pada lapisan ini dapat ditemukan fungsi-fungsi dasar pada smartphone seperti menelepon, mengirim pesan singkat, menjalankan web browser, melakukan akses pada kontak, dan lain-lain.

Gambar 3 adalah diagram yang menunjukkan penggunaan platform Android berdasarkan akses yang dilakukan pada Google Play. Data dikumpulkan oleh Google selama 14 hari, berakhir pada tanggal 2 Juli 2012 (Android, 2012).

#### D. Google Map

Google Maps merupakan sebuah peta dan web direction dari Google. Google Maps menyediakan arah dan tujuan atau dapat diisi oleh pengguna sehingga dapat menampilkan rute jalan pada map. Selain itu Google Maps mendukung tampilan 2D dan 3D dari bumi dengan bantuan satelit. Google Maps juga dapat menampilkan pemandangan fotografi bumi, memperlihatkan jalan dan sekitarnya secara nyata.



Gambar 3 Penggunaan Platform Android berdasarkan Google Play

Version	Codename	API Level	Distribution
1.5	Cupcake	3	0.2%
1.6	Donut	4	0.5%
2.1	Eclair	7	4.7%
2.2	Froyo	8	17.3%
2.3 - 2.3.2	Gingerbread	9	0.4%
2.3.3 - 2.3.7		10	63.6%
3.1	Honeycomb	12	0.5%
3.2		13	1.9%
4.0 - 4.0.2	Ice Cream Sandwich	14	0.2%
4.0.3 - 4.0.4		15	10.7%

Gambar 4 Penggunaan Platform Android berdasarkan Google Play (dalam persen)

Berikut ini adalah beberapa fitur Google Maps:

- Membuat aplikasi berbasis lokasi
- Dapat digunakan pada mobile application
- Dapat melakukan visualisasi data geospasial
- Dapat dikustomisasi melalui Google Maps API yang telah diberikan oleh Google (misalnya dapat memberikan pin point dan membuat rute jalan)

#### E. Internet

Menurut Greenlaw dan Hepp (2002, p99), Internet menunjukkan sistem informasi global yang:

- Secara logika terhubung melalui alamat global yang unik berdasarkan Internet Protocol (IP).
- Memungkinkan untuk melakukan komunikasi menggunakan TCP/IP.
- Menyediakan layanan komunikasi tingkat tinggi.

Pengertian dari Internet dapat disederhanakan menjadi: sistem jaringan komputer global yang terbentuk dari komputer-komputer yang saling terhubung, beserta pengguna dan datanya.

Secara umum ada banyak manfaat yang dapat diperoleh apabila seseorang mempunyai akses ke Internet. Berikut ini sebagian dari apa yang tersedia di Internet:

- Informasi untuk kehidupan pribadi: kesehatan, rekreasi, hobby, pengembangan pribadi, rohani, dan sosial.
- Informasi untuk kehidupan profesional atau pekerja: sains, teknologi, perdagangan, saham, komoditas, berita bisnis, asosiasi profesi, asosiasi bisnis, berbagai forum komunikasi.

Satu hal yang paling menarik ialah keanggotaan Internet tidak mengenal batas Negara, ras, kelas ekonomi, ideology, maupun faktor-faktor lain yang biasanya dapat menghambat pertukaran pikiran. Internet adalah suatu komoditas dunia yang sifatnya sangat demokratis serta memiliki kode etik yang dihormati segenap anggotanya. Manfaat Internet terutama diperoleh melalui kerjasama antar pribadi atau kelompok tanpa mengenal batas jarak dan waktu. Untuk lebih meningkatkan kualitas sumber daya manusia, sudah waktunya untuk para profesional memanfaatkan jaringan Internet dan menjadi bagian dari masyarakat informasi dunia.

#### F. Unified Modeling Language (UML)

Unified Modeling Language adalah sebuah bahasa yang diterima dan digunakan oleh software developer dan software analyst sebagai suatu bahasa yang cocok untuk merepresentasikan grafik dari suatu relasi antar entitas-entitas software (Gornik, 2003). Dengan menggunakan UML, tim pengembang software akan mempunyai banyak keuntungan, seperti memudahkan komunikasi dengan sesama anggota tim tentang software apa yang akan dibuat, memudahkan integrasi ke dalam area pengerjaan software karena bahasa ini berbasis meta-models dimana meta-models bisa mendefinisikan proses-proses untuk mengkonstruksikan konsep-konsep yang ada. UML juga menggunakan format input dan output yang sudah mempunyai bentuk standar yaitu XML Metadata Interchange (XMI), menggunakan aplikasi dan pemodelan data yang universal, merepresentasikan dari tahap analisis ke implementasi lalu ke *deployment* yang terpadu, dan mendeskripsikan keutuhan tentang spesifikasi software (Hofmeister, Nord, dan Soni, 1999).

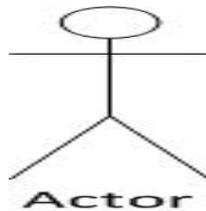
UML menyediakan kumpulan alat yang sudah terstandarisasi, yang digunakan untuk mendokumentasikan analisis dan perancangan sebuah sistem perangkat lunak. (Kendall & Kendall, 2005, p663) Peralatan utama UML adalah diagram-diagram yang digunakan untuk membantu manusia dalam memvisualisasikan proses pengembangan sebuah sistem perangkat lunak, sama seperti penggunaan denah (blueprint) dalam pembuatan bangunan.

### G. Use Case Model

Use case model adalah teknik pemodelan untuk mendapatkan *functional requirement* dari sebuah sistem, menggambarkan interaksi antara pengguna dan sistem, menjelaskan secara naratif bagaimana sistem akan digunakan, menggunakan skenario untuk menjelaskan setiap aktivitas yang mungkin terjadi. Ada beberapa bagian didalam use case model (Dharwiyanti dan Wahono, 2003).



Gambar 5 Use Case pada Use Case Diagram



Gambar 6 Actor pada Use Case Diagram

- Use Case*, untuk mengetahui action atau prosedur apa yang ada didalam sistem.
- Actor*, siapa saja yang terlibat dalam action tersebut.
- Relationship*, bagaimana actions saling berelasi satu sama lain didalam sistem.

### H. Class Diagram

Class diagram merupakan diagram paling umum yang dijumpai dalam pemodelan berbasis UML. Didalam Class diagram terdapat class dan interface beserta atribut-atribut dan operasinya, relasi yang terjadi antar objek, *constraint* terhadap objek-objek yang saling berhubungan dan *inheritance* untuk organisasi class yang lebih baik. Class diagram juga terdapat *static view* dari elemen pembangun sistem. Pada intinya Class diagram mampu membantu proses pembuatan sistem dengan memanfaatkan konsep *forward* ataupun *reverse engineering* (Rational Software Corporation, 1997).

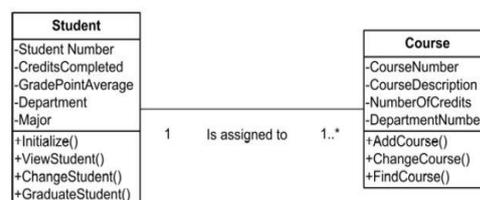
Class diagram mempunyai 2 komponen penting, yaitu:

- Structural*, yaitu ciri pembeda objek.
- Behavioral*, yaitu tingkah laku atau kegiatan yang mampu dilakukan oleh objek.

Berbagai simbol yang hadir didalam class diagram

antara lain adalah:

- Class*, yang berfungsi untuk merepresentasikan tipe dari data yang dimilikinya. Class diagram dapat ditampilkan dengan menunjukkan atribut dan operasi yang dimilikinya atau hanya menunjukkan nama class-nya saja. Dapat juga kita tuliskan nama class dengan atributnya saja atau nama class dengan operasinya.
- Attribute*, merupakan data yang terdapat didalam class dan instance-nya dengan operator.
- Operation*, berfungsi untuk merepresentasikan fungsi-fungsi yang ditampilkan oleh class dan instance-nya dengan operator.
- Association*, digunakan untuk menunjukkan bagaimana dua class berhubungan satu sama lainnya. Association ditunjukkan dengan sebuah garis yang terletak diantara dua class. Didalam setiap association terdapat multiplicity, yaitu simbol yang mengindikasikan berapa banyak instance dari class pada ujung association yang satu dengan instance class di ujung association lainnya.
- Generalizations*, berfungsi untuk mengelompokkan class ke dalam hirarki inheritance.
- Aggregation*, merupakan bentuk khusus dari association yang merepresentasikan hubungan “part-whole”. Bagian “whole” dari hubungan ini sering disebut dengan assembly atau aggregate. Class yang satu dapat dikatakan merupakan bagian dari class yang lain yang ikut membentuk class tersebut.
- Composition*, merupakan jenis aggregation yang lebih kuat diantara dua class yang memiliki association dimana jika *whole* ditiadakan, maka *part*-nya juga ikut ditiadakan. Berbeda dengan aggregation, *part* akan tetap bisa berdiri sendiri meskipun bagian *whole*-nya ditiadakan.
- Penggunaan operator (+) dalam class diagram diartikan dengan public, operator (-) diartikan private, dan operator (#) diartikan protected.



Gambar 7 Contoh Class Diagram (Kendall & Kendall, 2005, p685)

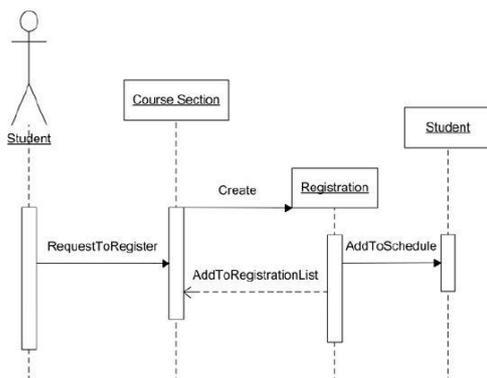
Class diagram menggambarkan karakteristik statis sistem tanpa menjelaskan proses secara mendetil. Sebuah class diagram juga memperlihatkan hubungan antar class. (Kendall & Kendall, 2005, p678)

### I. Sequence Diagram

Menjelaskan interaksi obyek-obyek yang saling berkolaborasi (berhubungan), mirip dengan *activity diagram* yaitu menggambarkan alur kejadian sebuah aktivitas tetapi lebih detil dalam menggambarkan aliran data termasuk data atau behaviour yang dikirimkan atau diterima namun kurang mampu menjelaskan detil dari sebuah algoritma (Hofmeister, Nord, & Soni, 1999).

Dalam *sequence diagram* terdapat beberapa bagian.

- *Participant*, yaitu objek yang terkait dengan sebuah urutan proses.
- *Lifeline*, menggambarkan daur hidup sebuah objek.
- *Activation*, suatu titik waktu dimana sebuah objek mulai berpartisipasi dalam sebuah sequence.
- *Time*, elemen paling penting dalam sequence diagram yang konteksnya adalah urutan, bukan durasi.
- *Return*, suatu hasil kembalian sebuah operasi. Operasi mengembalikan hasil tetapi boleh tidak ditulis jika tidak ada perbedaan dengan Getternya.



Gambar 8 Contoh Sequence Diagram (Kendall & Kendall, 2005, p682)

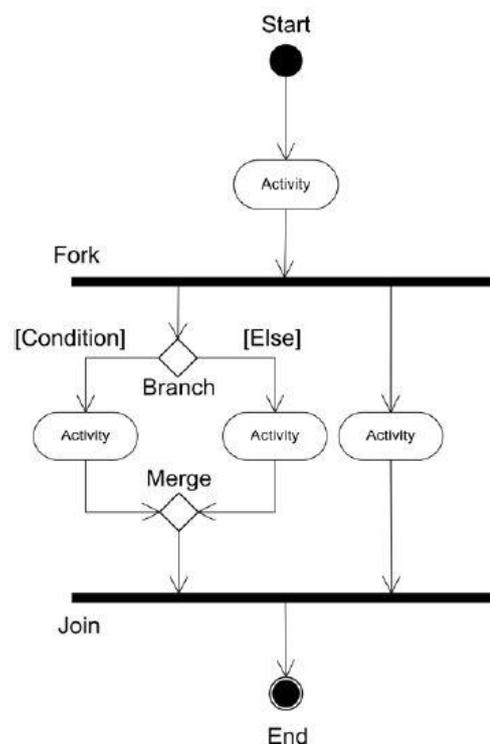
Dalam penggunaannya, *sequence diagram* tepat untuk memperlihatkan dengan jelas bagaimana urutan kejadian suatu proses karena didalamnya terlihat interaksi beberapa objek.

### J. Activity Diagram

Teknik untuk menjelaskan *business process*,

*procedural logic*, dan *work flow*. Bisa dipakai untuk menjelaskan teks use case dalam notasi grafis dengan menggunakan notasi yang mirip *flow chart*, meskipun terdapat sedikit perbedaan notasi (Hofmeister, Nord, & Soni, 1999).

- *Nodes*, menandakan initial dan final node, final node boleh lebih dari 1.
- *Activity*, aktivitas sistem dapat berupa aktivitas fisik juga bagi user.
- *Flow/edge*, arah sebuah proses.
- *Fork*, awal sebuah proses paralel.
- *Join* akhir proses paralel.
- *Condition*, kondisi yang dituliskan dalam bentuk teks
- *Decision*, implementasi if dan then.
- *Merge*, penyatuan beberapa flow.
- *Partition*, siapa atau apa yang menjalankan aktivitas.



Gambar 9 Contoh Activity Diagram (Kendall & Kendall, 2005, p672)

### K. Desain User Interface

#### 1. Pedoman Perancangan User Interface

Pada era komputerisasi, perancang interface telah menuliskan pedoman untuk mencatat pengetahuannya dan mencoba untuk memandu dalam upaya untuk perancangan-perancangan interface selanjutnya. Pedoman dari Apple dan Microsoft yang mempengaruhi perancang *desktop-interface*, telah diikuti oleh banyak dokumen yang berisi tentang pedoman perancangan *user interface*.

Namun banyak kritikus yang melakukan protes karena pedoman tersebut dapat menjadi terlalu spesifik, tidak lengkap, sulit untuk diterapkan dan kadang-kadang terdapat kesalahan. Argumen pendukung yang dibangun berdasarkan pengalaman dari perancang ahli berkontribusi untuk peningkatan kualitas interface (Shneiderman, 2009, hal 57).

## 2. Organisasi Tampilan

Perancangan tampilan adalah topik besar dengan banyak kasus. Smith dan Mosier pada tahun 1986 (Shneiderman, 2009, hal 58) memberikan lima sasaran sebagai pedoman mereka tentang tampilan data, yaitu:

- 1) Konsistensi tampilan data. Dalam proses desain, terminologi, singkatan, format, warna, kapitalisasi dan lainnya harus di standarisasi.
- 2) Perpaduan informasi yang efisien oleh pengguna. Formatnya harus serupa dengan operator dan harus berhubungan tugas yang diperlukan dalam menampilkan data atau informasi.
- 3) Minimalisasi beban ingatan pengguna. Pengguna tidak harus mengingat informasi dari satu tampilan untuk digunakan pada tampilan lainnya.
- 4) Kompatibilitas tampilan data dengan pemasukan data. Format tampilan data atau informasi harus terhubung dengan jelas pada format untuk memasukkan data.
- 5) Fleksibilitas kendali oleh pengguna dari tampilan data. Pengguna harus dapat mengambil informasi dari form yang telah ditampilkan secara mudah.

## 3. Mendapatkan Perhatian Pengguna

Shneiderman (2009, hal 60) mengungkapkan bahwa informasi *subtansial* akan ditampilkan pengguna untuk pekerja yang biasa dilakukannya, kondisi pengecualian atau informasi yang bergantung pada waktu harus ditampilkan untuk mendapatkan perhatian pengguna:

- 1) Intensitas. Menggunakan hanya 2 tingkatan saja. Menggunakan intensitas yang lebih

tinggi untuk menarik perhatian.

- 2) Penandaan. Memberi garis bawah objek, menunjuk objek dengan tanda panah atau menggunakan indicator seperti asterisk, dash, bulatan dan sebagainya.
- 3) Skala. Gunakan sampai dengan 4 skala, dengan ukuran skala yang besar untuk lebih banyak menarik perhatian.
- 4) Pemilihan font. Menggunakan sampai dengan 3 jenis font. Misalnya Times New Roman, Arial, dan Calibri.
- 5) Inverse video. Menggunakan pewarnaan yang kontras satu dengan yang lain.
- 6) Kelap-kelip. Menampilkan teks, gambar atau warna-warna yang kelap-kelip dengan hati-hati dan dalam area tertentu.
- 7) Warna. Menggunakan sampai dengan 4 warna standar, dengan tambahan warna yang dipersiapkan untuk pemakaian tambahan. Misalnya warna hitam, merah, biru, dan kuning.
- 8) Suara. Menggunakan bunyi yang lembut untuk timbal balik yang positif dan suara yang parau atau lebih kasar untuk kondisi-kondisi kesalahan tertentu yang cukup fatal.

## 4. Memfasilitasi Entri Data

Tugas entri data dapat menyita waktu pengguna cukup lama dan dapat menjadi sumber frustrasi dan memungkinkan terjadinya kesalahan yang berbahaya. Smith dan Mosier pada tahun 1986 (Shneiderman, 2009, hal 61) memberikan 5 objektif sebagai bagian dari pedoman mereka tentang entri data:

- 1) Konsistensi transaksi pemasukan data
- 2) Minimalisasi aksi pemasukan dari pengguna
- 3) Minimalisasi beban ingatan pengguna
- 4) Kompatibilitas input data dengan tampilan data
- 5) Fleksibilitas kendali

## 5. Prinsip Perancangan User Interface

Shneiderman (2009, hal 62) juga berpendapat, disaat pedoman difokuskan semakin sempit, prinsip-prinsip cenderung untuk lebih mendasar, diterapkan secara luas dan terus-menerus. Contohnya adalah menetapkan tingkat kemampuan pengguna. Mempelajari tentang pengguna adalah ide yang sederhana tetapi sulit dan sayangnya biasanya tujuannya dianggap remeh. Tidak seorangpun akan

membantah tentang prinsip ini, namun banyak perancang dengan mudah mengasumsikan bahwa mereka mengerti para pengguna dan juga tugas dari pengguna tersebut. Shneiderman (2009, hal 63-64) menyarankan pemisahan tingkat kemampuan pengguna dibagi berdasarkan perbedaan tujuan akhir perancangan, yaitu:

- 1) *Novice* atau pengguna biasa.
- 2) *Knowledge intermittent users* atau pengguna yang memiliki cukup pengetahuan.
- 3) *Expert frequent users* atau pengguna ahli.

## 6. Bahasa Pemrograman Java

Java merupakan bahasa pemrograman yang digunakan untuk kebutuhan umum seperti sistem pakar, aplikasi basis data, *mobile application*, dan lainnya (Gosling, dkk. 2005).

Bahasa Java ini bersifat *concurrent*, berbasis kelas dan berorientasi objek. Java dirancang cukup sederhana sehingga programmer dapat dengan cepat terbiasa menggunakan bahasa tersebut. Bahasa pemrograman Java ini memiliki relasi dengan bahasa pemrograman C dan C++ tetapi diorganisasikan cukup berbeda, dengan beberapa aspek dari C dan C++ yang dihilangkan dan mengikutsertakan beberapa ide dari bahasa pemrograman lain.

Bahasa pemrograman Java adalah bahasa yang relatif tingkat tinggi dengan perincian dan representasi bahasa mesin tidak melalui Java. Itu termasuk *Automatic Storage Management* yang biasanya menggunakan *garbage collector* untuk menghindari masalah keamanan dari pembersihan memori secara eksplisit yang dilakukan oleh bahasa pemrograman C dan C++. Implementasi pengumpulan sampah dengan kinerja tinggi dapat dibatasi untuk mendukung pemrograman sistem dan aplikasi real time.

Java merupakan bahasa pemrograman yang digunakan dalam membuat dan mengembangkan aplikasi pada platform Android. Java merupakan bahasa pemrograman berorientasi obyek, dan dirancang agar mudah untuk dipakai dan digunakan pada platform independent sehingga dapat dijalankan di berbagai jenis sistem operasi dan arsitektur komputer.

## III. PEMBAHASAN

### A. Gambaran Umum Sistem

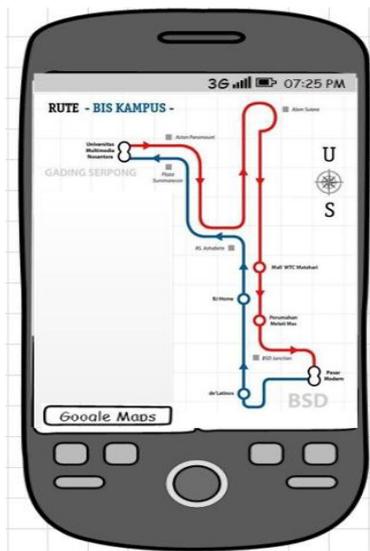
Prototipe atau sistem yang akan dirancang diperuntukkan bagi mahasiswa Universitas Multimedia Nusantara pengguna jasa transportasi bis kampus yang melakukan perjalanan dari rumah menuju kampus dan sebaliknya.

Sistem atau prototipe aplikasi ini dirancang pada platform Android sehingga hanya dapat digunakan oleh mahasiswa yang menggunakan *smartphone* sistem operasi Android. Sistem atau aplikasi yang dibuat ini berguna agar pengguna dapat melihat informasi dan posisi bis kampus. Beberapa informasi bis yang dapat dilihat oleh pengguna antara lain: letak posisi bis kampus, rute bis kampus, titik pemberhentian bis kampus, dan jadwal bis kampus.

Sistem atau aplikasi yang akan dirancang ini dapat memberikan informasi mengenai posisi keberadaan pengguna dan bis kampus didalam sebuah gambar atau map. Dalam melakukan hal tersebut diperlukan pengambilan data berupa string *National Marine Electronics Association (NMEA) sentence* sesuai dengan *NMEA 0183 Communication Protocol*. Namun tidak semua NMEA sentence akan digunakan pada perancangan prototipe aplikasi ini. Data NMEA sentence yang digunakan adalah data dalam derajat, yaitu *Longitude* (koordinat y atau derajat bujur Barat atau Timur) dan *Latitude* (koordinat x atau derajat lintang Utara atau Selatan).

Beberapa deskripsi dasar dan fitur yang ada pada aplikasi yang dibuat:

- Peta dasar yang akan digunakan adalah *Google Maps*. Pada Google Maps terdapat beberapa jenis tampilan peta antara lain adalah *Satelite View* dan *Street View*. Pada prototipe aplikasi yang akan dibuat akan menggunakan tampilan *Street View* seperti pada aplikasi-aplikasi map yang telah ada di Android Market sehingga pengguna juga sudah terbiasa dengan tampilan map tersebut.
- Setiap kali pengguna membuka aplikasi, secara otomatis menangkap posisi bis paling baru dari web server. *Refresh* otomatis dilakukan setiap 1 menit.
- Pada saat membuka aplikasi secara *default* akan menampilkan peta, rute bis kampus, titik pemberhentian bis kampus, dan posisi bis kampus.
- Penjadwalan Bis akan disimpan dalam database sehingga apabila terjadi perubahan jadwal tidak perlu melakukan update pada aplikasi, tetapi akan membaca data jadwal bis pada database di web server.
- Untuk mendapatkan posisi terakhir bis diperlukan sebuah aplikasi *send point* yang nantinya perlu diinstal pada sebuah handphone Android yang perlu diletakan di bis. Aplikasi tersebut berguna untuk membaca dan mengirimkan koordinat posisinya dalam format titik *Latitude* dan *Longitude*. Titik *Latitude* dan *Longitude* yang



Gambar 10 Sketsa Tampilan Halaman Utama

didapatkan akan dikirim ke database di web server. Dari database tersebut data titik *Latitude* dan *Longitude* dibaca oleh aplikasi *tracking* dan ditampilkan agar dapat melihat posisi bis terakhir.

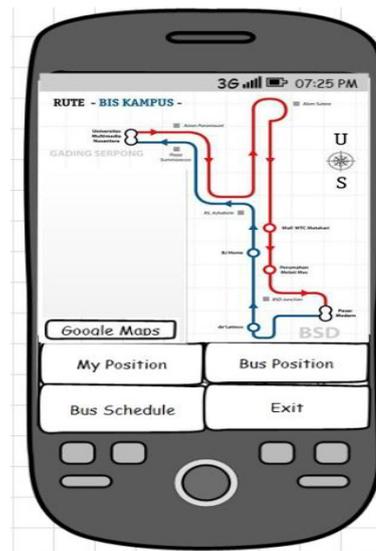
#### B. Perancangan Sketsa User Interface

Sketsa *user interface* pada aplikasi ini terbagi menjadi beberapa bagian sesuai dengan kebutuhan aplikasi. Sketsa user interface yang dirancang antara lain adalah *user interface* halaman utama, *user interface* halaman utama dengan menu panel, dan *user interface* halaman jadwal bis.

Pada halaman utama (lihat Gambar 10 diatas) akan menampilkan peta yang berdasarkan Google Maps. Pada halaman utama ini user dapat melihat rute berangkat dan pulang bis kampus, letak posisi bis, dan titik-titik pemberhentian bis kampus

Apabila user menekan tombol “menu” pada handphone Android, maka akan muncul menu panel seperti pada gambar 11 dimana terdapat beberapa menu seperti “*My Position*”, “*Bus Position*”, “*Bus Schedule*”, dan “*Exit*”. Berikut ini adalah penjelasan dari keempat menu tersebut:

- **My Position:** memperlihatkan posisi pengguna aplikasi.
- **Bus Position:** melakukan refresh terhadap posisi bis.
- **Bus Schedule:** menampilkan halaman jadwal bis.
- **Exit:** keluar dari aplikasi.



Gambar 11 Sketsa Tampilan Halaman Utama dengan Menu Panel

#### C. Sketsa Tampilan Halaman Jadwal Bis

Pada halaman jadwal bis (Gambar 12), ditampilkan jadwal keberangkatan bis dari titik-titik pemberhentian bis. Halaman jadwal bis ini dibuat dengan tabel. Agar memudahkan pergantian atau update jadwal bis, database sehingga kedepannya apabila terdapat perubahan jadwal keberangkatan bis kampus ini tidak perlu melakukan update pada aplikasi.

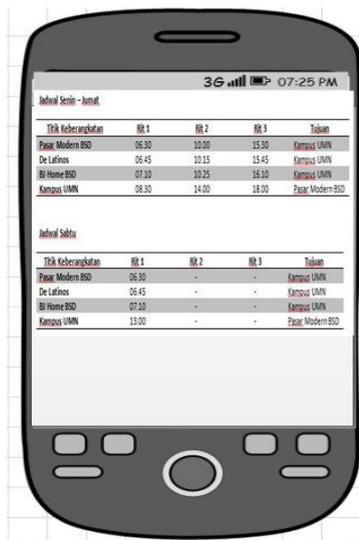
Pada halaman jadwal bis ini apabila menekan tombol “back” pada handphone Android akan kembali ke halaman utama. Apabila menekan tombol “menu” pada handphone Android tidak akan menampilkan atau melakukan action apapun.

#### D. Desain Database

Kebutuhan database pada aplikasi ini dibutuhkan untuk menyimpan berbagai informasi bis yang dibutuhkan dan jadwal bis. Tabel yang akan dibuat dalam database antara lain adalah tabel Bus, tabel Log\_Bus, tabel Jadwal, tabel Lokasi, dan tabel Titik\_Keberangkatan.

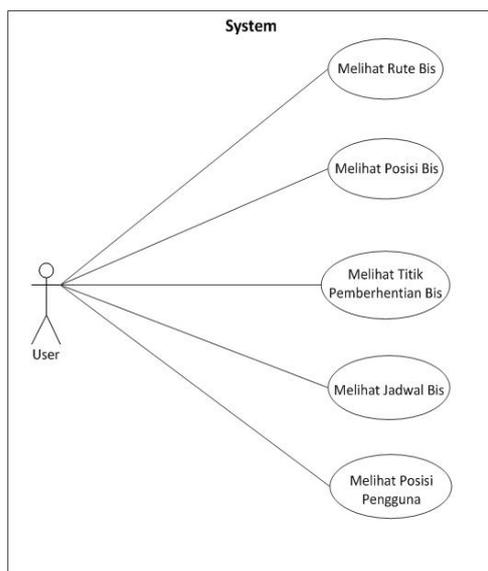
#### E. Use Case

Pada use case diagram ini terdiri dari beberapa *use case* yang dapat dilakukan oleh actor User. Gambar 13 adalah gambar use case diagram aplikasi tracking bis kampus.



Gambar 12 Sketsa Tampilan Halaman Jadwal Bus

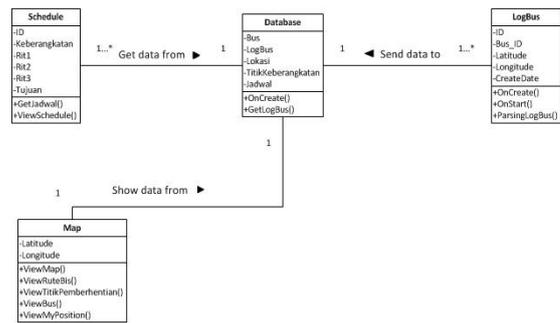
Actor User ini dapat melihat rute bis, posisi bis, titik pemberhentian bis, dan posisinya pada halaman utama. Selain itu actor User juga dapat melihat jadwal bis pada halaman jadwal bis.



Gambar 13 Use Case Diagram

F. Class Diagram

Class diagram pada aplikasi ini terdiri dari beberapa class, antara lain class Map yang menggambarkan objek map pada aplikasi, class Database menggambarkan objek database yang digunakan pada aplikasi, class LogBus menggambarkan objek yang mengirimkan posisi terakhir bis yang akan disimpan dalam database, dan Class Schedule menggambarkan objek yang digunakan untuk menyimpan dan menampilkan jadwal bis.

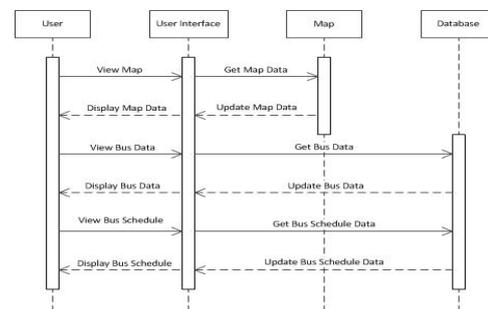


Gambar 14 Class Diagram

G. Sequence Diagram

Sequence diagram menjelaskan tentang interaksi objek-objek yang saling berhubungan. Pada sequence diagram ini menggambarkan alur kejadian sebuah aktivitas dengan menggambarkan aliran data dari aplikasi yang dibuat. Gambar 15 adalah gambar sequence diagram pada aplikasi tracking bis kampus.

Pada sequence diagram, *User* melihat melalui *User Interface*. *User Interface* meminta data *Map* melalui *Google Maps API* untuk menampilkan peta. *User* melihat data bis melalui *User Interface*, data bis diambil dari database. Data bis yang diambil antara lain rute bis, posisi bis, dan titik pemberhentian bis.



Gambar 15 Sequence Diagram

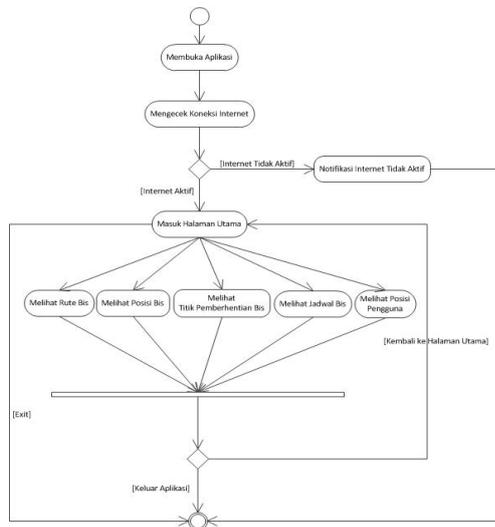
User juga dapat melihat jadwal bis melalui *User Interface* pada halaman jadwal. Pada saat User ingin melihat jadwal bis, *User Interface* halaman jadwal mengambil data jadwal bis dari database.

H. Activity Diagram

Aktivitas dimulai saat membuka aplikasi. Saat pertama kali dibuka, aplikasi akan mengecek ketersediaan koneksi internet. Apabila tidak terdapat koneksi internet akan mendapatkan notifikasi bahwa internet tidak aktif. Apabila terdapat koneksi internet maka akan masuk ke halaman utama.

Pada halaman utama terdapat beberapa *activity* yang dapat dilihat, antara lain melihat rute bis, melihat posisi bis, melihat titik pemberhentian bis, melihat jadwal bis, dan melihat posisi pengguna. Apabila

sudah melakukan *activity* tersebut dapat memilih pilihan untuk kembali ke halaman utama atau keluar dari aplikasi. Dari halaman utama dapat keluar secara langsung dari aplikasi dengan memilih pilihan Exit.



Gambar 16 Activity Diagram

#### I. Alat dan Pembangunan Aplikasi

Dalam pemrograman pada aplikasi, kode program yang dibuat menggunakan spesifikasi software sebagai berikut:

- Bahasa Pemrograman: Java, PHP
- Software Prasyarat: Eclipse 3.6.2, Android SDK, Java Runtime, Apache Server.
- Extension Library: Libcurl, android.jar dan maps.jar (com.google.android.maps)
- Tools Tambahan: Dreamweaver, phpMyAdmin.

*Minimum hardware specification* untuk komputer yang dibutuhkan untuk membangun aplikasi.

- Operating System: Windows XP
- Space Requirement: 1 GB
- Random Access Memory Requirement: 512 MB
- Tools Tambahan: Handphone Android minimum Froyo.

#### IV. SIMPULAN

Kesimpulan yang didapatkan dari penelitian ini adalah.

1. Prototipe aplikasi *tracking* bis kampus ini dirancang untuk bagian Marketing yang mengurus bis kampus dan mahasiswa pengguna jasa transportasi bis kampus dan menggunakan *smartphone* dengan sistem operasi Android versi 2.2 (Froyo). Prototipe aplikasi dibangun

menggunakan Android versi 2.2 (Froyo) karena penggunaanya cukup banyak dan dapat digunakan juga pada *smartphone* dengan sistem operasi Android yang lebih baru dari versi 2.2 (Froyo).

2. Prototipe aplikasi *tracking* bis ini merupakan fasilitator untuk pengguna agar dapat melihat dan mengetahui rute bis kampus, posisi bis kampus, titik pemberhentian bis kampus dan jadwal bis kampus.

#### DAFTAR PUSTAKA

- [1] Connolly, Thomas, and Carolyn Begg. (2009). Database Systems: A Practical Approach to Design, Implementation and Management. Addison Wesley
- [2] Dharwiyanti, S., & Wahono, R. S. (2003). *Pengantar Unified Modelling Language (UML)*. Retrieved September 23, 2011, from Universitas Gunadarma Staff Site: [http://setia.staff.gunadarma.ac.id/Downloads/files/6077/Modul\\_UML.pdf](http://setia.staff.gunadarma.ac.id/Downloads/files/6077/Modul_UML.pdf)
- [3] Gornik, D. (2003). *UML Data Modelling Profile*. Retrieved September 2011, 24, from IBM: <http://www.jeckle.de/files/RationalUML-RDB-Profile.pdf>
- [4] Gosling, James, Bill Joy, Guy Steele, and Gildad Braca. (2005) *The Java™ Language Specification*. Addison Wesley.
- [5] Greenlaw, Raymond dan Hepp, Ellen. (2002). *Fundamentals of the Internet and the world wide web*, Second Edition. McGraw-Hill, New York.
- [6] Hanan, A. (2011). *Panduan PHP Bahasa Indonesia*. Retrieved September 20, 2011, from DOKU SAUNG FOSS: <http://doku.saung-foss.web.id/download/doku-id-php.pdf>
- [7] Hofmeister, C., Nord, R., & Soni, D. (1999). *Describing Software Architecture with UML*. Retrieved September 2011, 24, from Åbo Akademi: <http://www.users.abo.fi/lpetre/SA10/paper99.pdf>
- [8] Kendall, Kenneth. E., dan Kendall, Julie. E. (2005). *System Analysis and Design*, 6th edition. Prentice Hall. New Jersey.
- [9] McNamara, Joel. (2004). *GPS For Dummies*. Willey Publishing, Indiana.
- [10] Pujiadi. (2011, Juni 04). *JAVA Sebagai Bahasa Pemrograman Berorientasi Objek*. Retrieved September 21, 2011, from Universitas Airlangga: [http://skp.unair.ac.id/repository/Guru-Indonesia/JAVASEBAGAIBAHASA\\_PUJIADI,S.Pd.,M.Pd.,M.Kom.\\_8952.pdf](http://skp.unair.ac.id/repository/Guru-Indonesia/JAVASEBAGAIBAHASA_PUJIADI,S.Pd.,M.Pd.,M.Kom._8952.pdf)
- [11] Rational Software Corporation. (1997, September 1). *UML Notation Guide*. Retrieved September 22, 2011, from Washington University in St. Louis School: <http://www.cs.wustl.edu/~kjc/cse132/forms/UML%20notation%20guide.pdf>
- [12] Shneiderman, Ben, Cathrine Plaisant, Maxine Cohen, and Steve Jacobs. (2009). *Designing User Interface: Strategic for Effective Human-Computer Interaction*. Addison Wesley.
- [13] Android, <http://developer.android.com/about/index.html> (terakhir diakses pada tanggal 5 Juli 2012)
- [14] Google Maps, <https://developers.google.com/maps/> (terakhir diakses pada tanggal 5 Juli 2012)
- [15] Hendra Gunawan. "Pengguna Ponsel Naik Menjadi 53 Persen", <http://www.tribunnews.com/2011/05/31/pengguna-ponsel-naik-menjadi-53-persen> (terakhir di akses pada tanggal 6 Juli 2012).
- [16] phpMyAdmin, [http://www.phpmyadmin.net/home\\_page/](http://www.phpmyadmin.net/home_page/) (terakhir diakses pada tanggal 5 Juli 2012).
- [17] Ryan McLaughlin. "Android at 48% Market Share in Q2", <http://hothardware.com/News/Android-at-48-Market-Share-in-Q2/> (terakhir diakses pada tanggal 6 Juli 2012).
- [18] Trimble – All about GPS, [http://www.trimble.com/gps\\_tutorial/](http://www.trimble.com/gps_tutorial/) (terakhir diakses pada 5 Juli 2012). Tabel Penjelasan NMEA Sentence.