BAB II

LANDASAN TEORI

Landasan teori diperoleh dari studi literatur. Studi literatur diperlukan untuk mengeksplorasi teori – teori yang diperlukan dalam menyelesaikan tugas akhir ini. Fungsi dari teori adalah sebagai alat dalam mencapai satuan pengetahuan yang sistematis dan menjadi pembimbing bagi penulis dalam melakukan penelitian.

2.1 Kehamilan

Kehamilan manusia terjadi selama 40 minggu antara waktu menstruasi terakhir dan kelahiran (38 minggu dari pembuahan). Istilah medis untuk wanita hamil adalah gravida, sedangkan manusia di dalamnya disebut embrio (minggu – minggu awal) dan kemudian janin (sampai kelahiran). Seorang wanita yang hamil untuk pertama kalinya disebut primigravida atau gravid I. Seorang wanita yang belum pernah hamil dikenal sebagai gravida 0. Dalam banyak masyarakat. definisi medis dan legal kehamilan manusia dibagi menjadi tiga periode triwulan, sebagai cara memudahkan tahap berbeda dari perkembangan janin. Triwulan pertama membawa resiko tertinggi keguguran (kematian alami embrio atau janin), sedangkan pada masa triwulan ke-2 perkembangan janin dapat dimonitor dan didiagnosa. Triwulan ke-3 menandakan awal ‘viabilitas’, yang berarti janin dapat tetap hidup bila terjadi kelahiran awal atau kelahiran dipaksakan. Karena kemungkinan viabilitas janin yang telah berkembang,

definisi budaya dan legal dari hidup seringkali menganggap janin dalam triwulan ke-3 adalah sebuah pribadi hidup yang baru.

2.1.1 Definisi Kehamilan

Definisi kehamilan menurut dr. Widiyastuti HQD, SpOG, M. Kes., M.Hkes adalah peristiwa istimewa setiap wanita sebagai anugerah perwujudan dari impian dan kebahagiaan.

Pada kehamilan terjadi perubahan fisik dan mental sehingga penting memahami perubahan – perubahan yang terjadi pada tubuh ibu untuk menghindari kekhawatiran dan kecemasan yang berlebihan dan ibu dapat menjalani masa kehamilan dengan nyaman dan bahagia.

2.1.2 Tahap – Tahap Kehamilan

Kehamilan terbagi menjadi tiga tahap atau trisemester yaitu trisemester pertama (dari hari pertama menstruasi terakhir sampai minggu ke-14), trisemester kedua (minggu ke-14 sampai minggu ke-28) dan trisemester ketiga (minggu ke-28 sampai kelahiran).

1. **Trisemester Pertama**

   a. Perkembangan Ibu

      Tubuh terasa lelah akibat adanya perubahan hormon dalam indung telur dan peningkatan produksi hormon *pogesteron*. Pembentukan plasenta menutupi 1/3 bagian dalam uterus. Berat ibu bertambah 1,5 kg atau rata –rata 300 gr perminggu.

b. Perkembangan Janin

Embrio awal terbagi menjadi lapisan – lapisan sel yang akan membentuk berbagai organ dan jaringan ditubuh bayi. Pada minggu ke-3 dan ke-4 mulai terbentuknya jantung, peredaran darah dan saluran pencernaan. Hingga minggu ke-14 sebagian organ – organ tubuh telah berfungsi termasuk sistem syaraf, sistem otot semakin kuat dan pembuluh darah mulai berkembang.

2. Trisemester Kedua

a. Perkembangan Ibu

Plasenta mulai memproduksi hormon *esterogen*, memanjang dengan ukuran relatif dan menjadi lebih tebal. Ibu akan mulai merasakan gerakan janin. Kontraksi mulai terasa meski tidak sakit tapi perut akan terasa kaku.

b. Perkembangan Janin

Seluruh organ dan struktur tubuh telah terbenuk. Tumbuh alis, bulu mata rambut. Janin mulai menendang, menghisap dan menggeliat. Kelopak mata mulai membuka dan menutup. Janin bisa bernafas dan mengatur suhu tubuhnya. Pada minggu ke-27 terbentuk zat surfactant dalam paru – paru bayi. Detak jantung dapat didengarkan menggunakan Doopler. Pada minggu ke-28 berat badan janin 2/3 dari ukuran bayi saat lahir.

3. Trimester Ketiga

a. Perkembangan Ibu

Kelahiran semakin dekat, posisi janin semakin ke bawah. Kadang – kadang mengalami gangguan pernafasan. Colostrum mulai keluar. Periksakan kehamilan ibu ke bidan atau dokter sesuai jadwal.

b. Perkembangan Janin

Cadangan lemak terbentuk dibawah kulit. Seluruh bagian uterus atau rahim terisi oleh janin. Saat kelahiran berat janin antara 2,5 – 4 kg dengan panjang 47 – 53 cm.

2.2 Metode Penelitian

Metode penelitian menurut Sugiyono (2011 : 3), pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Data yang diperoleh melalui penelitian itu adalah data empiris (teramati) yang mempunyai kriteria tertentu yaitu valid, reliable, dan obyektif.

Metode penelitian ini merupakan bagian dari metodologi yang secara khusus mendeskripsikan tentang cara mengumpulkan dan menganalisis data. Setiap penelitian mempunyai tujuan dan kegunaan tertentu. Secara umum tujuan penelitian ada tiga macam yaitu yang bersifat penemuan, pembuktian, dan pengembangan. Dalam upaya mencapai tujuan dari sebuah penelitian diperlukan metode yang sesuai dan mudah.

Menurut Sugiyono (2011 : 29), bahwa berdasarkan tingkat eksplanasinya terdapat 3 macam metode penelitian yaitu sebagai berikut :

a. Penelitian Deskriptif

Penelitian deskriptif adalah suatu metode yang digunakan untuk menggambarkan atau menganalisis suatu hasil penelitian tetapi tidak digunakan untuk membuat kesimpulan yang lebih luas.

b. Penelitian Komparatif

Penelitian komparatif adalah suatu penelitian yang bersifat membandingkan. Disini variabelnya masih sama dengan variabel mandiri tetapi untuk sample yang lebih dari satu, atau dalam waktu yang berbeda.

c. Penelitian Asosiatif

Penelitian asosiatif merupakan penelitian yang bertujuan untuk mengetahui pengaruh ataupun juga hubungan antara dua variabel atau lebih. Penelitian ini mempunyai tingkatan tertinggi dibandingkan dengan diskriptif dan komparatif karena dengan penelitian ini dapat dibangun suatu teori yang dapat berfungsi unguk menjelaskan, meramalkan dan mengontrol suatu gejala.

2.3 Desain Penelitian

Menurut Moh. Nazir (2003 : 84) memaparkan bahwa desain penelitian adalah semua proses yang diperlukan dalam perencanaan dan pelaksanaan penelitian.

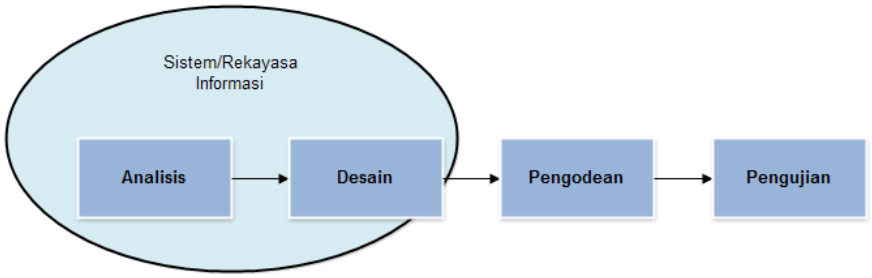
Keseluruhan dari perencanaan bertujuan untuk menjawab pertanyaan penelitian dan mengantisipasi beberapa kesulitan yang mungkin timbul selama proses penelitian, hal ini penting karena desain penelitian merupakan strategi untuk mendapatkan data yang dibutuhkan untuk keperluan pengujian hipotesis atau untuk menjawab pertanyaan penelitian dan sebagai alat untuk mengontrol variabel yang berpengaruh dalam penelitian.

2.4 Model-Model Pengembangan Sistem

Terdapat beberapa metode dalam melakukan pengembangan sistem, namun dalam hal ini hanya akan membahas dua metode yang mendekati dalam melakukan pengembangan sistem. Adapun dua metode yang mendekati yaitu sebagai berikut :

2.4.1 Model *Waterfall*

Model *waterfall* sering juga disebut sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model waterfall menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).



Gambar 2.2 Model *Waterfall*
(Sumber : **Rosa A.S – M.Shalahuddin.** 2011. *Rekayasa Perangkat Lunak (Terstuktur dan Berorientasi Objek)*. Modula. Bandung)

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga didokumentasikan.

c. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian focus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

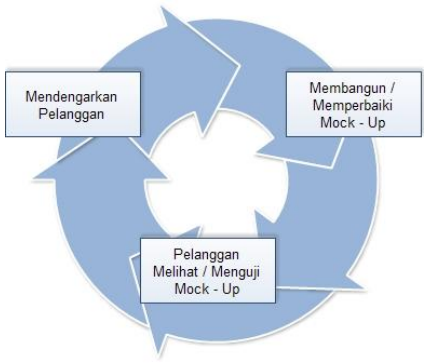
Dari kenyataannya yang terjadi sangat jarang model *waterfall* dapat dilakukan sesuai alurnya karena :

a. Perubahan spesifikasi perangkat lunak terjadi di tengah alur pengembangan.
b. Sangat sulit bagi pelanggan untuk mendefinisikan semua spesifikasi di awal alur pengembangan. Pelanggan sering kali butuh contoh untuk menjabarkan spesifikasi kebutuhan sistem lebih lanjut.
c. Pelanggan tidak mungkin bersabar mengakomodasi perubahan yang diperlukan di akhir alur perkembangan.

Dengan berbagai kelemahan yang dimiliki model *waterfall* tapi model ini telah menjadi dasar dari model-model lain yang dalam melakukan perbaikan model pengembangan perangkat lunak.

2.4.2 Model Prototype

Model prototipe dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak.



Gambar 2.3 Model *Prototype*
(Sumber : **Rosa A.S – M.Shalahuddin.** 2011. *Rekayasa Perangkat Lunak (Terstuktur dan Berorientasi Objek)*. Modula. Bandung)

Model prototipe dimulai dari mengumpulkan kebutuhan pelanggan tehadap perangkat lunak yang akan dibuat. Lalu dibuatlan program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangat lunak yang sudah jadi. Program prototipe ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau user.

Model prototipe juga memiliki kelemahan sebagai berikut :

a. Pelanggan dapat sering mengubah-ubah ata menambah sepesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan, karena adanya

iterasi ini dapat menyebabkan pengembang banyak mengalah dengan pelanggan karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.

b. Pengembang lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan prototipe dengan waktu yang cepat sehingga pengembangan lebih sering melakukan segala cara (tanpa idealis) guna menghasilkan prototipe untuk didemonstrasikan. Hal ini dapat menyebabkan kualitas perangkat lunak yang kurang baik atau bahkan menyebabkan iterasi tanpa akhir.

Model prototipe cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena pelanggan sering kali kesulitan menyampaikan kebutuhannya secara detail tanpa melihat gambaran yang jelas. Utnuk mengantisipasi agar proyek dapat berjalan sesuai dengan target waktu dan biaya di awal, maka sebaiknya spesifikasi kebutuhan sistem harus sudah disepakati oleh pengembang dengan pelanggan secara tertulis. Dokumen tersebut akan menjadi patokan agar spesifikasi kebutuhan sistem masih dalam ruang lingkup proyek.

2.5 Metode Pendekatan

Pendekatan sistem dapat diartikan sebagai titik tolak atau sudut pandang kita terhadap proses sebuah sistem yang merujuk pada pandangan tentang terjadinya suatu proses yang bersifat masih sangat umum dalam mewadahi, mengispirasi, dan melatari metode pendekatan dalam cakupan teoritis tertentu.

Didalam melakukan penelitian ini, penulis menggunakan metode pendekatan sistem berorientasi objek, namun ada baiknya apabila kita melakukan perbandingan terlebih dahulu antara pendekatan terstruktur dan pendekatan berorientasi objek.

2.5.1 Pendekatan Terstruktur

Pada pendekatan terstruktur ini permasalaha dilihat sebagai urutan sesuatu yang harus dikerjakan, seperti menerima masukan, pemrosesan, kemudian menghasilkan keluaran. Apa yang dilakukan pada pendekatan terstuktur terutama adalah menulis daftar perintah yang harus diikuti oleh komputer, kemudian mengorganisasikan perintah-perintah tersebut kedalam kelompok-kelompok yang dinamakan fungsi / prosedur.

Kelemahan pendekatan terstuktur adalah data global, pada program yang berukuran sangat besar sangat sulit untuk dilacak. Selain itu kelemahannya yaitu tidak menggambarkan dunia nyata (*real world*) dengan baik. Sebabnya adalah karena fungsi-fungsi berorientasi pada aksi dan tidak terhubung langsung dengan permasalahan. Pendekatan ini lebih memungkinkan perangkat lunak sesuai dengan keinginan pengembang, bukan sesuai dengan kebutuhan/keinginan pengguna.

Sifat- sifat dari pemrograman terstruktur adalah sebagai berikut :

1. Memuat teknik pemecahan masalah yang logis dan sistematis.
2. Memuat algoritma yang efesien, efektif dan sederhana.
3. Program disusun dengan logika yang mudah dipahami.
4. Tidak menggunakan perintah *GOTO*.

5. Biaya pengujian program relatif rendah.
6. Memiliki dokumentasi yang baik.
7. Biaya perawatan dan dokumentasi yang dibutuhkan relatif rendah

2.5.2 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perngkat lunak, sistem informasi, atau sistem lainnya). Pendekatan ini akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, dan komponen lainnya dan dapat berinteraksi satu sama lain.

Suatu program dapat dikatakan program berbasis objek (OOP), karena terdapat :

1. *Encapsulation* (Pembungkusan)

Pembungkusan adalah mekanisme pemrograman yang membungkus kode dan data yang dimanipulasi dan menjaganya supaya terhindar dari interferensi dan penggunaan yang tidak perlu. Salah satu caranya dengan membentuk objek.

2. *Inheritance* (Pewarisan)

Pewarisan memungkinkan programer meletakkan objek yang sama dalam satu kelas dan kelas-kelas lain dapat mewarisi objek tersebut. kelas yang mengandung objek yang sama dari beberapa kelas lain dinamakan *super class* atau *parent class*. Kelas yang mewarisi dinamakan *sub class* atau *child class*. Pewarisan menghasilkan kelas hirarki.

3. *Polymorphism* (Polimorfisme-Perbedaan Bentuk)

Polimorfisme artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai Polimorfisme, bila objek-objek itu mempunyai antar muka yang identik namun mempunyai perilaku-perilaku yang berbeda.

Menurut Adi Nugroho (2005 : 30), alat-alat yang digunakan dalam pendekatan analisis dan pemrograman berorientasi objek yaitu dengan notasi UML dengan membuat tujuh diagram yaitu:

a. *Use case diagram*

Diagram ini bersifat statis dimana diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

b. *Activity diagram*

Diagram ini bersifat dinamis dengan memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam

pemodelan fungsi-fungsi dalam suatu sistem dan member tekanan pada aliran kendali antar objek.

c. *Sequence diagram*

Diagram ini bersifat dinamis, dalam interaksinya menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

d. *Collaboration diagram*

Diagram ini bersifat dinamis, dalam interaksinya menekankan pada organisasi structural dari objek-objek yang menerima serta mengirim pesan (*message*).

e. *Class diagram*

Diagram ini bersifat statis, yang memperlihatkan himpunan kelas-kelas, antarmuka, kolaborasi, serta relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis sering pula diagram kelas ini memuat kelas-kelas aktif.

f. *Component diagram*

Diagram ini bersifat statis, yang memperlihatkan organsisasi serta ketergantungan pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas, antarmuka, serta kolaborasi.

g. *Deployment diagram*.

Diagram ini bersifar statis, yang memperlihatkan konfigurasi saat aplikasi dijalankan (saat *run-time*). Diagram ini memuat simpul-simpul (*node*) beserta komponen-komponen yang ada didalamnya. *Deployment* diagram sangat erat hubungannya dengan diagram komponen dimana *deployment* diagram memuat satu arau lebih komponen.

2.6 Android

2.6.1 Sejarah Android

Pada tahun 2005 Google mengakuisisi Android Inc yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Yang kemudian pada tahun itu juga memulai membangun platform Android secara intensif. Kemudian pada tanggal 12 November 2007 Google bersama Open Handset Alliance(OHA) yaitu konsorsium perangkat mobile terbuka, merilis Google Android SDK, setelah mengumumkannya seminggu sebelumnya, dan sambutanya sangat luar biasa, hampir semua media berita tentang IT dan Programming memberitakan tentang dirilisnya Android SDK (*Software Development Kit*).

Google bersama dengan OHA merilis paket software SDK yang lengkap untuk mengembangkan aplikasi pada perangkat mobile yaitu : Sistem operasi, Middleware dan aplikasi utama untuk perangkat mobile. Sebagai Programmer dan Developer kita bisa melakukan segalanya, mulai dari membuat aplikasi pengiriman SMS hanya dengan dua baris kode, hingga mengganti event pada Home Screen perngkat Android.

Selain itu, bahkan dengan mudah kita bisa membuat dan mengkustomisasi Sistem Operasinya, atau mengganti semua aplikasi default dari Google.



Gambar 2.3 Logo Android

Semua aplikasi yang dibuat untuk Android akan memiliki akses yang setara dalam mengakses seluruh kemampuan handset, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan platform Android ini, Programmer dan Developer secara penuh akan bisa mengkustomisasi perangkat androidnya.

Android built in pada Linux Kernel(Open Linux Kernel), dengan sebuah mesin virtual yang telah didesain dan untuk mengoptimalkan penggunan sumber daya memori dan hardware pada lingkungan perangkat mobile. Dalvik adalah nama dari Android Virtual Machine, yang merupakan interpreter virtual mesin yang akan mengeksekusi file kedalam format Dalvik Executeable(*.dex). sebuah format yang telah dirancang untuk ruang penyimpanan yang efisien dan eksekusi memori yang terpetakan.

Dalvik Virtual Machine(Dalvik VM) berbasis register, dan dapat mengeksekusi kelas yang telah terkompilasi pada compiler bahasa Java, kemudian di

transformasikan ke dalam native format dengan menggunakan tool “dx” yang telah terintegrasi. Kita mungkin telah mengenal JavaVM(Java Virtual Machines), yang saat ini bisa kita temukan pada setiap komputer desktop. Berbeda dengan DalvikVM, JavaVM berbasis stack. DalvikVM memiliki keunggulan dengan menggunakan Registered Based, ini karena pada prosesor perangkat genggam telah dioptimasi untuk eksekusi berbasis register.

Android saat ini tidak hanya berjalan pada handphone, beberapa vendor menanamkan Android pada Tablet, Internet Tablet, E-Book Reader, Laptop, dan gadget lainnya. Dengan begitu akan sangat berharga sekali mempelajari platform ini, dengan arsitekturnya yang terbuka, maka platform ini Android adalah platform mobile masa depan.

2.6.2 Definisi Android

Android merupakan subset perangkat lunak untuk perangkat mobile yang meliputi sistem operasi middleware dan aplikasi inti yang di release oleh Google. Sedangkan Android SDK(Software Development Kit) menyediakan Tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform Android dengan menggunakan bahasa pemrograman Java. Dikembangkan bersama antara Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, NVIDIA yang tergabung dalam OHA(Open Handset Alliance) dengan tujuan membuat standar terbuka untuk perangkat bergerak(mobile device).

2.7 Arsitektur Android

Arsitektur Android dalam paket sistem operasi android tediri dari beberapa unsur seperti tampak pada gambar 2.3. Secara sederhana arsitektur android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu framework yang menyediakan dan mengatur alur proses aplikasi (Google IO, Android Anatomy and Physiology).

2.7.1 Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (Interprocess Communication) untuk mengatur aplikasi dan lubang keamanan.

2.7.2 Libraries

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar Berkeley Software Distribution (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

1. Media Library untuk memutar dan merekam berbagai macam format audio dan video.
2. Surface Manager untuk mengatur hak akses layer dari berbagai aplikasi.
3. Graphic Library termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D.
4. SQLite untuk mengatur relasi database yang digunakan pada aplikasi.
5. SSl dan WebKit untuk browser dan keamanan internet.

Pustaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan Native Development Toolkit (NDK).

2.7.3 Android Runtime

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. Android Runtime merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam Android Runtime terdapat 2 bagian utama, diantaranya:

1. Pustaka Inti, android dikembangkan melalui bahasa pemrograman Java, tapi Android Runtime bukanlah mesin virtual Java. Pustaka inti android

menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus android.

2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format Dalvik Executable (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti threading dan manajemen memori yang terbatas (Nicolas Gramlich, Andbook, anddev.org).

2.7.4 Application Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan user interface dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut (Hello Android 2nd Edition):

1. Activity Manager, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”Backstack“ untuk navigasi penggunaan.
2. Content Providers, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.

3. Resuource Manager, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan file layout.
4. Location Manager, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.
5. Notification Manager, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada status bar.

2.7.5 Application Layer

Puncak dari diagram arsitektur android adalah lapisan aplikasi dan widget. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam Android runtime dengan menggunakan kelas dan service yang tersedia pada framework aplikasi.

Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada android semua aplikasi, baik aplikasi inti (native) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (Application Programming Interface) yang sama.

2.8 Perangkat Lunak Pembantu

2.8.1 Eclipse

Eclipse merupakan sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse:

1. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. 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.
3. 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.

2.8.2 JDK (Java Development Kit 7)

JDK adalah perangkat lunak Java yang didalamnya sudah include library untuk keperluan debugging dan library untuk keperluan running. JDK adalah perangkat lunak yang digunakan untuk membangun suatu aplikasi-aplikasi Java. Tanpa JDK, tidak akan bisa membangun aplikasi tersebut. Sedangkan JRE (Java Runtime Environment) adalah perangkat lunak yang digunakan untuk keperluan runtime atau menjalankan program yang telah dikompilasi sebelumnya oleh JDK.