**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang Masalah**

Perkembangan telepon selular *(handphone)* pada jaman sekarang sudah mengalami kemajuan yang semakin pesat,menjadi suatu hal yang berguna dan penting bagi masyarakat. hal ini disebabkan karena semakin murahnya harga telepon selular untuk ukuran masyarakat kelas menengah di Indonesia. perkembangan teknologi pada telepon seluler sekarang sudah berkembang dengan menggunakan teknologi sistem operasi Android yang memungkinkan pengguna telepon seluler dengan kata lain smartphone untuk membuat aplikasi berbasis sistem operasi Android yang dapat dijalankan pada telepon selular yang mendukung Android platform. dengan adanya teknologi Android, pengguna smartphone seluler dapat melakukan kreasi sendiri ataupun men-download aplikasi Android untuk kemudian digunakan pada telepon selularnya. semakin murahnya harga handpone di Indonesia, memungkinkan masyarakat jaman sekarang sudah banyak memiliki smartphone Android. Dengan adanya smartphone android dapat membantu pelayanan umum untuk memudahkan masyarakat yang menggunakan kendaraan bermotor roda empat dalam mendeteksi rambu-rambu lalu lintas sekarang ini dan untuk mengurangi adanya pelanggaran rambu-rambu lalu lintas yang berakibatkan terjadinya kemacetan pada kota-kota besar serta bertambahnya angka kecelakaan lalu lintas oleh karena itu dibutuhkan sistem untuk dapat mendeteksi rambu-rambu lalu lintas dengan menggunakan fitur media player atau voice sehingga pengguna dapat dengan nyaman mengakses aplikasi tersebut. Pada smartphone android terdapat aplikasi yang bisa di install yaitu Google maps yang bermanfaat untuk memberikan petunjuk jalan bagi pengguna akan tetapi google maps belum memiliki fitur untuk mendeteksi rambu-rambu lintas berdasarkan latar belakang diatas, maka penulis tertarik untuk membuat suatu aplikasi berbasis mobile phone dengan judul "Simulasi sistem pendeteksi rambu-rambu lalu lintas berbasi android di kota manado"

* 1. **Perumusan Masalah**

Berdasarkan latar belakang masalah diatas, maka perumusan masalahnya adalah bagaimana membuat suatu aplikasi sistem pendeteksi rambu-rambu lalu lintas berbasis android yang bersifat simulasi yang bermanfaat bagi masyarakat khususnya pengguna berkendaraan bermotor untuk dapat mengetahui informasi rambu-rambu lalu lintas melui fitur media player atau suara.

* 1. **Tujuan Penelitian**

Tujuan dalam penulisan Tugas Akhir ini yaitu :

1. Untuk membuat aplikasi simulasi sistem pendeteksi rambu-rambu lalu lintas berbasis Android
2. Mempermudah pengendara kendaraan bermotor roda empat mengetahui adanya rambu-rambu lalu lintas disertai fitur media player (voice)
3. Mengurangi adanya pelanggaran rambu-rambu lintas dan kecelakaan lalu lintas

**1.4 Manfaat Penelitian**

1. Manfaat Akademis

Dengan melakukan pembuatan simulasi sistem pendeteksi rambu rambu lalu lintas berbasis android, dapat menambah wawasan dan pengetahuan bagi para pembaca, terutama kepada pihak yang melakukan pembuatan aplikasi untuk menambah pengetahuan dan keahlian.

1. Manfaat Praktis

Dapat bermanfaat bagi pengguna kendaraan bermotor beroda empat untuk dapat mengetahui informasi rambu-rambu lalu lintas dengan menggunakan media player atau voice.

* 1. **Batasan Masalah**

Karena begitu luasnya permasalahan yang ada, maka pokok permasalahan yang akan dibahas pada penelitian ini hanya dibatasi / dititik beratkan pada hal sebagai berikut :

1. Sistem pendeteksi rambu-rambu lalu lintas yang akan dibuat dalam tugas akhir ini hanya bersifat simulasi
2. Aplikasi dibuat berbasis Android
3. Data-data yang diinput tidak sesuai yang ada dilapangan
4. Bahasa pemrograman yang digunakan adalah Java Language dengan memakai IDE (Integrated Development Environment) Android Studio.
5. Simulasi dibuat untuk pengguna kendaraan bermotor roda empat
6. Aplikasi ini hanya bisa di akses pada Android yang memiliki versi minimal Android 4.1-4.3.1 (Jelly Bean) API Level 16-18
   1. **Sistematika Penulisan**

Sebelum memulai dan merumuskan masalah yang terdapat dalam perusahaan sebagai pembahasan, terlebih dahulu di kemukakan sistematika penulisan yang dipergunakan dalam penulisan Tugas Akhir ini agar lebih terarah sesuai dengan tujuan yang dikehendaki. Adapun sistematika penulisan dari skripsi ini adalah sebagai berikut :

BAB I. PENDAHULUAN

Dalam Bab ini penulis mengemukakan gambaran keseluruhan latar belakang masalah, maksud dan tujuan penulisan skripsi, batasan dan rumusan masalah, serta sistematika penulisan.

BAB II. TINJAUAN PUSTAKA

Menjelaskan dan memaparkan teori-teori yang didapat dari sumber-sumber yang relevan dan efisien untuk digunakan sebagai panduan dalam penelitian serta penyusunan Tugas Akhir dalam pembuatan simulasi sistem pendeteksi rambu-rambu lalu lintas berbasis Android

BAB III METODOLOGI PENELITIAN

Dalam bab ini penulis menjelaskan lebih rinci tentang penelitian. Tempat penelitian, cara mendesain, dan bagaimana dengan metode pengumpulan data dalam penelitian ini, alur program sehingga akan memudahkan dalam penulisan Tugas Akhir ini.

BAB IV. IMPLEMENTASI SISTEM

Dalam bab ini penulis menjelaskan lebih konkrit lagi tentang desain yang akan ditampilkan pada ***“Simulasi sistem pendeteksi rambu-rambu lalu lintas Berbasis Android”*** tersebut, serta informasi apa saja yang akan dimuat di dalam program tersebut, juga hasil output listing program atau sintak-sintak yang digunakan dalam bahasa pemrograman tersebut.

BAB V. PENUTUP

Dalam bab ini, Penulis memaparkan dan menyampaikan kesimpulan yang didapat dari hasil penelitian dan perancangan sistem, serta saran yang diberikan untuk pengembangan selanjutnya yang akan di gunakan, agar dapat bermanfaat dan dalam melakukan perbaikan di kemudian hari.

**BAB II**

**TINJAUAN PUSTAKA**

* 1. **Perangkat Lunak**
     1. **Pengertian Perangkat Lunak**

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak computer yang khusus melakukan pekerjaan tertentu. Dengan kata lain, aplikasi bertugas membantu pengguna untuk mempermudah suatu pekerjaan yang sifatnya khusus. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

**2.1.2. Klasifikasi Perangkat Lunak**

Perangkat Lunak dapat digolongkan menjadi beberapa kelas klasifikasi, antara lain :

1. Perangkat lunak perusahan (enterprise)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk
   * 1. **Aplikasi *Mobile***

Aplikasi *mobile* adalah sebuah aplikasi yang memungkinkan penggunanya untuk melakukan mobilitas tanpa melakukan pemutusan komunikasi. Aplikasi ini biasanya terdapat pada perangkat mobile seperti telepon seluler,tablet, pc, PDA dan perangkat mobile lainnya.

Berikut ini merupakan beberapa karakteristik keunggulan dari aplikasi mobile :

1. Ukuran *file* yang lebih kecil dibandingkan dengan aplikasi desktop.
2. Penggunaan memori yang lebih kecil.
3. Penggunaan daya yang lebih rendah.
4. Melakukan pekerjaan yang lebih ringan atau lebih khusus.
   1. **Android**

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi.Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.Awalnya, Google Inc. membeli Android Inc yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance,* konsordium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC. Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.Pada saat perilisan perdana Android menyatakan mendukung pengembangan open source pada perangkat mobile. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis smartphone yang menggunakan Android sebagai sistem operasinya.Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 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. Pada masa saat ini kebanyakan vendor-vendor *smartphone*sudah memproduksi smart phone berbasis Android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos,Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, LG, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi vendor smart phone didunia yang memproduksi android. Hal ini karena android itu adalah sistem operais yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

Tidak hanya menjadi sistem operasi di *smartphone,* saat ini Android menjadi pesaing utama dari Apple pada sistem operasi Table PC.Pesatnya pertumbuhan Android selain factor yang disebutkan diatas adalah karena Android menjadi pesaing utama dari Apple pada sistem operasi Table PC.Pesatnya pertumbuhan Android selain factor yang disebutkan operasi Table PC. Pesatnya pertumbuhan Android selain factor yang disebutkan diatas adalah karena Android itu sendiri adalah platform yang sangat lengkap baik itu sistem operasinya, Aplikasi dan Tool Pengembangna, Market aplikasi android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia.

**2.4 Kelebihan Android**

Android : Platrom Masa Depan Android dipuji sebagai “*platform* mobile pertama yang Lengkap, Terbuka, dan Bebas”.

1. Lengkap (Complete Platform); Pada desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan platform Android. Android merupakan sistem operasi yang aman dan memungkinkan untuk peluang pengembangan aplikasi.
2. Terbuka (*Open Source Platform);* Platform); Platform Android disediakan melalui lisensi open source, Pengembang dapat dengan bebas untuk mengembangkan aplikasi, Android sendiri menggunakan Linux Kernel 2.6
3. Free (*Free Platform) :*Android adalah *Platform/*aplikasi yang bebas untuk *developer.* Tidak ada lisensi atau biaya royalty untuk dikembangkan pada *platform* Android. Tidak ada biaya royalty untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan di perlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk Android dapat disidtribusikan dan diperdagangkan dalam bentuk apapun. Android merupakan generasi baru *platform mobile, platform* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya.Sistem operasi yang mendasari Android dilisensikan di bawah GNU, General Public Lisensi Versi 2 (GPL.v2), yang sering dikenal dengan istilah “*copyleft”* lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh dibawah terms.Android didistribusikan di bawah Lisensi *Apache Software*(ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya.Komersialisasi pengembang (produsen handset khususnya) dapat memilih untuk meningkatkan platform tanpa harus memberikan perbaikan mereka ke masyarakat *Open Source.*Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan aplikasi merekan di bawah skema lisensi apapun yang mereka inginkan.Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka dibawah skema lisensi apapun yang mereka inginkan. Pengembang memiliki beberapa pilihan ketika membuat membuat aplikasi yang berbasis android. Aplikasi Android dapat dikembangkan pada sistem operasi berikut:
4. Windows XP Vista/Seven.
5. Mac OS X (Mac OS X 10.4.8 atau lebih baru)

**2.5 The Dalvik Virtual Machine**

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di dalam *Dalvik Virtual Machine*(DVM) bukan di *Java Virtual Machine* (JVM) , sebenarnya banyak persamaanya dengan *Java Virtual Machine* (VM) seperti Java ME *(Java Mobile Edition)*, tetapi Android menggunakan *Virtual Machine* sendiri yang menurut saya dikustomisasi dan dirancang untuk memastikan bahwa beberapa *feature- feature* berjalan lebih efisien pada perangkat mobile.

*Dalvik Virtual Machine* (DVM) adalah “register bases” sementara *Java Virtual Machine* (JVM) adalah “stack based”, DVM didesain dan ditulis oleh Dan Bornsten dan beberapa engineers Google lainnya. Jadi bisa kita katakana *“Dalvik equals(Java) == False”.*Dalvik Virtual Machine menggunakan kernel Linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, threading, dan proses serta menajamen memori. Ini memungkinkan kita untuk menulis Aplikasi C / C + sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataan kita harus banyak memahami Arsitektur dan proses sistem dari kernel linux yang digunakan dalam Android tesebut.

Semua *hardware* yang berbasis Android dijalankan dengan menggunakan *Virtual Machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengesekusi *executable file,* sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas Bahasa java dan dikompilasi menggunakan *tools* yang disediakan dalam SDK Android.

**2.6 Android SDK (Software Development Kit)**

Android SDK adalah tools API *(Application Programming Interface)*yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan Bahasa pemograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware*dan aplikasi kunci yang release oleh Google. Saat ini disediakan Android SDK (*Software Development Kit)* sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan Bahasa pemograman Java. Sebagai Platform aplikasi-netral, Android memberi anda kesempatan untuk membuat Aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone.*Beberapa fitur-fitur Android yang paling penting adalah :

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable.*
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile.*
3. *Integrated browser* berdasarkan engine open source Webkit
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi openg ES 1,0 (Opsional akselerasi hardware)
5. SQLite untuk penyimpanan data
6. Media support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Tephony (tergantung hardware)
7. Bluetooth, EDGE, 3G, dan WiFi (tergantung *hardware*)
8. Kamera, GPS , kompas, dan *accelerometer* (tergantung hardware)
9. Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, *tools*untuk *debugging,* profil dan kinerja memori, dan plugin untuk IDE Android Studio.

**2.7 ADT (Android Development Tools)**

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

Mengembangkan aplikasi android dengan menggunakan ADT di ecplise sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi android. Berikut adalah versi ADT untuk ecplise yang sudah dirilis :

* ADT 12.0.0 (July 2011)
* ADT 11.0.0 (June 2011)
* ADT 10.0.1 (March 2011)
* ADT 10.0.0 (February 2011)
* ADT 9.0.0 (January 2011)
* ADT 8.0.1 (December 2010)
* ADT 8.0.0 (December 2010)
* ADT 0.9.9 (September 2010)
* ADT 0.9.8 (September 2010)
* ADT 0.9.7 (May 2010)
* ADT 0.9.6 (March 2010)
* ADT 0.9.5 (Decembar 2009)
* ADT 0.9.4 (October 2009)

Semakin tinggi platform android yang kita gunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya munculnya platform baru diikuti oleh munculnya versi ADT yang terbaru, karena biasanya munculnya platform bari diikuti oleh munculnya versi ADT yang terbaru. Untuk melakukan instalasi ADT di eclipse dapat dilakukan secara on-line maupun offline. Untuk download ADT ini dapat dilakukan di [*http://developer.android.com/sdk/ecplise-adt.html*](http://developer.android.com/sdk/ecplise-adt.html)*.*

**2.14 Pengertian GPS *(Global Positioning System)***

Menurut (Winardi, 2006) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Sistem yang serupa dengan GPS antara lain GLONASS Rusia, Galileo Uni Eropa, IRNSS India. Sistem GPS, yang nama aslinya adalah NAVSTAR GPS (Navigation Satellite Timing and Ranging Global Positioning System), mempunyai tiga segmen yaitu : satelit, pengontrol,

dan penerima / pengguna. Satelit GPS yang mengorbit bumi, dengan orbit dan kedudukan yang tetap (koordinatnya pasti), seluruhnya berjumlah 24 buah dimana 21 buah aktip bekerja dan 3 buah sisanya adalah cadangan. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberinama GPS reciever yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi di ubah menjadi titik yang dikenal dengan nama Way-point nantinya akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik. Sejak tahun 1980, layanan GPS yang dulunya hanya untuk leperluan militer mulai terbuka untuk publik. Uniknya, walau satelit-satelit tersebut berharga ratusan juta dolar, namun setiap orang dapat menggunakannya dengan gratis. (Andy, 2009). Satelit-satelit ini mengorbit pada ketinggian sekitar 12.000 mil dari permukaan bumi. Posisi ini sangat ideal karena satelit dapat menjangkau area coverage yang lebih luas. Satelit-satelit ini akan selalu berada posisi yang bisa menjangkau semua area di atas permukaan bumi sehingga dapat meminimalkan terjadinya blank spot (area yang tidak terjangkau oleh satelit). Setiap satelit mampu mengelilingi bumi hanya dalam waktu 12 jam. Sangat cepat, sehingga mereka selalu bisa menjangkau dimana pun posisi seseorang di atas permukaan bumi.

GPS reciever sendiri berisi beberapa integrated circuit (IC) sehingga murah dan teknologinya mudah untuk di gunakan oleh semua orang. GPS dapat digunakan utnuk berbagai kepentingan, misalnya mobil, kapal, pesawat terbang, pertanian dan di integrasikan dengan komputer maupun laptop. Berikut beberapa contoh perangkat GPS receiver :



**Gambar 2.12** Macam-macam GPS *receiver*

**2.14.1 Sistem GPS**

Sistem GPS terdapat ada tiga bagian yaitu :

1. Sistem Satelit GPS
2. Segmen Pengontrol GPS
3. Segmen Pengguna GPS
4. **Sistem Satelit GPS**

Untuk menginformasikan posisi user, 24 satelit GPS yang ada di orbit sekitar 12,000 mil di atas kita. Bergerak konstan bergerak mengelilingi bumi 12 jam dengan kecepatan 7,000 mil per jam.

Satelit GPS berkekuatan energi sinar matahari, mempunyai baterai cadangan untuk menjaga agar tetap berjalan pada saat gerhana matahari atau pada saat tidak ada energi matahari. Roket penguat kecil pada masing-masing satelit agar dapat mengorbit tepat pada tempatnya.



Gambar 2.13 Simulasi Posisi Satelit GPS

(Sumber: Andi, 2009)

Satelit GPS adalah milik Departemen Pertahanan (Department of Defense)

Amerika, adapun hal-hal lainnya mengenai GPS ini:

1. Nama satelit adalah NAVSTAR

2. GPS satelit pertama kali adalah tahun 1978

3. Mulai ada 24 satelit dari tahun 1994

4. Satelit di ganti tiap 10 tahun sekali

5. GPS satelit beratnya kira-kira 2,000 pounds

6. Kekuatan transmiter hanya 50 watts atau kurang

Satelit-satelit GPS harus selalu berada pada posisi orbit yang tepat untuk menjaga akurasi data yang dikirim ke GPS reciever, sehingga harus selalu dipelihara agar posisinya tepat. Stasiun-stasiun pengendali di bumi ada di Hawaii, Ascension Islan, Diego Garcia, Kwajalein dan Colorado Spring. Stasiun bumi tersebut selalu memonitor posisi orbit jam jam satelit dan di pastikan selalu tepat.

1. **Segmen Pengontrol GPS**

Segmen control terdiri dari stasiun kontrol utama atau stasiun bumi *(Master Ground Station-MGS)* terletak dekat Colorado Springs di Colorado dan stasiun monitor *(Monitor Station-MS)* serta *Ground Antennas Stations (GAS),* yang keseluruhannya bertugas memastikan performa kinerja seluruh sistem satelit dan keakuratannya. Fungsi ini menyangkut beberapa tugas antara lain **:**

* Menjaga agar satelit tetap berada pada lintasan yang seharusnya, dengan jalan mengamati semua satelit secara terus menerus.
* Memantau semua status dan kelayakan semua bagian satelit.
* Menentukan dan menjaga waktu semua system GPS
* Mengkonfigurasi ulang semua peralatan GPS.
* Melacak semua satelit GPS secara aktif dan mengumpulkan data dari masing-masing satelit yang kemudian semua informasi itu dikirim ke MGS, dimana data dan satelit dan parameter waktu ditentukan.

1. **Segmen Pengguna GPS**

Segmen pengguna merupakan bagian penerima dari sistem GPS Fungsinya sebagai penerima sinyal dari dua frekuensi GPS, menguraikan dan memproses sinyal satelit yang diterimanya. Segmen pengguna ini terdiri dari beberapa bagian utama utama yaitu **:** antena dengan pre-amplifier, bagian RF *(Radio Frequency)* dengan pengidentifikasi dan pemroses sinyal dan data, tampilan serta perekam data yang kemudian untuk menentukan dan menampilkan posisi, kecepatan, waktu dan sebagainya.

Karena alat ini bergantung penuh pada satelit , maka sinyal satelit menjadi sangat penting. Alat navigasi berbasisi satelit ini tidak dapat bekerja maksimal ketika ada gangguan pada sinyal satelit. Ada banyak hal yang dapat mengurangi kekuatan sinyal satelit antara lain**:**

* Kondisi geogarafis dimana pengguna masih dapat melihat langit yang cukup luas, alat ini masih dapat berfungsi.
* Hutan. Makin lebat hutannya, maka makin berkurang sinyal yang dapat diterima.
* Air. Jangan berharap dapat menggunakan alat ini ketika menyelam.
* Kaca film mobil, terutama yang mengandung metal.
* Alat-alat elektronik yang dapat mengeluarkan gelombang elektromagnetik.
* Gedung-gedung. Tidak hanya ketika didalam gedung, berada diantara dua buah gedung tinggi juga akan menyebabkan efek seperti berada di dalam lembah.
* Sinyal yang memantul, misal bila berada diantara gedung- gedung tinggi, dapat mengacaukan perhitungan alat navigasi sehingga alat navigasi dapat menunjukkan posisi yang salah atau tidak akurat yang berakibat membingungkan bagi pengguna dalam menentukan arah serta tujuan.

*2.15 Class Diagram*

*Class diagram* merupakan pemodelan yang menggambarkan suatu stuktur sistem dari segi pendefinisian kelas-kelas pada sistem yang di bangun. *Class diagram* akan memberikan gambaran tentang sistem dan relasi-relasi yang ada didalamnya. (Rosa A.S & M. Shalahudin, 2013:141).

**Tabel 2.1** Aturan dan Simbol *Class Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Kelas / *Class*   |  | | --- | | **Nama-kelas** | | +atribut | | +operasi() | | Atribut kelas pada struktur sistem. |
| Antarmuka / *Interface*  Bullet.png  **Nama\_Interface** | Sama dengan konsep *interface* dalam pemrograman berorientasi *object*. |
| Asosiasi / *Association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*. |
| Asosiasi Berarah / *Directed Association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan *multiplicity*. |
| Generalisasi / *Generalization* | Relasi antar kelas dengan makna generalisasi-spesifikasi (umum khusus). |
| Kebergantungan / *Dependency*  Garis.png | Relasi antar kelas dengan makna kebergantungan antar kelas. |
| Agregasi / *Aggregation* | Relasi antar kelas dengan makna semua-bagian (*whole-part*) |

2.16 *Use Case Diagram*

*Use case diagram* merupakan pemodelan yang digunakan untuk menggambarkan perilaku sistem yang akan di bangun. *Use case diagram* mendefinisikan sebuah interaksi antara aktor dengan sistem yang akan dibangun. *Use case diagram* sangat berguna untuk mengetahui fungsi dan siapa saja yang ada di dalam sistem. (Rosa A.S & M. Shalahudin, 2014:155).

**Tabel 2.2** Aturan dan Simbol *Use Case Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Use case  Usecase.png | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *use case*. |
| Aktor / *Actor*  Actors.png  **Nama\_aktor** | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat di luar sistem informasi yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| Asosisasi / *Association* | Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor. |
| Ekstensi / *Extend*  *<< extend >>*  Garis.png | Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan. |
| Menggunakan / *Include*  *<< include >>*  Garis.png | Relasi *use case* tambah ke sebuah *use case* yang ditambahkan memerlukan *include* untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case*. |
| Generalisasi / *Generalization* | Hubungan geralisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| *Dependency*  Garis.png | Sebuah element bergantung dalam beberapa cara ke element lainnya. |
| Asosiasi Berarah / *Directed Association* | Relasi antar *use case* dengan makna *use case* yang satu digunakan oleh *use case* lain |

2.17 *Activity Diagram*

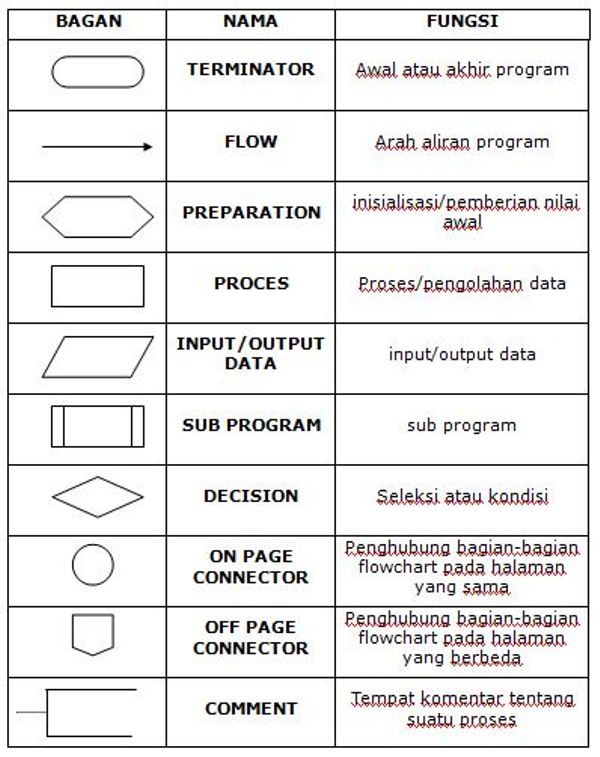
*Activity diagram* adalah pemodelan yang menggambarkan aliran kerja pada sebuah sistem. Disini *activity diagram* tidak menggambarkan aktifitas yang dilakukan oleh aktor, namun menggambarkan aktifitas yang dilakukan oleh sistem. *Activity diagram* menggambarkan bagaimana sistem itu diawali, *decision* yang terjadi pada sistem dan bagaimana sistem itu berakhir. (Rosa A.S & M. Shalahudin, 2014:161).

**Tabel 2.3** Aturan dan Simbol *Activity Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Status Awal.png | Status awal aktififtas sistem, sebuah diagram aktifitas memiliki sebuah status awal. |
| Aktivitas.png | Aktivitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja. |
| Percabangan.png | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan.png | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |

**2.18 Pengertian bagan Alir/Flowchart**

Menurut Krismiaji (2010:71)dalam bukunya yang berjudul *Sistem Informasi Akuntansi,* menjelaskan bahwa Bagan alir merupakan teknik analitis yang digunakan untuk menjelaskan aspek-aspek sistem informasi secara jelas, tepat dan logis.

**Tabel 2.4** Jenis-jenis Flowchart

(Arifianto, R. 2014)

Bagan alir menggunakan serangkaian simbol standar untuk menguraikan prosedur pengolahan transaksi yang digunakan oleh sebuah perusahaan, sekaligus menguraikan aliran data dalam sebuah sistem.

*2.19 Black Box Testing*

*Black Box Testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja pada bagian *internal*. Para *tester* memandang perangkat lunak sebagai “kotak hitam” yang tidak penting di lihat isinya, tapi cukup dikenali proses *testing* di bagian luar, *testing* ini banyak berkonsentrasi pada bagaimana proses berjalannya perangkat lunak. (Soetam Rizky, 2011:265).

**2.20 Metode *Incremental***

Pengertian incremental model pada jurnal berjudul Metode Pengembangan sistem adalah : “Model pengembangan sistem pada *software engineering* berdasarkan *requirement software* yang dipecah menjadi beberapa fungsi atau bagian sehingga model pengembangannya secara bertahap.” *Incremental model* merupakan perbaikan dari *Waterfall Model*. Ide dasar dari model ini adalah mengembangkan *software* secara meningkat (*increment*) atau bertahap berdasarkan kemampuan fungsional. Pada tahun 1971 Harlan Mills (IBM) mengusulkan bahwa lebih baik mengembangkan *software* daripada membuat kembali *software* dari awal. Dimulai dengan membangun sistem sederhana yang mendukung, memiliki 33 fungsi sederhana, kemudian menambahkan dan mengembangkan software tersebut. Incremental model termasuk kategori *evolutionary software process models* karena bersifat iteratif atau mengandung perulangan. Hasil proses berupa produk yang makin lama makin lengkap atau bertambah baik sampai versi terlengkap dihasilkan sebagai produk akhir dari proses yang dilakukan.

1. **Kelebihan Incremental Model**

a. Merupakan model dengan manajemen yang sederhana

b. Pengguna tidak perlu menunggu sampai seluruh sistem dikirim untuk mengambil keuntungan dari sistem tersebut. Increment yang pertama

sudah memenuhi persyaratan mereka yang paling kritis, sehingga perangkat lunak dapat segera digunakan.

c. Resiko untuk kegagalan proyek secara keseluruhan lebih rendah. Walaupun masalah masih dapat ditemukan pada beberapa *increment.*

d. Karena layanan dengan prioritas tertinggi diserahkan pertama dan *increment* berikutnya diintegrasikan dengannya, sangatlah penting bahwa layanan sistem yang paling penting mengalami pengujian yang ketat. Ini berarti bahwa pengguna akan memiliki kemungkinan kecil untuk memenuhi kegagalan perangkat lunak pada increment sistem yang paling bawah.

e. Nilai penggunaan dapat ditentukan pada setiap *increment* sehingga fungsionalitas sistem disediakan lebih awal. Memiliki risiko lebih rendah terhadap keseluruhan pengembagan sistem,

f. Prioritas tertinggi pada pelayanan sistem adalah yang paling diuji

**b.) Kekurangan Incremental Model**

Mungkin terjadi kesulitan untuk memetakan kebutuhan pengguna ke dalam rencana spesifikasi masing-masing hasil *increment*.

**c.) Karakteristik**

Berikut adalah beberapa karakteristik pada *incremental model*:

a. Kebutuhan *user* menjadi prioritas dan prioritas tertinggi dimasukkan dalam awal *increment*.

b. Merupakan kombinasi dari *waterfall model*, yaitu dengan melakukan tahap- tahap *waterfall model* secara iteratif.

c. Hasil atau produk dari *increment* 1 menjadi dasar perbaikan untuk *increment* 2, demikian seterusnya hingga produk dianggap sempurna.

d. Model ini cocok jika anggota tim pengembang perangkat lunak sangat terbatas dan proyek yang dikerjakan berukuran kecil (tidak lebih dari 200.000 baris *coding*).

e. Mungkin terjadi kesulitan untuk memetakan kebutuhan pengguna ke dalam rencana spesifikasi masing-masing hasil *increment*.

f. Produk hasil *increment* pertama biasanya produk inti (*core product*), yaitu produk yang memenuhi kebutuhan dasar. Produk tersebut digunakan oleh pengguna atau menjalani *review* atau pengecekan detil. Hasil *review* tersebut menjadi bekal untuk pembangunan pada *increment* berikutnya. Hal ini terus dikerjakan sampai produk yang komplit dihasilkan.

**2.21 Tahapan *Incremental Model***

Pada awal tahapan dilakukan penentuan kebutuhan dan spesifikasi. Kemudian dilakukan perancangan arsitektur *software* yang terbuka, agar dapat diterapkan pembangunan per-bagian pada tahapan selanjutnya. Penjelasan pada gambar:

1. *Requirement* adalah proses tahapan awal yang dilakukan pada *incremental model* adalah penentuan kebutuhan pengguna
2. *Design* adalah tahap selanjutnya, perancangan software yang terbuka agar dapat diterapkan sistem pembangunan per-bagian pada tahapan selanjutnya.
3. *Implementation* setelah melakukan proses desain selanjutnya ada pengkodean.
4. *Test* merupakan tahap pengujian dalam model ini.