

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

II.1.1. Sistem

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Prajudi Atmosudirdjo menyatakan bahwa suatu sistem terdiri atas objek-objek, atau unsur-unsur, atau komponen-komponen yang berkaitan dan berhubungan satu sama lainnya sedemikian rupa sehingga unsur- unsur tersebut merupakan suatu kesatuan pemrosesan atau pengolahan yang tertentu. (Tata Sutabri;2004:2).

Ada beberapa pendapat mengenai sistem di antaranya adalah :

- a. Sistem adalah Suatu kesatuan yang terdiri dari dua atau lebih komponen atau sub sistem yang berinteraksi untuk mencapai suatu tujuan. (Jogiyanto Hartono: 2005:683).
- b. Sistem adalah seperangkat elemen yang membentuk suatu prosedur atau bagan yang mencari suatu tujuan atau tujuan-tujuan bersama dengan mengoperasikan data atau barang pada waktu rujukan tertentu untuk menghasilkan informasi atau energi atau barang. (Robert G. Murdick/Joel E. Ross/James R. Clageet : 1993:6).

Jadi pengertian sistem secara umum adalah jaringan kerja sama bagian-bagian atau unsur-unsur yang saling berhubungan guna mencapai tujuan yang diinginkan.

II.1.2. Karakteristik Sistem

Model umum sebuah sistem terdiri dari input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud menurut (Tata Sutabri ; 2004 : 11) adalah sebagai berikut :

a. Komponen sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk suatu kesatuan. Komponen- komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut dengan supra sistem.

b. Batasan sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

c. Lingkungan luar sistem (*Environment*)

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

d. Penghubung sistem (*Interface*)

Sebagai media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

e. Masukan sistem (*Input*).

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

f. Keluaran sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

g. Pengolahan sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Sasaran sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

II.1.3. Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan (Tata Sutabri : 2005 : 23).

II.1.4. Sistem Informasi

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar dengan laporan-laporan yang diperlukan (Tata Sutabri : 2005 : 42).

II.2. Sistem Informasi geografis

Sistem Informasi Geografi (SIG) merupakan suatu rancangan sistem informasi untuk mengerjakan data berunsur atau koordinat geografis. Teknologi SIG menyatu dengan operasi database seperti pencarian data dan analisa statistik dan analisa geografis yang disajikan dalam bentuk peta. (Riadi dan Kifli Nasution;2008:4).

a. Komponen Sistem Informasi Geografis

Menurut (Ricky Agus : 2012:5), Sistem Informasi Geografis bekerja berdasarkan integrasi komponen, yaitu: Hardware, Software, Data, Manusia, dan Metode. Kelima komponen tersebut dapat dijelaskan sebagai berikut:

1. Hardware

Sistem Informasi Geografis memerlukan spesifikasi komponen hardware

yang sedikit lebih tinggi dibanding spesifikasi komponen sistem informasi lainnya. Hal tersebut disebabkan karena data-data yang digunakan dalam SIG, penyimpanannya membutuhkan ruang yang besar dan dalam proses analisisnya membutuhkan memory yang besar dan processor yang cepat. Beberapa Hardware yang sering digunakan dalam Sistem Informasi Geografis adalah: Personal Computer (PC), Mouse, Digitizer, Printer, Plotter, dan Scanner.

2. Software

Sebuah software SIG haruslah menyediakan fungsi dan tool yang mampu melakukan penyimpanan data, analisis, dan menampilkan informasi geografis.

3. Data

Hal yang merupakan komponen penting dalam SIG adalah data. Secara fundamental, SIG bekerja dengan 2 tipe model data geografis, yaitu model data vector dan model data raster. Dalam model data vector, informasi posisi point, garis, dan polygon disimpan dalam bentuk koordinat x,y. Bentuk garis, seperti jalan dan sungai dideskripsikan sebagai kumpulan dari koordinat-koordinat point. Bentuk polygon, seperti daerah penjualan disimpan sebagai pengulangan koordinat yang tertutup. Data raster terdiri dari sekumpulan grid atau sel seperti peta hasil scanning maupun gambar atau image. Masing-masing grid memiliki nilai tertentu yang bergantung pada bagaimana image tersebut digambarkan.

4. Manusia

Komponen manusia memegang peranan yang sangat menentukan, karena tanpa manusia maka sistem tersebut tidak dapat diaplikasikan dengan baik. Jadi manusia menjadi komponen yang mengendalikan suatu sistem sehingga menghasilkan suatu analisa yang dibutuhkan.

5. Metode

SIG yang baik memiliki keserasian antara rencana desain yang baik dan aturan dunia nyata, dimana metode, model dan implementasi akan berbeda untuk setiap permasalahan.

II.2.1. Sistem Development Life Cycle (SDLC)

Menurut (**James R; 2001:45**) Proses pengembangan sistem mempunyai beberapa tahapan mulai dari sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan dan dipelihara. Adapun tahapan-tahapan tersebut adalah sebagai berikut:

A. Perencanaan Sistem (*system planning*)

1. Penerimaan untuk studi sistem
2. Pengamatan awal
3. Studi kelayakan

B. Analisa Sistem (*system analysis*) meliputi beberapa langkah:

1. Mendefenisikan kembali masalah
2. Memahami sistem yang ada
3. Menentukan permintaan pemakai dan kendala yang ada
4. Membuat logika dan menyelesaikan

C. Desain Sistem (*system design*) meliputi:

1. Desain output
2. Desain input

D. Implementasi Sistem (*system implementation*) meliputi beberapa langkah:

1. Membangun sistem
2. Pembuatan program

II.2.2. Perancangan Antar Muka

Menurut Fathansyah dalam bukunya yang berjudul Perancangan Sistem (2002:93) yang menjadi salah satu kriteria penting dari sebuah antar muka adalah tampilan yang menarik. Seorang pengguna, apalagi pengguna baru, biasanya tertarik untuk mencoba sebuah program aplikasi dengan terlebih dahulu tertarik pada suatu tampilan yang ada di hadapan matanya. Dokumentasi rancangan dapat dikerjakan atau dilakukan dalam beberapa cara:

- a. Membuat sketsa pada kertas
- b. Menggunakan peranti prototipe GUI.
- c. Menuliskan tekstual yang menjelaskan tentang kaitan antara satu jendela dengan jendela yang lain,
- d. Menggunakan peranti bantu yang disebut CASE (Computer Aided Software Engineering).

II.2.3. Cara Pendekatan Perancangan Antar Muka

Menurut Fathansyah dalam bukunya yang berjudul Perancangan Sistem (2002:102) Program aplikasi, pada dasarnya dapat dikelompokkan ke dalam dua kategori besar, yakni program aplikasi untuk keperluan khusus dengan pengguna yang khusus pula (*special purpose software*) dan program aplikasi yang akan digunakan oleh banyak pengguna (*general purpose software*), yang juga sering dikenal dengan sebutan *public software*. Karena perbedaan pada calon pengguna, maka perancang program antarmuka harus benar-benar memperhatikan hal ini.

Pada kelompok pertama, yakni pada program aplikasi untuk keperluan khusus, misalnya program aplikasi untuk inventori gudang, pengelolaan data akademis mahasiswa, pelayanan reservasi hotel, dan program-program aplikasi serupa, kelompok calon pengguna yang akan memanfaatkan program aplikasi tersebut dapat dengan mudah diperkirakan, baik dalam hal keahlian pengguna, maupun ragam antarmuka yang akan digunakan. Untuk kelompok ini ada satu pendekatan yang dapat dilakukan, yakni pendekatan yang disebut dengan *user-centered design approach*. Cara pendekatan ini berbeda dengan *user design approach*.

Pendekatan secara *user centered design* adalah perancangan antarmuka yang melibatkan pengguna. Pelibatan pengguna disini tidak diartikan bahwa pengguna harus ikut memikirkan bagaimana implementasinya nanti, tetapi pengguna diajak untuk aktif berpendapat ketika perancang antarmuka sedang menggambar “wajah” antarmuka. Dengan kata lain, perancang dan pengguna duduk bersama-sama untuk merancang wajah antarmuka yang diinginkan

pengguna. Pengguna menyampaikan keinginannya, sementara perancang menggambar keinginan pengguna tersebut sambil menjelaskan keuntungan dan kerugian wajah antarmuka yang diinginkan oleh pengguna, serta kerumitan implementasinya. Dengan cara seperti ini, pengguna seolah-olah sudah mempunyai gambaran nyata tentang antarmuka yang nanti akan mereka gunakan

II.2.4. Prinsip Dan Petunjuk Perancangan Antar Muka

Antar muka pengguna secara alamiah terbagi menjadi empat komponen: model pengguna, bahasa perintah, umpan balik dan penampilan informasi. Model pengguna merupakan dasar dari tiga komponen yang lain. Model pengguna merupakan model konseptual yang diinginkan oleh pengguna dalam memanipulasi informasi dan proses yang diaplikasikan pada informasi tersebut.

Setelah pengguna mengetahui dan memahami model yang ia inginkan, dia memerlukan peranti untuk memanipulasi model itu. Peranti pemanipulasian model ini sering disebut dengan bahasa perintah (command language), yang sekaligus merupakan komponen kedua dari antarmuka pengguna. Idealnya, program komputer kita mempunyai bahasa perintah yang alami, sehingga model pengguna dengan cepat dapat dioperasikan.

Komponen ketiga adalah umpan balik. Umpan balik di sini diartikan sebagai kemampuan sebuah program yang membantu pengguna untuk mengoperasikan program itu sendiri. Umpan balik dapat berbentuk: pesan-pesan penjelasan, pesan penerimaan perintah, indikasi adanya obyek terpilih, dan penampilan karakter yang diketikkan lewat papan ketik. Beberapa bentuk umpan balik terutama ditujukan kepada pengguna yang belum berpengalaman dalam

menjalankan program aplikasi itu. Umpan balik dapat digunakan untuk memberi keyakinan bahwa program telah menerima perintah pengguna dan dapat memahami maksud perintah tersebut. Komponen keempat adalah tampilan informasi. Komponen ini digunakan untuk menunjukkan status informasi atau program ketika pengguna melakukan suatu tindakan. Pada bagian ini, perancang harus menampilkan pesan-pesan tersebut seefektif mungkin, sehingga mudah dipahami oleh pengguna. (Fathansyah; 2002 : 65).

II.2.5. Urutan Perancangan

Perancangan dialog, seperti halnya perancangan sistem uang lain, harus dikerjakan secara top-down. Proses perancangannya dapat dikerjakan secara stepwise refinement sebagai berikut:

- a. Pemilihan Ragam Dialog, dipengaruhi oleh karakteristik populasi pengguna (pengguna mula, menengah, atau pengguna ahli), tipe dialog yang diperlukan, dan kendala teknologi yang ada untuk mengimplementasikan ragam dialog tersebut. Ragam dialog yang terpilih dapat berupa sebuah program tunggal, atau sekumpulan ragam dialog yang satu sama lain saling mendukung.
- b. Perancangan Struktur Dialog, Tahap kedua adalah melakukan analisis tugas dan menentukan model pengguna dari tugas tersebut untuk membentuk struktur dialog yang sesuai. Dalam tahap ini pengguna banyak dilibatkan, sehingga pengguna langsung mendapatkan umpan balik yang berupa diskusi informal maupun prototipe dari dialog yang nantinya akan ia digunakan.

- c. Perancangan Format Pesan, Pada tahap ini tata letak tampilan dan keterangan tekstual secara terinci harus mendapat perhatian lebih. Selain itu, kebutuhan data masukan yang mengharuskan pengguna untuk memasukkan data ke dalam komputer juga harus dipertimbangkan dari segi efisiensinya. Salah satu contohnya adalah dengan mengurangi pengetikan yang tidak perlu dengan cara mengefektifkan penggunaan tombol.
- d. Perancangan Penanganan Kesalahan, Bentuk-bentuk penanganan kesalahan yang dapat dilakukan antara lain adalah:
- Validasi pemasukan atau: misalnya jika pengguna harus memasukkan bilangan positif, sementara ia memasukkan data negatif atau nol, maka harus ada mekanisme untuk mengulang pemasukan data tersebut.
 - Proteksi pengguna: program memberi peringatan ketika pengguna melakukan suatu tindakan secara tidak sengaja, misalnya penghapusan berkas.
 - Pemulihan dari kesalahan: tersedianya mekanisme untuk membatalkan tindakan yang baru saja dilakukan.
 - Penampilan pesan salah yang tepat dan sesuai dengan kesalahan yang terjadi pada waktu itu.
 - Perancangan Struktur Data, Struktur data yang diperlukan untuk mengimplementasikan dialog berbasis grafis jauh lebih rumit dibandingkan dengan struktur data yang diperlukan pada dialog berbasis tekstual. Meskipun demikian, sesulit atau semudah apapun struktur data yang akan digunakan, struktur data tersebut harus diturunkan dari

spesifikasi antarmuka yang telah dibuat. Hal ini perlu ditekankan agar keinginan pengguna dan model sistem yang telah dirancang saling mempunyai kecocokan satu sama lain.

Perancangan Tampilan Berbasis Teks, Pada perancangan tampilan untuk antarmuka berbasis teks, ada enam faktor yang harus dipertimbangkan agar diperoleh tata letak tampilan yang berkualitas tinggi. (Fathansyah; 2002 : 98).

II.3. Java

Java adalah suatu teknologi di dunia *software* komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu *platform*. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi *programmer* yang telah mengenal C/C++. Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan. sebagai bahasa

Pemrograman Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua *platform*. Dan juga dirancang untuk menghasilkan aplikasi-aplikasi dengan performansi yang terbaik, seperti aplikasi database Oracle 8i/9i yang core-nya dibangun menggunakan bahasa pemrograman Java. Sedangkan Java bersifat *neutral architecture*, karena *Java Compiler* yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut sebagai *Java Bytecode*. Sebagai sebuah *platform*, Java terdiri atas dua bagian utama, yaitu:

1. *Java Virtual Machine (JVM)*.
2. *Java Application Programming Interface (Java API)*.

II.3.1. Sejarah Java

Java diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari *Sun Microsystem* yang memiliki kode *Green* dengan tujuan untuk menghasilkan bahasa komputer sederhana yang dapat dijalankan di peralatan sederhana dengan tidak terikat pada *arsitektur* tertentu. Mulanya disebut OAK, tetapi karena OAK sendiri merupakan nama dari bahasa pemrograman computer yang sudah ada. Maka Sun mengubahnya menjadi Java.

Sun kemudian meluncurkan *browser* dari Java yang disebut *Hot Java* yang mampu menjalankan *applet*. Setelah itu teknologi Java diadopsi oleh *Netscape* yang memungkinkan program Java dijalankan di *browser Netscape* yang kemudian diikuti *Internet Explorer*. Karena keunikanya dan kelebihanya, teknologi Java mulai menarik banyak *vendor* seperti IBM, *Symantec*, *Inprise*.

Sun merilis versi awal Java secara resmi pada awal tahun 1996 yang kemudian terus berkembang hingga muncul JDK 1.1, kemudian JDK 1.2 yang mulai disebut sebagai versi Java2 karena banyak mengandung peningkatan dan perbaikan. Perubahan utama adalah adanya *Swing* yang merupakan teknologi GUI (Graphical User Interface) yang mampu menghasilkan window yang *portabel*. Dan pada tahun 1998-1999 lahirlah teknologi J2EE (Java 2 Enterprise Edition) yang berbasis J2SE yang diawali dengan servlet dan EJB kemudian diikuti JSP. Java juga menjadi lebih cepat populer di lingkungan *server side* dikarenakan

kelebihannya di lingkungan network dan terdistribusi serta kemampuan *multithreading*. Sedangkan J2ME (Java 2 Micro Edition) dapat menghasilkan aplikasi *mobile* baik *games* maupun *software* yang dapat dijalankan di peralatan *mobile* seperti ponsel. (Widianto; 2011 :).

II.3.2. Java Virtual Machine (JVM)

Java Virtual Machine (JVM) adalah sebuah mesin *imajiner* (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. *Java Virtual Machine* (JVM) menyediakan *spesifikasi hardware* dan *platform* dimana kompilasi kode Java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis Java menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh *Java Virtual Machine* (JVM).

Aplikasi program Java diciptakan dengan *file* teks berekstensi *.java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi (Jakaria, 2010, *Pengenalan Java*).

II.4. Aplikasi Eclipse

Menurut *Widianto Pratama (Tutorial Android Programming, 2011)* Eclipse merupakan komunitas *open source* yang bertujuan menghasilkan *platform* pemrograman terbuka. Eclipse terdiri dari *framework* yang dapat dikembangkan

lebih lanjut, peralatan bantu untuk membuat dan memmanage *software* sejak awal hingga diluncurkan. *Platform* Eclipse didukung oleh ekosistem besar yang terdiri dari vendor teknologi, *start-up inovatif*, universitas, riset institusi serta individu.

Banyak orang mengenal Eclipse sebagai IDE (*Integrated Development Environment*) untuk bahasa Java, tapi Eclipse lebih dari sekedar IDE untuk Java. Komunitas Eclipse memiliki lebih dari 60 proyek *open source*. Proyek-proyek ini secara konsep terbagi menjadi 7 kategori :

1. Enterprise Development
2. Embedded and Device Development
3. Rich Client Platform
4. Rich Internet Applications
5. Application Frameworks
6. Application Lifecycle Management (ALM)
7. Service Oriented Architecture (SOA)

Secara umum Eclipse digunakan untuk membangun *software inovatif* berstandar industri, dan alat bantu beserta *frameworknya* membantu pekerjaan menjadi lebih mudah (*Widianto Pratama, Tutorial Android Programming, 2011*).

II.4.1. Lisensi

Eclipse menggunakan EPL (*Eclipse Public License*), yaitu lisensi yang memungkinkan organisasi untuk menjadikan Eclipse sebagai produk komersialnya, dan pada saat yang sama meminta orang yang melakukan perubahan untuk mengkontribusikan hasilnya kembali kepada komunitas.

II.5. Sistem Operasi Android

Android adalah sistem operasi mobile yang berjalan pada kernel **Linux**, yang dirilis pada 21 Oktober 2008. Awalnya, sistem operasi ini dikembangkan oleh Android, Inc, yang kemudian dibeli oleh *Google*, dan yang terakhir, sistem operasi ini dibeli oleh *Open Handset Alliance*, sebuah consortium dari 47 perusahaan *hardware*, *software*, dan *telecom* yang didirikan untuk membuat open standard bagi perangkat lunak mobile. Sistem operasi ini bersifat *free* dan *open source*. Perangkat mobile yang mendukung sistem operasi ini di antaranya adalah HTC Dream dan HTC Magic, ponsel keluaran vendor asal Taiwan, HTC.

Awalnya, *Google* membeli Android, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, HTC, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*. *Google* sendiri ternyata mempunyai alasan cukup kuat untuk melirik pangsa ini, karena perkembangan teknologi telepon seluler dewasa ini sudah bukan merupakan evolusi lagi, melainkan sebuah revolusi. Babak baru dalam dunia telekomunikasi *nirkabel* ini terus bergulir dengan cepat. Jika sekarang seseorang mempunyai PC di rumah, dan notebook untuk ke kantor atau kuliah, serta berkomunikasi melalui telepon seluler. Maka pergerakan yang kemudian terjadi sekarang adalah, orang mulai berpikir bagaimana menyatukan semuanya dalam satu gengaman.

Sebenarnya hal tersebut telah mulai dipenuhi dengan munculnya PDA/*smartphone*, di mana seseorang dapat merangkum semua kebutuhan

komputasinya dalam satu genggam. Dan perkembangan inilah yang membuat *Google* berambisi untuk menguasai pangsa ini. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Adapun Fitur-fitur Android adalah sebagai berikut :

- a. *Framework Aplikasi* yang mendukung penggantian komponen dan *reusable*.
- b. Mesin *virtual Dalvik* dioptimalkan untuk perangkat *mobile*.
- c. *Integrated browser* berdasarkan *engine open source WebKit*.
- d. Grafis yang dioptimalkan dan didukung oleh perpustakaan grafis 2D, grafis 3D berdasarkan spesifikasi *opengl ES 1,0* (Opsional akselerasi hardware).
- e. *SQLite* untuk penyimpanan data.
- f. *Media Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- g. *GSM Telephony* (tergantung hardware).
- h. *Bluetooth, EDGE, 3G, dan WiFi* (tergantung hardware).
- i. *Kamera, GPS, kompas, dan accelerometer* (hardware tergantung).

II.5.1. Kelebihan Android

Adapun kelebihan Android adalah sebagai berikut ini (*Widianto Pratama, Tutorial Android Programming, 2011*)

1. *Lengkap (Complete Platform)* : Android dikatakan lengkap karena Android menyediakan tools untuk membangun software yang sangat lengkap dibanding dengan platform lain. Para pengembang dapat melakukan

pendekatan yang komprehensif ketika mereka mengembangkan suatu aplikasi pada platform Android.

2. Terbuka (*Open Source Platform*) : Platform Android diciptakan dibawah lisensi *open source*, dimana para pengembang bebas untuk mengembangkan aplikasi pada platform ini. Android menggunakan Linux kernel 2.6.
3. Bebas (*Free Platform*) : Android adalah *platform mobile* yang tidak memiliki batasan dalam mengembangkan aplikasinya. Tidak ada lisensi dalam mengembangkan aplikasi Android. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.
4. Dapat menulis file di SD card.
5. Dapat mengolah database dengan SQLite.
6. *Application framework* berbasis komponen yang memudahkan reuse.
7. *Dalvik virtual machine* dioptimisasi untuk menjalankan program Java di mobile devices
8. *Integrated browser* berbasis WebKit engine
9. *Optimized graphics* tersedia 2D graphics library; dan OpenGL ES 1.0 untuk 3D graphics
10. Media support untuk MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF
11. Support pada GSM Telephony fasilitas telepon.
12. Mendukung menggunakan jaringan Bluetooth, EDGE, 3G, dan WiFi
13. Mendukung penggunaan *device* Kamera, GPS, *compass* dan *accelerometer*
14. Mendukung Multitouch .

II.6. UML

II.6.1. Pengenalan UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang memandu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP).

UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem yang berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar CORBA (*Common Object Request Broker Architecture*). (Martin Fowler, 2005 : 1)

II.6.2. Diagram-Diagram UML

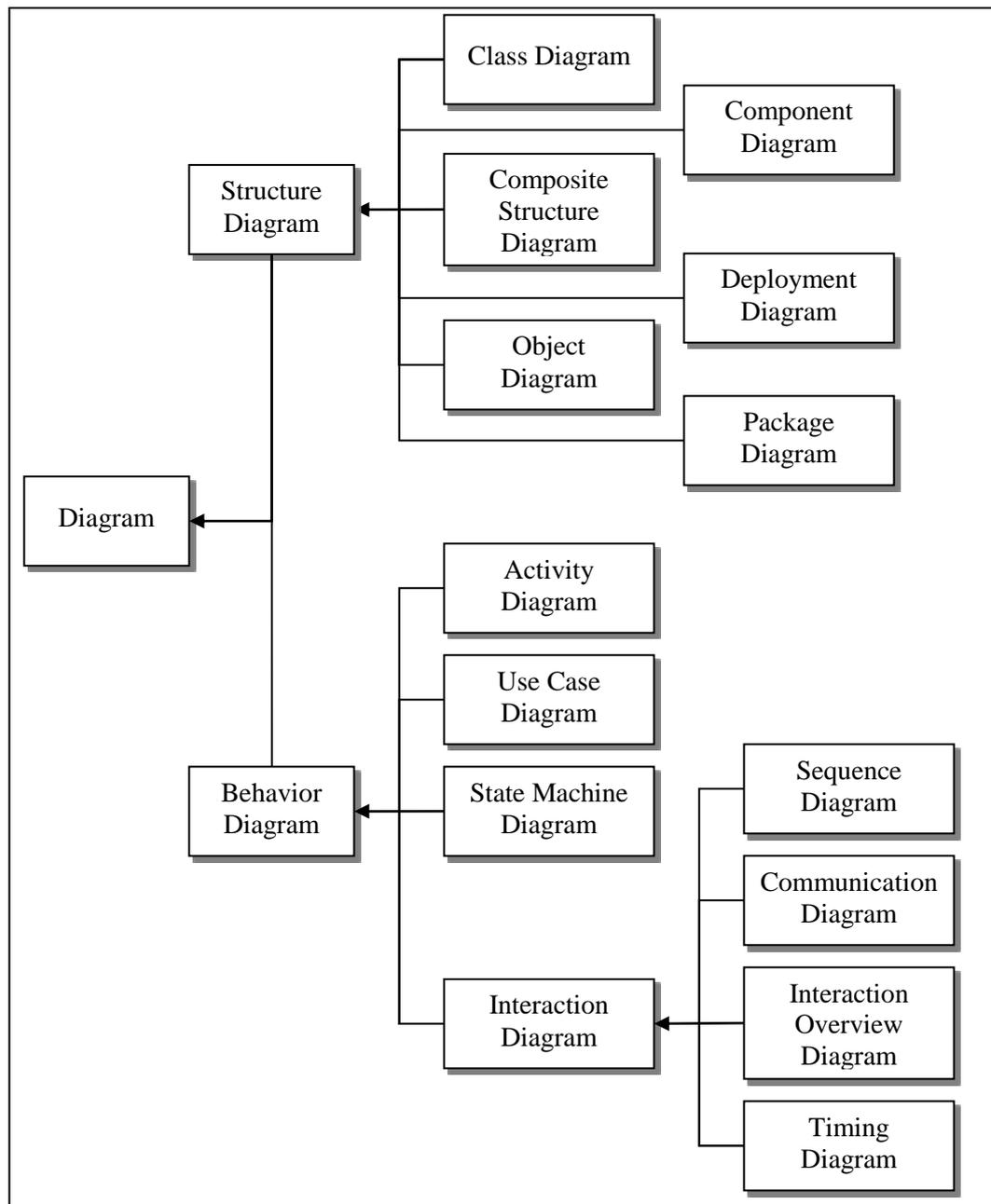
UML terdiri dari diagram, notasi, konsep dan aturan yang digunakan dalam memodelkan sistem. Diagram UML terdiri dari 13 jenis diagram yang memiliki fungsi dan notasi masing-masing. Kesembilan diagram ini dapat dibagi menjadi 2 kategori, yaitu diagram yang menggambarkan struktur yang statis dari sistem dan diagram yang menggambarkan struktur yang dinamis dari sistem.

1. Struktur Diagram

Merupakan diagram yang menggambarkan struktur hubungan statis dari elemen-elemen yang ada dalam sebuah model diantaranya *class*, *package*, dan *relationship* yang terjadi. (Martin Fowler, 2005 : 17)

2. Behavior Diagram

Merupakan kumpulan diagram yang menggambarkan hubungan dinamis antara class yang berada dalam komponen model.



Gambar II.3. Klasifikasi Jenis Diagram UML

Sumber : Martin Fowler (2005:19)

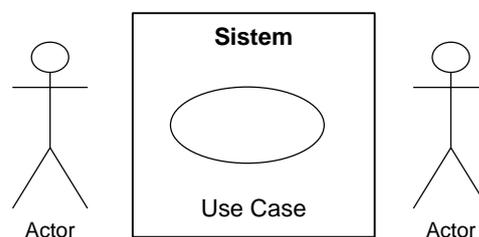
II.6.3. Karakter Diagram UML

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu.

1. Use Case Diagram.

Use case adalah abstraksi dari interaksi antara sistem dan actor. Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Use case merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata user. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

Diagram use case menunjukkan 3 aspek yaitu: aktor, use case dan sistem / sub sistem boundary. Actor mewakili peran orang, sytem yang lain atau laot ketika berkomunikasi dengan use case. Gambar berikut mengilustrasikan aktor, use case dan boundary.



Gambar II.4. Use Case Model

(Sumber : Munawar, 2005:64)

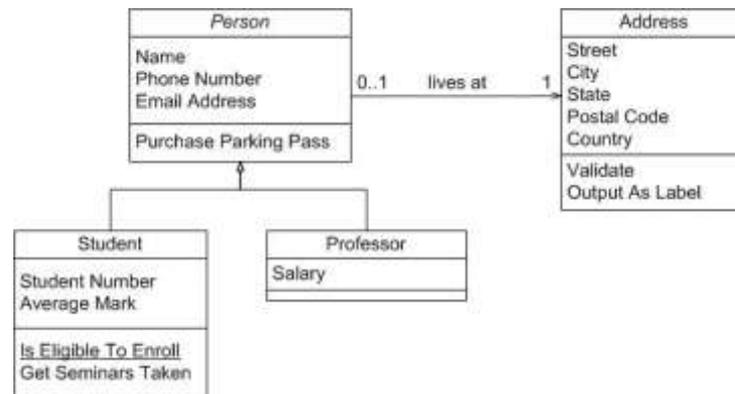
Tabel II.1. Komponen Use Case Diagram

No.	Nama Komponen	Keterangan	Simbol
1	Actor	Actor juga dapat berkomunikasi dengan object , maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram.	
2	Association	Association Asosiasi digunakan untuk menghubungkan actor dengan use case. Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara Actor dengan use case	
3	System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.	
4	Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor	

Sumber : Grady Booch (2005 : 30)

2. Class Diagram.

Class adalah dekripsi kelompok obyek-obyek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah system. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa class diagram. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem.



Gambar II.5. Class diagram

(Sumber : Munawar, 2005:221)

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok, yaitu :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Berikut ini merupakan komponen-komponen pada class diagram.

Tabel II.2. Komponen Class Diagram

No.	Nama Komponen	Keterangan	Simbol
1	Class	Class Class adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah class	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Nama Class</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">+ atribut</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">+ atribut</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">+ atribut</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">+ method</div> <div style="border: 1px solid black; padding: 5px;">+ method</div>

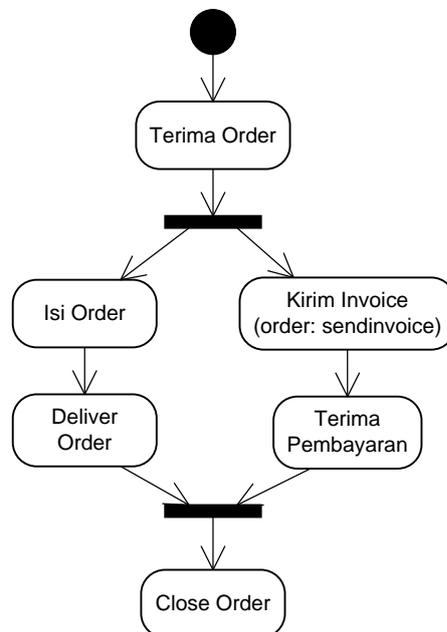
		digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari class. Bagian tengah mendefinisikan property/atribut class. Bagian akhir mendefinisikan method/method dari sebuah class.	
2	Association	Association Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 class dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship. (Contoh: One-to-one, one-to-many, many-to-many).	<u>1.n</u> Owred by <u>1</u>
3	composition	Composition Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi Composition terhadap class tempat dia bergantung tersebut. Sebuah relationship composition digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.	
4	Dependency	Dependency Kadangkala sebuah class menggunakan class yang lain. Hal ini disebut dependency. Umumnya	

		penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain. Sebuah dependency dilambangkan Sebagai panah bertitik-titik.	
5.	Aggregation	Aggregation mengindikasikan keseluruhan bagian relationship dan biasa disebut relasi	

Sumber : Grady Booch (2005 : 34)

3. Activity Diagram.

Menggambaran rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi.

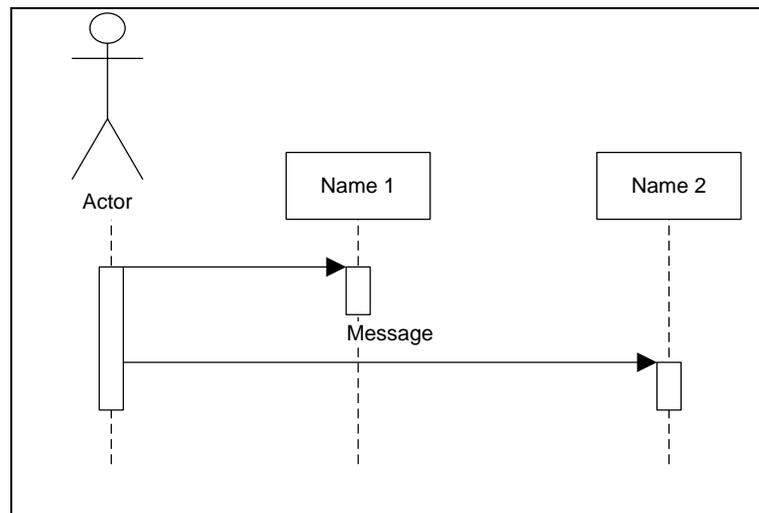


Gambar II.6. Activity diagram modifikasi

(Sumber : Munawar, 2005:112)

4. Sequence Diagram.

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antaraobject, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.



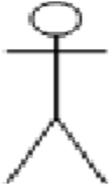
Gambar II.7. Simbol-simbol yang ada pada sequence diagram

(Sumber : Munawar, 2005:89)

Berikut ini merupakan komponen-komponen pada sequence diagram.

Tabel II.3. Komponen Sequence Diagram

No.	Nama Komponen	Keterangan	Simbol
1	Object	Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama object didalamnya yang diawali dengan sebuah titik koma.	

2	Actor	Actor juga dapat berkomunikasi dengan object , maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram.	
---	-------	--	---

Sumber : Grady Booch (2005 : 42)