BAB II

LANDASAN TEORI

2.1. Pengertian Perangkat Lunak (*Software*)

Menurut Hariyanto Bambang (2004:3) perangkat lunak adalah produk yang seharusnya dirancang dan dibangun lewat aktivitas-aktivitas rekayasa perangkat lunak yang berdisiplin dan sistematis. Perangkat lunak dalam arti lebih luas terdiri dari program-program yang dieksekusi computer dalam beraneka ukuran dan beraneka ragam arsitektur, dokumen-dokumen berupa *hard-copy* dan bentuk-bentuk maya, dan data berupa angka-angka dan teks juga representasi informasi gambar, video dan audio.

Sedangkan menurut Pressman (2012 : 5) perangkat lunak adalah : (1) instruksi-instruksi (program computer) yang ketika dijalankan menyediakan fitur-fitur; (2) struktur data yang memungkinkan program-program memanipulasi informasi, dan (3) informasi deskriptif pada salinan tercetak dan bentuk-bentuk maya yang menggambarkan pengoperasian dan penggunaan program-program.

2.1.1. Kategori Perangkat Lunak

Menurut Ian Sommervile :

Kategori rpoduk-produk perangkat lunak adalah :

1. Produk-produk generic,
2. Produk-produk yang disesuaikan. Hariyanto Bambang (2004 : 4)

Menurut Pressman :

Perangkat lunak terbagi menjadi

1. Perangkat lunak sistem, kumpulan program untuk melayani program-program yang lain. Contoh perangkat lunak sistem antara lainsistem operasi, kompilator, editor, utilitas-utilitas pengelolaan file, *device driver*, pemroses telekomunikasi. Perangkat lunak ini dicirikan interaksi yang tinggi dengan perangkat keras.
2. Perangkat lunak waktu nyata, perangkat lunak yang memonitor/menganalisis/mengendalikan kejadian-kejadian dunia nyata disaat terjadinya. Komponen perangkat lunak waktu nyata antara lain komponen yang mengumpulkan informasi dari lingkungan eksternal, komponen analisis yang mentransformasikan informasi ke format yang diperlukan aplikasi dan komponen pemantauan *(monitoring)* yang mengkoordinasikan semua komponen lain sehingga dapat member tanggapan secara nyata.
3. Perangkat lunak rekayasa dan sains, astronomi, volkanologi, biologi molekuler sampai manufaktur, CAD, simulasi sistem dan sebagainya. Perangkat lunak tempelan *(embedded software)*, produk-produk intelejen telah menjadi biasa di setiap pasar consumer dan industri. Perangkat lunak embedded berada di ROM *(read only memory)* dan digunakan untuk mengendalikan produk-produk dan sistem-sistem untuk pasar consumer dan industri. Perangkat lunak embedded dapat melakukan fungsi-fungsi sangat terbatas dan esoterik (seperti kendali keypad untuk oven microwave) atau menyediakan fungsi signifikan

dan kemampuan kendali (seperti fungsi-fungsi digital di otomobil seperti kendali bahan bakar, tampilan dashboard, dan sistem pengereman).

4. Perangkat lunak berbasis web, halaman-halaman web diambil oleh browser adalah perangkat lunka yang bekerjasama (berisi) dengan instruksi-instruksi yang dapat dieksekusi (misalnya CGI, HTML, Perl, atau Java) dan data (seperti hypertext dan beragam format visual dan audio), jaringan yang menjadi computer massif yang menyediakan sumber daya pernagkat lunak hamper tidak terbatas yang dapat diakses oleh siapapun melalui modem.
5. Perangkat lunak intelejensia buatan, perangkat lunak AI menggunakan algoritma-algoritma non-numerik untuk menyelesaikan persoalan-persoalan kompleks yang sulit dilakukan analisis secara komputasi atau sederhana. Sistem pakar, juga disebut sistem berbasis pengetahuan, pengenalan pola (citra dan suara), jaringan syaraf tiruan, membukitian teorema, dan *game playing* adalah representasi aplikasi-aplikasi dalam katergori ini. Hariyanto Bambang (2004 : 4)

2.1.2. Software Engineering (Rekayasa Perangkat Lunak)

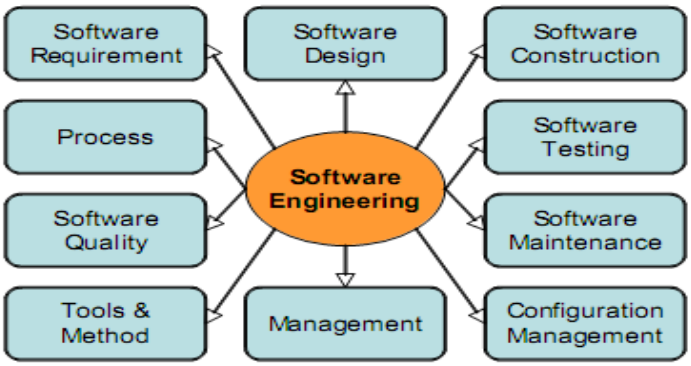
Istilah Reakayasa Perangkat Lunak (RPL) secara umum disepakati sebagai terjemahan dari istilah Software engineering. Istilah Software Engineering mulai dipopulerkan pada tahun 1968 pada software engineering Conference yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (software) dan program komputer.

Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (O'Brien, 1999).

Pengertian RPL sendiri adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dari pengertian ini jelaslah bahwa RPL tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan "semua aspek produksi" pada pengertian di atas, mempunyai arti semnua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari RPL.

2.1.2.1Ruang Lingkup Software Engineering (Rekayasa Perangkat Lunak)

Sesuai dengan definisi yang telah disampaikan sebelumnya, maka ruang lingkup Software Engineering (RPL) dapat digambarkan sebagai berikut:



Gambar 2.1 Ruang lingkup RPL (Abran et.al., 2004).

a. Software Requirements berhubungan dengan spesifikasi kebutuhan dan persyaratan perangkat lunak
b. Software desain mencakup proses penampilan arsitektur, komponen, antar muka, dan karakteristik lain dari perangkat lunak
c. Software construction berhubungan dengan detail pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan
d. Software testing meliputi pengujian pada keseluruhan perilaku perangkat lunak
e. Software maintenance mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan
f. Software configuration management berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu
g. Software engineering management berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak
h. Software engineering tools and methods mencakup kajian teoritis tentang alat bantu dan metode RPL
i. Software engineering process berhubungan dengan definisi, implementasi pengukuran, pengelolaan, perubahan dan perbaikan proses RPL
j. Software quality menitik beratkan pada kualitas dan daur hidup perangkat lunak

2.1.2.2. Tahapan Software Engineering (Rekayasa Perangkat Lunak)

Meskipun dalam pendekatan berbeda-beda, namun model-model pendekatan memiliki kesamaan, yaitu menggunaka pola tahapan analysis – design – coding(construction) – testing – maintenance.

1. **Analisis sistem** adalah sebuah teknik pemecahan masalah yang menguraikan sebuah sistem menjadi komponen-komponennya dengan tujuan mempelajari seberapa bagus komponen-komponen tersebut bekerja dan berinteraksi untuk meraih tujuan mereka.

   Analisis mungkin adalah bagian terpenting dari proses rekayasa perangkat lunak. Karena semua proses lanjutan akan sangat bergantung pada baik tidaknya hasil analisis. Ada satu bagian penting yang biasanya dilakukan dalam tahapan analisis yaitu pemodelan proses bisnis.

2. **Model proses** adalah model yang memfokuskan pada seluruh proses di dalam sistem yang mentransformasikan data menjadi informasi (Harris, 2003). Model proses juga menunjukkan aliran data yang masuk dan keluar pada suatu proses. Biasanya model ini digambarkan dalam bentuk Diagram Arus Data (Data Flow Diagram / DFD). DFD meyajikan gambaran apa yang manusia, proses dan prosedur lakukan untuk mentransformasi data menjadi informasi.
3. **Disain perangkat lunak** adalah tugas, tahapan atau aktivitas yang difokuskan pada spesifikasi detil dari solusi berbasis computer (Whitten et al, 2004).

   Disain perangkat lunak sering juga disebut sebagai physical design. Jika tahapan analisis sistem menekankan pada masalah bisnis (business rule), maka

sebaliknya disain perangkat lunak fokus pada sisi teknis dan implementasi sebuah perangkat lunak (Whitten et al, 2004).

Output utama dari tahapan disain perangkat lunak adalah spesifikasi disain. Spesifikasi ini meliputi spesifikasi disain umum yang akan disampaikan kepada stakeholder sistem dan spesifikasi disain rinci yang akan digunakan pada tahap implementasi. Spesifikasi disain umum hanya berisi gambaran umum agar stakeholder sistem mengerti akan seperti apa perangkat lunak yang akan dibangun. Biasanya diagram USD tentang perangkat lunak yang baru merupakan point penting dibagian ini. Spesifikasi disain rinci atau kadang disebut disain arsitektur rinci perangkat lunak diperlukan untuk merancang sistem sehingga memiliki konstruksi yang baik, proses pengolahan data yang tepat dan akurat, bernilai, memiliki aspek user friendly dan memiliki dasar-dasar untuk pengembangan selanjutnya.

Desain arsitektur ini terdiri dari desain database, desain proses, desain user interface yang mencakup desain input, output form dan report, desain hardware, software dan jaringan. Desain proses merupakan kelanjutan dari pemodelan proses yang dilakukan pada tahapan analisis.

4. **Konstruksi** adalah tahapan menerjemahkan hasil disain logis dan fisik ke dalam kode-kode program komputer.
5. **Pengujian** sistem melibatkan semua kelompok pengguna yang telah direncanakan pada tahap sebelumnya. Pengujian tingkat penerimaan terhadap perangkat lunak akan berakhir ketika dirasa semua kelompok pengguna

menyatakan bisa menerima perangkat lunak tersebut berdasarkan kriteria-kriteria yang telah ditetapkan.

6. **Perawatan dan Konfigurasi.** Ketika sebuah perangkat lunak telah dianggap layak untuk dijalankan, maka tahapan baru menjadi muncul yaitu perawatan perangkat lunak.

2.2. Pengelolaan

Pengelolaan diartikan sebagai suatu rangkaian pekerjaan atau usaha yang dilakukan oleh sekelompok orang untuk melakukan serangkaian kerja dalam mencapai tujuan tertentu.

Menurut Harsoyo (1977 : 121) yang saya kutip dari skripsi Dedi Rosadi (2011 : 26,27) pengelolaan adalah suatu istilah yang berasal dari kata “kelola” mengandung arti serangkaian usaha yang bertujuan untuk menggali dan memanfaatkan segala potensi yang dimiliki secara efektif dan efisien guna mencapai tujuan tertentu yang telah direncanakan sebelumnya. Sedangkan menurut Wardoyo (1980 : 41) pengelolaan adalah suatu rangkaian kegiatan yang berintikan perencanaan, pengorganisasian, penggerakan dan pengawasan yang bertujuan menggali dan memanfaatkan sumberdaya alam yang dimiliki secara efektif untuk mencapai tujuan organisasi yang telah ditentukan.

2.3. Konsep Dasar Informasi

Informasi adalah data yang telah diproses dan memiliki arti atau manfaat bagi penggunanya.Informasi sebagai hasil dari pengolahan data dalam suatu organisasi sangatlah penting Karen ainformasi merupakan landasan untuk mengambil suatu keputusan dan data merupakan sumber dari informasi.

Konsep dasar informasi menurut H.M Jogiyanto (2005 : 8) adalah data yang diolah menjadi bentuk yang lebih berguna dan penting bagi yang menerimanya. Dari pengertian diatas penulis dapat menyimpulkan bahwa informasi merupakan suatu (output) dari suatu data yang diolah dengan metode pendekatan dan pengembangan tertentu.

2.3.1. Definisi Informasi

Informasi merupakan salah satu unsure yang sangat penting di dalam suatu organisasi. Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, sehingga informasi tersebut sangat penting artinya bagi suatu organisasi.

Menurut H.M Jogiyanto (2005 : 8) informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.3.2. Kualitas Informasi

Menurut H.M Jogiyanto (2005 : 10) kualitas informasi terdiri dari 3 (tiga) hal, yaitu :

a. Akurat

   Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan, harus jelas mencerminkan maksudnya.Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

b. Tepat Waktu

Tepat waktu berarti informasi yang dating pada penerima tidak boleh terlambat. Informasi yang sudah usangn tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka akan berakibat fatal untuk organisasi.

c. Relevan

Relevan berarti informasi tersebut mempunyai manfaat untuk pemakaiannya.Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.3.3. Nilai Informasi

Menurut H.M Jogiyanto (2005 : 11), nilai dari informasi (*value of information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai apabila manfaatnya lebih afektif dibandingkan dengan biaya mendapatkannya. Kegunaan informasi adalah untuk mengurrangi hal ketidakpastian di dalam proses pengambilan keputusan tentang suatu keadaan. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah yang tertentu dengan biaya untuk memperolehnya.Sebagian besar informasi tidak dapat persis ditaksirkan keuntungannya dengan suatu nilai uang, tetapi dapat ditaksir dari nilai efektifitasnya.

2.4. Hasil Penelitian Matahari dan Antariksa

Menurut Kaloka.P Sri (2009 : 72) cuaca Antariksa adalah kondisi di matahari dan di angin surya, magnetosfer, ionosfer, dan termosfer yang dapat mempengaruhi kondisi dan kemampuan teknologi, baik yang landas bumi maupun ruang angkasa, dan membahayakan kehidupan kehidupan dan kesehataan manusia. Cuaca antariksa disebabkan adanya aktivitas matahari yang melontarkan milyaran ton partikel dan plasma berenergi tinggi serta radiasi gelombang elektro magnetik. Lontaran partikel dan radiasi yang mengarah ke bumi ini akan mempengaruhi lapisan atmosfer, serta system teknologi dan aktivitas manusia di bumi.

2.5. Sejarah *Android*

Android adalah sebuah system operasi untuk perangkat *mobile* berbasis linux ang mencakup system 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 mengembangan Android dibentuklah *Open Handset Aliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Aliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Di dunia ini terdapat dua jenis distributor system operasi Android. Pertama yang mendapat dukukang 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 system operasinya.Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, Android, pernagkat *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 sebagian besar *vendor-vendor smartphone* sudah memproduksi *smartphone* 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 smartphone* di dunia yang memproduksi android. Hal ini, karena android itu adalah system operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh *vendor* manapun. Tidak hanya menjadi system operasi di *smartphone*, saat ini Android menjadi pesaing utama dari Apple pada system operasi Tablet PC.

Android merupakan generasi baru *platform mobile,* yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android lisensikan sesuai dibawah GNU, General Public Lisensi Versi 2 (GPLv2), yang sering dikenal dengan istilah "copyleft".Android didistribusikan di bawah lisensi *Apache Software* (ASL/ Apache2). Pengembangan aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apa pun yang mereka inginkan. Pengembangan memiliki beberapa pilihan ketika membuat aplikasi yang berbasis android. Sebagian besar pengembangan menggunakan *Eclipse* yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi Android.*Eclipse* adalah IDE yang paling popular untuk pengembangan Android, karena memiliki Android *plug-in* yang tersedia untuk memfasilitasi pengembangan Android. Selain itu, *Eclipse* juga mendapat dukungan langsung dari Google untuk menjadi IDE pengembangan aplikasi Android, dengan adanya penambahan *plugin* untuk *eclipse* untuk membuat *project* android di mana *source software* langsung dari situs resminya Google. Hal diatas tidak menutup kemungkinan untuk menggunakan IDE yang lain seperti Netbeans untuk melakukan pengembngan android. Aplikasi Android dapat dikembangkan pada system operasi berikut :

1. Windows XP Vista/ Seven
2. Mac OS X (Mac OS X $_{10.4.8}$ atau lebih baru)
3. Linux

2.5.1. Arsitektur*Android*

Menurut Safaat H. Nazruddin (2012 : 6) secara garis besar arsitektur *android* dapat dijelaskan dan digambarkan sebagai berikut :

a. *Applications* dan *Widgets*

Applications 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 ditulis menggunakan bahasa pemograman Java.

b. *Aplications Framework*

Android adalah "Open development Platform" yaitu Android menawarkan kepada pengembang atau member kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inivatif. Pengembangan bebas untuk mengakses perangkat keras, akses informasi *resources,* menjalankan *service background,* mengatur *alarm,* dan menambah status *notifications,* dan sebagainya. Pengembang memiliki akses penuhmenuju API *framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunaka (*resue*).

Sehingga bisa kita simpulkan *Applications Frameworks* ini adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem

operasi Android, karena pada*layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon. Komponen-komponenn yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut :

a. *Views*
b. *Content Provider*
c. *Resource manager*
d. *Notification manager*
e. *Activity manager*

c. *Libraries*

Libraries ini adalah *layer* dimana fitur-fitur Android berbeda, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, *Layer* ini meliputi berbagai *library* C/C++ inti seperti Libc 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 *Webkit* 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

d. *Andoid run Time*

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Mechine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi Android. Didalam *Android Run Time* dibagi menjadi dua bagian, yaitu :

a. *Core Libraries* : Aplikasi android dibangun dalam bahasa java, sedangkan Dalvik sebagai virtual mesinnya bukan Virtual Mechine Java, sehingga diperlukan *libraries* yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh *Core Libraries.*

b. *Dalvik Virtual Mechine* : 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 tingkat rendah.

e. *Linux kernel*

Linux kernel adalah *layer* dimana inti dari *operating* sistem dari Android itu berada. Berisi *file-file system* yang mengatur sistem *processing, memory, resource, drivers,* dan system-sistem operasi android lainnya. Linuk kernel yang digunakan android adalah linux kernel *relase* 2.6.

2.5.2. Versi *Android*

Menurut Safaat H. Nazaruddin, telepon pertama yang memakai system operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2010 diperkirakan hamper semua *vendor* seluler didunia menggunakan Android sebagai *operating system.* Adapun versi-versi Android yang pernah dirilis, sebagai berikut :

a. *Android versi 1.1*
b. *Android versi 1.5 (Cupcake)*
c. *Android versi 1.6 (Donut)*
d. *Android versi 2.0/2.1 (Éclair)*
e. *Android versi 2.2 (Froyo : Frozen Yoghurt)*
f. *Android versi 2.3 (Gingerbread)*
g. *Android versi 3.0 (honeycomb)*
h. *Android versi 3.1*
i. *Android versi 3.2*
j. *Android versi 4.0*

Android versi 3.0 keatas adalah generasi *platform* yang digunakan untuk tablet PC. Sementara versi 4.0 sudah merupakan *platform* yang bisa digunakan di *smartphone* dan tablet pc.

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 pemrograman Java. Android merupakan subset perangkat

lunak untuk ponsel yang meliputi system operasi, *middleware* dan aplikasi kunci yang di-*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 pemrograman Java. Sebagai *platform* aplikasi-netral, Android memberikan Anda kesempatan untuk membuat Aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone / Smartphone.* Beberapa fitur-fitur android yang paling penting adalah :

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 libraries 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, GIF), GSM *Telephony (*tergantung hardware).
g. *Bluetooth,* EDGE, 3G, dan Wifi (tergantung hardware).
h. Kamera, GPS, kompas, dan *accelerometer* (tergantung hardware).
i. Lingkungan *development* yang lengkap dan kaya termasuk perangkat *emulator, tools* untuk *debugging,* profil dan kinerja memori dan *plugin* untuk IDE *Eclips.*

2.7. Metode Pendekatan dan Pengembangan Sistem.

Untuk dapat melakukan langkah-langkah sesuai dengan yang diberikan oleh metodologi pengembangan sistem berorientasi objek, maka dibutuhkan alat untuk melaksanakannya. Alat-alat yang digunakan dalam suatu metodologi umumnya berupa suatu gambar, diagram, atau grafik.

2.7.1. Metode Pendekatan Sistem

Merupakan bagan alir data yang menunjukkan arus dari dokumen-dokumen yang berupa laporan dan formulir termasuk tembusan tembusannya. Bagan alir ini menggunakan simbol – simbol yang sama dengan bagan alir sistem.

2.7.2. Alat Bantu Analisis

Alat – alat apa saja yang digunakan untuk membantu dalam menganalisis mengerjakan penelitian untuk skripsi, alat bantu analisis yang digunakan antara lain :

1. ***Use Case Diagram***

   Use case diagram dapat digunakan untuk menentukan kebutuhan apa saja yang diperlukan dari suatu sistem. Jadi dapat digambarkan dengan detail bagaimana suatu sistem memproses atau melakukan sesuatu, bagaimana cara actor akan menggunakan sistem, serta apa saja yang dapat dilakukan terhadap sistem.

2. Activity Diagram

Activity diagram digunakan untuk menganalisis behavior dengan use case yang lebih kompleks dan menunjukkan interaksi-interaksi di antara satu dengan yang lainnya.

3. Class Diagram

Diagram ini menggambarkan perbedaan yang mendasar diantara class-class, hubungan antar class, dan dimana sub-sistem class tersebut. Pada class diagram terdapat nama class, attributes, operations, serta association (hubungan antar class).

4. *SequenceDiagram*

Sequence Diagram secara khusus menjabarkan behavior sebuah scenario tunggal. *Sequence Diagram* menunjukan sebuah objek contoh dan pesan-pesan yang melewati objek dalam *Use Case*.

5. *Component Diagram*

Diagram komponen mendefinisikan masing – masing modul (*component*) yang membentuk arsitektur sistem atau perangkat lunak.

6. *Deployment Diagram*

Peringkat penebaran-komponen (*deployment*) melakukan alokasi terhadap arsitektur sistem atau perangkat lunak, terhadap komponen – komponen di dalamnya serta terhadap antarmuka pada konfigurasi fisik yang akan ditempati oleh perangkat lunak.