BAB II

LANDASAN TEORI

Pada bab ini dijelaskan tentang dasar teori permasalahan yang digunakan untuk proses analisis design, rekayasa perangkat lunak, implementasi dan dasar teori yang berkaitan dengan permasalahan yang dibahas serta metodelogi pengembangan yang digunakan untuk memperjelas dan memecahkan permasalahan yang ada terutama dari segi perumusan dan identifikasi masalah.

2.1 Teori Tentang Permasalahan

Dengan semakin pesatnya perkembangan ilmu pengetahuandan teknologi saat ini, khususnya tekenologi informasi komputer, maka diiringi pula dengan peningkatan dalam pengolahan data dengan menggunakan komputer. Banyak perusahaan ataupun instansi dan lembaga-lembaga pemerintahan dan swasta yang menggunakan kecanggihan teknologi informasi komputer untuk membuat peningkatan yang signifikan dalam menjalankan roda kegiatan prusahaan, yang menuntun kepada perbaikan terhadap penyajian informasi yang dibutuhkan prusahaan secara cepat, akurat dan tepat guna sehingga diperoleh hasil yang lebih baik. Pada sub bab ini akan membahas mengenai pengertian prangkat lunak, Pemrograman Berorientasi Objek (PBO), Java, Android, Mysql, dan UML (*Unified Modeling Language*), Aplikasi, Peta, Fasilitas Umum, dan pengrtian Kota.

2.1.1 Pengertian Aplikasi

Berikut ini adalah pengertian aplikasi menurut Hendrayudi : “Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus). (Hendrayudi, 2009: 143)”.Dan menurut Ali Zaki : ”program aplikasi adalah komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data.(Ali Zaki, 2007: 11)”.

2.1.2 Pengertian Informasi

Informasi merupakan hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengolahan tersebut bisa menjadi informasi, hasil pengolahan data yang tidak memberikan makna atau arti serta tidak bermanfaat bagi seseorang bukanlah merupakan informasi bagi orang tersebut. Dari uraian tentang informasi ini ada tiga hal penting yang harus diperhatikan disini yaitu:

1. Informasi merupakan hasil pengolahan data
2. Memberikan makna atau arti
3. Berguna atau bermanfaat

Dan berikut merupakan pengertian Informasi menurut Kusrini dan Andri Koniyo : “ Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi.(Kusrini dan Andri Koniyo, 2007: 7)”.

Data belum memiliki nilai sedangkan informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatkannya

Informasi memiliki 3 kriteria :

1. Akurat

   Informasi harus bebas dari kesalahan, tidak bias ataupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

2. Tepat pada waktunya

   Informasi yang datang pada penerima tidak boleh terlambat. Di dalam pengambilan keputusan, informasi yang sudah using tidak lagi bernilai. Bila informasi datang terlambat sehingga pengambilan keputusan terlambat dilakukan, hal itu dapat berakibat fatal bagi perusahaan.

3. Relevan

   Informasi yang disampaikan harus mempunyai keterkaitan dengan masalah yang akan dibahasa dengan informasi tersebut. Informasi harus bermanfaat bagi pemakainya. Disamping karakteristik, nilai informasi juga ikut menentukan kualitasnya. Nilai informasi (value of information) ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih besar dibanding biaya untuk mendapatkannya.

2.1.3 Pengertian peta

Peta adalah gambaran permukaan bumi yang diproyeksikan ke dalam bidang datar denngan skala tertentu. Kartografi merupakan ilmu yang khusus mempelajari segala sesuatu tentang peta. Mulai dari sejarah, perkembangan, pembuatan, pengetahuan, penyimpanan, hingga pengawetan serta cara-cara penggunaan peta. Dalam makalah ini akan dibahas bagaimana proses pemetaan dan simbol pada peta.

Lalu apa fungsi dan tujuan pembuatan peta? Adan beberapa maksud dari pembuatan sebuah peta. Fungsi pembuatan peta antara lain:

1.Dengan adanya peta dapat menunjukkan posisi atau lokasi relatif yang hubungannya dengan lokasi asli dipermukaan bumi.

2. Peta mampu memperlihatkan ukuran.

3. Peta mampu menyajikan dan memperlihatkan bentuk.

4. Mengumpulkan dan menyeleksi data dari suatu daerah dan menyajikan diatas peta dengan simbolisasi.

Sedangkan tujuan pembuatan peta yaitu:

1. Untuk komunikasi informasi ruang.

2. Media menyimpan informasi.

3. Membantu pekerjaan.

4. Membantu dalam desain.

5. Analisis data spatial.

Dari fungsi dan tujuan diatas, maka peta bukan hanya berguna dalam menentukan lokasi namun juga dalam berbagai bidang. Selain itu, pembuatan peta

bukan semata-mata hanya karena untuk memperoleh uang, namun juga sangat berguna bagi hajat hidup masyarakat yang luas dalam keruangan.

Dalam proses pemetaan harus melalui beberapa tahapan mulai dari penyusunan ide hingga peta siap digunakan. Kesemua itu harus dilakukan dengan penuh hati-hati dan ketelitian agar diperoleh peta yang baik dan benar sera memiliki dilai artistik atau seni sehingga pengguna mampu menggunakan peta dengan maksimal dan pembuat dapat menghasilkan peta yang baik sehingga terjadi timbal balik antar pengguna dengan pembuat peta.

Dalam pemberian simbol pada peta juga harus diperhatikan agar peta mudah diketahui dan dipahami isi dan maksud peta tersebut. Pemberian simbol ini juga menentukan nilai keartistikan sebuah peta sehingga peta tersebut enak dipandang dan lebih jelas.

2.1.4 Pengertian Fasilitas Umum

fasilitas umum adalah fasilitas yang diadakan oleh pemerintah atau pihak swasta yang dapat dimanfaatkan oleh masyarakat umum dalam lingkungan pemukiman. Contoh dari fasilitas adalah seperti puskemas, klinik, sekolah, tempat ibadah, pasar, tempat rekreasi, taman bermain, tempat olahraga, ruang serbaguna, makam, dan lain sebagainya. Fasilitas tersebut adalah fasilitas dasar yang dibutuhkan manusia untuk hidup serta fasilitas yang dibutuhkan masyarakat untuk melakukan berbagai aktivitas sosial kemasyarakatan.

2.1.5 Pengertian Android

Menurut buku nazrudin safaat dalam bukunya yang berjudul ANDROID Pemprograman Aplikasi Mobile Smartphone dan Table PC Berbasis Android : "Android adalah sistem operasi untuk perangkat mobile berbasis linux yang mencakup system operasi, middleware dan aplikasi (Nazrudin safaat, 2011: 1)".

Android dikembangkan oleh *Google* bersama *Open Handset Allience* (OHA) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan Hardware, Software dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

2.1.5.1 Sejarah Android

Pada mulanya terdapat berbagai macam sistem operasi pada perangkat selular, iantaranya sistem operasi *Symbian, Microsoft Windsos Mobile, Mobile Linux, iPhone,* dan sistem operasi lainnya. Namun diantara sistem operasi yang ada belum mendukung standar dan penerbitan API yang dapat dimanfaatkan secara keseluruhan dan dengan biaya yang murah. Kemudian *Google* ikut berkecimpung didalamnya dengan platform Android, yang menjanjikan keterbukaan, keterjangkauan, open source, dan *framework* berkualitas.

Pada tahun 2005, Google mengakuisisi perusahaan Android Inc. untuk memulai pengembangan platform Android. Dimana terlibat dalam pengembangan ini Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada pertengahan 2007 sekelompok pemimpin industri bersama-sama membentuk aliansi perangkatselular terbuka, *Open Handset Alliance* (OHA). Bagian dari tujuan

aliansi ini adalah berinovasi dengan cepat dan menanggapi kebutuhan konsumen dengan lebih baik, dengan produk awalnya adalah platform Android. Dimana Android dirancang untuk melayani kebutuhan operator telekomunikasi, manufaktur handset, dan pengembang aplikasi. OHA berkomitmen untuk membuat Android open source dengan lisensi Apache versi II.0.

Android pertama kali diluncurkan pada 5 November 2007, dan smartphone pertama yang menggunakan sistem operasi Android dikeluarkan oleh T-Mobile dengan sebutan G1 pada bulan September 2008. Hingga saat ini Android telah merilis beberapa versi Android untuk menyempurnakan versi sebelumnya. Selain berdasarkan penomoran, pada setiap versi Android terdapat kode nama berdasarkan nama-nama kue. Hingga saat ini sudah terdapat beberapa versi yang telah diluncurkan, diantaranya: versi 1.5 dirilis pada 30 April 2009 diberi nama *Cupcake*, versi 1.6 dirilis pada 15 September 2009 diberi nama *Donut*, dan versi terakhir II.0 dirilis pada 26 Oktober 2009 diberi nama *Eclair*.

2.1.5.2 Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

1. *Application* dan *Widget*

   Application dan *Widgets* ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk

klien email, program sms, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi tersebut di tulis menggunakan bahasa pemprograman java.

2. *Application Frameworks*

Android *Application Frameworks* adalah *open development platform* yaitu android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi, *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*).

Sehingga bisa kita simpulkan *Applications Frameworks* ini adalah layer dimana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan disistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti content providers yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Application Frameworks* adalah sebagai berikut :

a. *Views*
b. *Content Provider*
c. *Resource Manager*
d. *Notification Manager*

e. Activity Manager

3. Libraries

Libraries ini adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, Layer ini meliputi berbagai library C/C++ inti seperti Libe dan SSL, serta :

a. Libraries media untuk pemutaran media audio dan video
b. Libraries untuk manajemen tampilan
c. Libraries graphics mencakup SGL dan OpenGL untuk grafis 2D dan 3D
d. Libraries SQLite untuk dukungan database
e. Libraries SSL dan Wevkit terintegrasi dengan web browser dan security
f. Libraries LiveWebcore mencakup modern web browser dengan engine embedded web view
g. Libraries 3D yang mencakup implementasi OpenGL ES 1.0 API's

4. Android Run Time

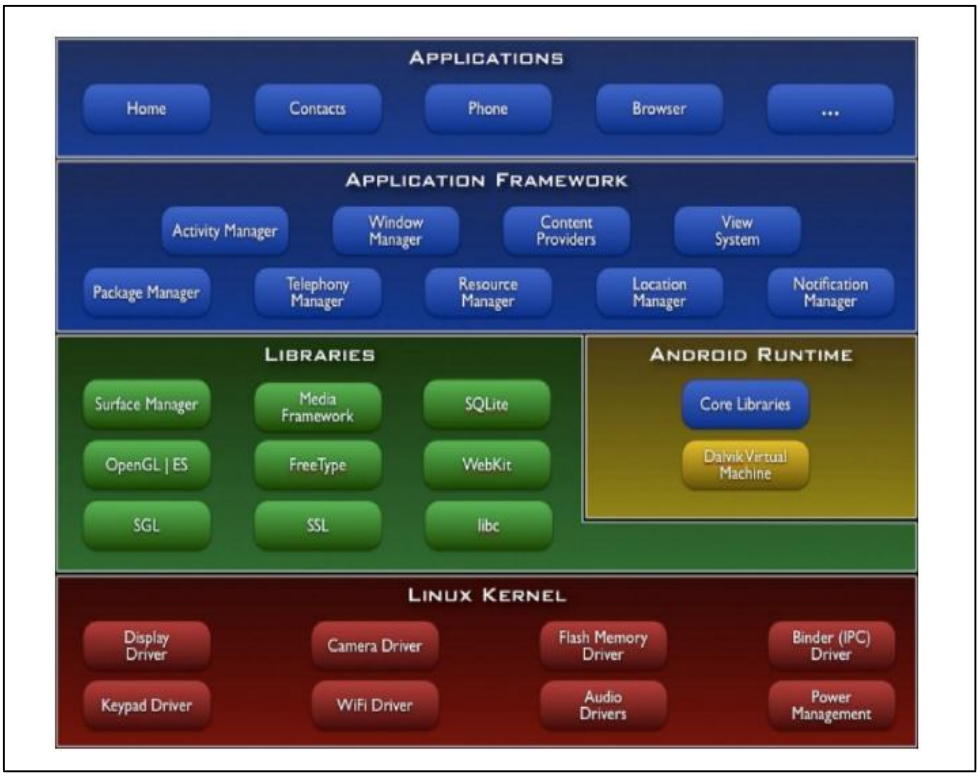
Layer yang membuat aplikasi Android dapat dijalankan dimana didalam prosesnya menggunakan implementasi linux, dalvik virtual machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi android. Di dalam android run time dibagi menjadi dua bagian yaitu :

a. Core libraries : aplikasi android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya buka virtual machine java, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh core libraries

b. Dalvik Virtual Machine : Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan threading dan manajemen rendah

5. Linux Kernel

   Linux kernel adalah layer dimana inti dari operating *system* dari android itu berada. Berisi file-file *system* yang mengatur *system processing, memory, resource, drivers,* dan sistem-sistem operasi android lainnya, linux kernel yang digunakan android adalah linux kernel release II.6



Gambar 2.1 Arsitektur Android

2.1.5.3 Fundamental Aplikasi

Aplikasi android ditulis dalam bahasa pembprograman java. Kode java dikompilasi bersama dengan data file *resource* yang ditubuthkan oleh aplikasi, dimana prosesnya di *package* oleh *tools* yagn dinamakan "apt *tools*" ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. File itulah yang kita sebut dengan aplikasi, dan nantinya dapat di install di perangkat mobile

Ada enam jenis komponen pada aplikasi android, yaitu :

a. *Activities*

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena Activity menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap Event. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada activity yang pertama kali tampil ketika aplikasi dijalankan.

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

b. Service

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di background untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi.

Service digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

c. *Broadcast Receiver*

Broadcast Receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *Broadcast* berasal dari sistem misalnya, Batre sudah hampir habis, informasi zona waktu telah berubah, atau pengguna telah merubah bahasa default pada perangkat. Sama halnya dengan *service*, Broadcast Receivers tidak menampilkan antarmuka pengguna. Namun, Broadcast Receivers dapat menggunakan Notification *Manager* untuk memberitahukan sesuatu kepada pengguna.

d. *Content Provider*

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

2 1.5.4 Tipe Aplikasi Android

Terdapat tiga aplikasi android menurut (Reto Meier, 2009, 29) :

a. *Foreground Activity*

Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini pasti

mempertimbangkan siklus hidup *activity*, sehingga perpindahan antar *activity* dapat berlangsung dengan lancar.

b. *Background Service*

Aplikasi yang memiliki interaksi terbatas dengan *user*, selain dari pengaturan konfigurasi, semua dari prosesnya tidak tidak tampak pada layar. Contohnya aplikasi penyaringan panggilan atau sms *auto* respon.

c. *Intermittent Activity*

Aplikasi yang masih membutuhkan beberapa masukkan dari pengguna, namun sebagian sangat efektif jika dijalankan di *background* dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu. Contohnya pemutar musik.

Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi aplikasi memiliki ciri-ciri dari semua kategori. Oleh karenanya perlu pertimbangan bagaimana aplikasi tersebut digunakan dan menentukan kategori aplikasi yang sesuai.

2.1.5.5 Kelebihan Android

Sudah banyak platform untuk perangkat selular saat ini, termasuk didalamnya *Symbian*, iPhone, *Windows Mobile*, *BlackBerry*, Java *Mobile Edition*, Linux *Mobile* (LiM0), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut (Muhammad amiral, 2010, 34) :

a. Lengkap (*Complete Platform*)

Para designer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan platform Android. Android merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun software dan memungkinkan untuk peluang pengembang aplikasi.

b. Terbuka (Open Source Plarform)

Platform Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux kernel II.6

c. *Free* (*free plarform*)

Android adalah *platform*/aplikasi yang bebas untuk *developer*. Tidak ada lisensi atau biaya *royalty* untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

2.1.6 Pengertian *Google Maps*

Google Maps adalah layanan gratis peta dan pemetaan digital yang bisa dimanfaatkan untuk mengamati peta dunia melalui *browser web*.

Google Maps memanfaatkan teknologi digital *imaging*, seperti foto Satelit sehingga kamu bisa melihat bagaimana landscape planet bumi apabila dilihat dari luar angkasa. Jadi, tanpa harus capek capek naik pesawat ulang alik untuk pergi

ke bulan seperti astronot Neil Amstrong, kita sudah bisa menikmati permukaan bumi.

2.1.6.1 Foto Satelit

Pengertian foto satelit adalah foto yg diambil dari luar angkasa dgn sudut pandang mata burung (*Bird's Point of view*). Sementara itu, penentuan lokasi pasti sebuah objek foto satelit dihasilkan melalui komputasi tringulasi satelit.

Definisi Tringulasi adalah Metode navigasi yang menggunakan rumus trigonometri sebuah segitiga dalam mengkunci posisi dan lokasi objek. Logikanya seperti menggunakan kompas sebagai salah satu sudut segitiga dan 2 sudut lagi sebagai 2 posisi dgn jarak pemisah antara keduanya telah diketahui sebelumnya. Contoh teknologi tringulasi satelit adalah GPS (*Global Positioning System*) yg memanfaatkan minimal 24 buah satelit yg mengorbit mengelilingi bumi dalam menentukan lokasi pengguna.

Tringulasi satelit memungkinan mengumpulan gambar grafis suatu wilayah secara remote. Tringulasi satelit ini akan menghasilkan foto dengan potensi error lebih kecil. Teknologi Foto Satelit ini bisa dimanfaatkan dalam melacak hotspot kebakaran hutan kalimantan, mengikuti perjalanan badai atau tornado gustav, dan juga operasi militer seperti misi intelejen inflitrasi pasukan ke *Hotzone*.

2.1.7 Pengertian GPS

Gps atau disebut juga *Global Positioning System* adalah sistem navigasi berbasis satelit yang dikembangkan oleh departemen pertahanan amerika. GPS terdiri dari 3 segmen :

1. Segmen angkasa
2. Segmen control / pengendali
3. Segmen pengguna

Dimana segmen angkasa terdiri dari 24 satelit yang beroperasi dalam 6 orbit pada ketinggian 20.200 km dan inklinasi 55 derajat dengan periode 12 jam (satelit akan kembali ke titik yang sama dalam 12 jam). Satelit tersebut memutari orbitnya sehingga minimal ada 6 satelit yang dapat dipantau pada titik manapun di bumi ini. Satelit tersebut mengirimkan posisi dan waktu kepada pengguna seluruh dunia.

Setiap satelit mentransmisikan dua sinyal yaitu 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 (perangkat 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.

Perangkat GPS yang dikhususkan buat sipil hanya menerima kode C/A pada sinyal L1 (meskipun pada perangkat GPS yang canggih dapat memanfaatkan sinyal L2 untuk memperoleh pengukuran yang lebih teliti). Perangkat GPS

menerima sinyal yang ditransmisikan oleh satelit GPS. Dalam menentukan posisi, kita membutuhkan paling sedikit 3 satelit untuk penentuan posisi 2 dimensi (lintang dan bujur) dan 4 satelit untuk penentuan posisi 3 dimensi (lintang, bujur, dan ketinggian). Semakin banyak satelit yang diperoleh maka akurasi posisi kita akan semakin tinggi. Untuk mendapatkan sinyal tersebut, perangkat GPS harus berada di ruang terbuka. Apabila perangkat GPS kita berada dalam ruangan atau kanopi yang lebat dan daerah kita dikelilingi oleh gedung tinggi maka sinyal yang diperoleh akan semakin berkurang sehingga akan sukar untuk menentukan posisi dengan tepat atau bahkan tidak dapat menentukan posisi. Melalui GPS kita dapat mengetahui keberadaan suatu objek di mana pun objek itu berada di seluruh muka bumi baik di darat, laut maupun udara.

2.1.8 Pengertian Pemrograman Berorientasi Objek.

Object oriented programming (OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam perangkat lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Adapun arti Pemograman Berorientasi Objek menurut Bambang Heryanto dalam bukunya yang berjudul Paradigma Pemograman berorientasi Objek :

> “Pemrograman berorientasi objek adalah suatu metode implementasi dimana program diorganisasikan sebagai kumpulan objek yang bekerja sama, masing-masing objek meninpresentasikan tautan dari kelas dan kelas-kelas itu merupakan anggota hiraki kelas-kelas yang disatukan lewat keterhubungan pewarisan. ” (Bambang Heryanto, 2004.paradigma pemograman berorientasi Objek.madiun.)

Tiga faktor penting dalam pemrogam berorientasi objek adalah:

1. Menggunakan objek-objek yang bukan algoritma-algoritma sebagai blok-blok bangun logi dasar.
2. Masing-masing objek adalah instan suatu kelas.
3. Kelas-kelas saling nerhubungan lewat keterhubungan pewaris.

Faktor utama ditemukannya pendekatan berorientasi objek dikarenakan adanya kekurangan-kekurangan pada pendekatan terstruktur, biaya perkembangan perangkat lunak berkembang seiring berkembangnya kebutuhan dari pengguna perangkat lunak, pemeliharaan yang sukar, lamanya penyelesaian suatu proyek, jangka waktu proyek yang hampir selalu terlambat, biaya perkembangan prangkat lunak dan sebagainya.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pemrogaman berorientasi objek adalah metode implementasi dimana program diorganisasikan sebagai komunitas objek yang bekerja sama dan berinteraksi. Masing-masing objek merepresintasikan instant suatu kelas yang menyediakan sejumlah layanan yang relatif sederhana, sehingga objek-objek tersebut saling berkomunikasi dan

bertukar informasi. Perilaku kompleks diperoleh darikerjasama antar objek-objek, kelas-kelas dapat merupakan anggota hirarki yang disatukan lewat hubungan pewarisan. Selain itu juga, pemrograman berorientasi objek dapat menciptakan suatu sinergi yang luar biasa sepanjang siklus pengembangan perangkat lunak (perencanaan, analisis, perancangan, implementasi, serta pengujian), sehingga dapat diterapkan pada perencanaan system secara umum menyangkut perangkat lunak, perangkat keras, dan system informasi secara keseluruhan.

2.2. Bahasa Pemrogaman yang Digunakan.

Adapun bahasa yang digunakan dalam pengembangan aplikasi penjualan ini adalah Java serta dalam pembangunan database mengunakan MySql, yang merupakan fasilitas yang sudah tersedia dalam NetBeans IDE 6.7.1.

2.2.1 Java

> “Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus

didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web. http://id.wikipedia.org/wiki/Java.

2.2.2 MySql.

" MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia

yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.” . http://id.wikipedia.org/wiki/MySql .

2.2.3 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:

- *Multi-platform*: Target sistem operasi Eclipse adalah *Microsoft Windows*, *Linux*, *Solaris*, AIX, HP-UX dan Mac OS X.
- *Mulit-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan *web*, dan lain sebagainya.

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

2.2.3.1. Sejarah eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM *Visual Age for* Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta untuk pengembangannya. Semenjak itu konsursium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

2.2.3.2 Arsitektur eclipse

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

- *Core platform*
- OSGi
- SWT (*Standard Widget Toolkit*)
- *JFace*
- *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 yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (C/C++ *Development Tools*). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya. Salah satu situs yang menawarkan plug-in secara gratis seperti Eclipse *downloads by project*.

2.2.3.3 Versi Peluncuran

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 dari 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.

Kode Peluncuran	Tanggal Peluncuran	*Platform*	Nama Proyek
Eclipse 3.0	28 Juni 2004	3.0	Eclipse Projects
Eclipse 3.1	28 Juni 2005	3.1	Eclipse Projects
Callisto	30 Juni 2006	3.2	Callisto projects
Europa	29 Juni 2007	3.3	Europa projects
Ganymede	25 Juni 2008	3.4	Ganymede projects
Galileo	24 Juni 2009	3.5	Galileo projects
Helios	23 Juni 2010	3.6	Helios Projects
Juno	21 juli 2012	3.7	Juno Projects

2.2.3.4 Kontroversi Eclipse

Hadirnya Eclipse di tengah-tengah persaingan IDE, terutama Java IDE, menimbulkan banyak kontroversi. Salah satunya adalah penggunaan user interface Eclipse. *Sun Microsystems* sebagai perusahaan yang membuat Java, selama ini mengeluarkan dua *library* untuk pengembangan *Graphical User Interface* (GUI), yaitu *Abstract Windowing Toolkit* (AWT) dan *Swing*. Akan tetapi dalam pembuatan Eclipse, dua library ini ditinggalkan karena berbagai alasan,

yang salah satunya adalah kedua library ini tidak memberikan look and feel yang baik. Sehingga untuk hal ini dikembangkanlah GUI-*library* yang baru.

Selain itu, *Sun* yang mempunyai IDE sendiri untuk pengembangan Java bernama *NetBeans*, menjadi gerah atas kehadiran Eclipse yang sampai saat ini komunitasnya cukup besar. Hal ini terlihat dengan menolaknya Sun untuk bekerja sama dalam Eclipse *Foundation*, walau telah diundang.

Nama Eclipse seolah-olah memberi pesan explisit: cahaya *Sun* (matahari) yang pancarannya dihalangi oleh *Eclipse* (gerhana).