

II. TINJAUAN PUSTAKA

2.1 Aplikasi

2.1.1 Perangkat Lunak Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah Microsoft Office dan OpenOffice.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering

kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat ditenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain :

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

Pada pengertian umumnya, aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya

2.1.2 Aplikasi Komputer

Aplikasi komputer adalah kumpulan perintah komputer yang menggunakan kemampuan komputer secara langsung dan menyeluruh untuk mengerjakan suatu tugas sesuai dengan kebutuhan pengguna.

Hal-hal yang perlu diperhatikan dalam pengembangan aplikasi komputer :

1. Kebutuhan dan batasan-batasan yang diinginkan pengguna harus ditentukan dan dinyatakan secara tegas.
2. Produk perangkat lunak harus dirancang sedemikian rupa sehingga mampu mengakomodasi paling tidak kepentingan tiga pihak berikut :
 - a. Pelaksana Implementasi
 - b. Pengguna
 - c. Pemelihara produk
3. Dilengkapi dengan dokumen-dokumen pendukung

Beberapa atribut yang dapat menjadi ukuran kualitas suatu aplikasi :

- Kegunaan
- Keandalan
- Kejelasan
- Efisiensi

(Anonim, 2011a)

2.2 Do'a

Doa adalah memohon atau meminta suatu yang bersifat baik kepada Allah SWT seperti meminta keselamatan hidup, rizki yang halal dan keteguhan iman. Sebaiknya kita berdoa kepada Allah SWT setiap saat karena selalu didengar olehNya.

Do'a Sebagai Awal dan Akhir Dari Aktivitas Keseharian

“Dan apabila hamba-hamba-Ku bertanya kepadamu (para rasul) tentang Aku maka (jawablah) bahwasanya Aku adalah dekat. Aku mengabulkan permohonan orang-orang yang mendo'a apabila ia berdo'a kepada-Ku”.

(Al Qur'an Dan Terjemahannya Al-Baqarah : 186 (dalam Asy Syarifain, 1971))

Berdo'a merupakan sesuatu hal yang dapat meringankan beban hidup seseorang sekaligus sebagai jembatan jiwa seorang hamba kepada penciptanya. Jembatan jiwa yang senantiasa dibutuhkan untuk mencapai puncak pengabdian kepada Yang Maha Kuasa karena dengan hal itulah, seorang muslim bisa dikatakan telah mengakui bahwa dirinya hanyalah seorang hamba yang selalu membutuhkan perlindungan dan pertolongan dari sang Rabb agar mendapat keselamatan di dunia dan akhiratnya.

Rasulullah merupakan sosok teladan bagi umat muslim karena keseharian beliau selalu diisi dengan bicara dan tingkah laku yang bernilai ibadah. Setiap aktivitas beliau selalu diawali dan diakhiri dengan do'a. Do'a yang diiringi dengan pujian dan permohonan kepada Allah SWT dengan penuh sungguh-sungguh dan tulus ikhlas.

Do'a yang telah dilakukan Rasulullah SAW sebagai contoh bagi umat muslim, yang tercantum dalam kitab 'Ihya' Ulumuddin' karya Al Imam Al-Ghazali yang diringkas sebagai berikut :

1. Orang yang berdo'a hendaknya memilih waktu-waktu yang utama, seperti hari arafah, bulan ramadhan, hari jum'at, waktu sahur (akhir malam).

2. Berdo'a dengan mengambil kesempatan saat amal perbuatan yang penting. Abu Hurairah r.a ,” Sesungguhnya pintu-pintu langit terbuka pada saat serbuan pasukan dalam pertempuran fisabilillah, pada waktu turun hujan, pada waktu mengerjakan shalat fardhu, maka gunakanlah kesempatan itu untuk berdo'a”.
3. Orang yang berdo'a hendaknya menghadap kiblat dan mengangkat kedua tangannya sehingga terbuka dan terlihat putihnya ketiak.
4. Orang yang berdo'a hendaknya melirihkan suaranya.
5. Jangan bersajak dan berirama dalam berdo'a. Merendah diri dan jangan dibuat-buat serta melampaui do'a-do'a yang maturasat (pilihan serta berkesinambungan dari nabi SAW).
6. Menunjukkan kerendahan diri, kekhusyuan, hasrat harapan dan kecemasan. Firman Allah SWT, dalam surat Al-Anbiyaa ayat 90, yang artinya :
 “Mereka adalah orang-orang yang selalu bersegera dalam perbuatan-perbuatan baik dan mereka berdo'a kepada Kami dengan harap dan cemas”.
7. Berkeyakinan akan dikabulkannya do'a dan selalu mengharap dengan kesungguhan hati.
8. Hendaknya mendesakkan do'anya dan mengulang-ulang do'a tiga kali.
 Do'a memiliki banyak manfaat bagi umat Muslim yang mengetahui dan menyadarinya, maka seorang Muslim akan selalu menjadikan do'a sebagai awal dan akhir dari segala aktivitas keseharian hidupnya sebagai permohonan dan rasa syukurnya terhadap Allah SWT.

(Anonim, 2011b)

2.3 Sistem Operasi

Sistem operasi adalah perangkat lunak sistem yang bertugas untuk melakukan kontrol dan manajemen perangkat keras serta operasi-operasi dasar sistem, termasuk menjalankan software aplikasi seperti program-program pengolah kata dan *web browser*.

Secara umum, Sistem Operasi adalah *software* pada lapisan pertama yang ditaruh pada memori komputer pada saat komputer dinyalakan. Sedangkan *software-software* lainnya dijalankan setelah Sistem Operasi berjalan, dan Sistem Operasi akan melakukan layanan inti umum untuk *software-software* itu. Layanan inti umum tersebut seperti akses ke *disk*, manajemen memori, *scheduling task*, dan antar-muka *user*. Sehingga masing-masing *software* tidak perlu lagi melakukan tugas-tugas inti umum tersebut, karena dapat dilayani dan dilakukan oleh Sistem Operasi. Bagian kode yang melakukan tugas-tugas inti dan umum tersebut dinamakan dengan "kernel" suatu Sistem Operasi.

Sistem operasi-sistem operasi utama yang digunakan komputer sistem umum (termasuk PC, komputer personal) terbagi menjadi 3 kelompok besar :

1. Keluarga Microsoft Windows - yang antara lain terdiri dari Windows Desktop Environment (versi 1.x hingga versi 3.x), Windows 9x (Windows 95, 98, dan Windows ME), dan Windows NT (Windows NT 3.x, Windows NT 4.0, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows

7 (Seven) yang dirilis pada tahun 2009, dan Windows Orient yang akan dirilis pada tahun 2014).

2. Keluarga Unix yang menggunakan antarmuka sistem operasi POSIX, seperti SCO UNIX, keluarga BSD (Berkeley Software Distribution), GNU/Linux, MacOS/X (berbasis kernel BSD yang dimodifikasi, dan dikenal dengan nama Darwin) dan GNU/Hurd.
3. Mac OS, adalah sistem operasi untuk komputer keluaran Apple yang biasa disebut Mac atau Macintosh.

Sedangkan komputer *mainframe*, dan super komputer menggunakan banyak sekali sistem operasi yang berbeda-beda, umumnya merupakan turunan dari sistem operasi UNIX yang dikembangkan oleh vendor seperti IBM AIX, HP/UX, dll.

(Anonim, 2010c)

2.4 Sistem Operasi Android

2.4.1 Sejarah Sistem Operasi 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 konsorium perangkat mobile terbuka, merilis Google Android SDK, setelah mengumumkannya seminggu sebelumnya. Dan sambutannya sangat luarbiasa, hampir semua media berita tentang IT & Programing memberitakan tentang dirilisnya Android SDK (*Software Developmnet Kit*).

(Mulyadi, 2010)

Perkembangan sistem operasi android antara lain:

1. 2007-2008: Produk awal

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar yang menggunakan Android pada sistem 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, perangkat mobile yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

2. Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan *email*.

3. Android versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke *headset bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

4. Donut (versi 1.6)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak;

teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

5. Android versi 2.0/2.1 (*Eclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

6. Android versi 2.2 (*Froyo : Frozen Youghurt*)

Pada 20 Mei 2010, Android versi 2.2 (*Froyo*) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan *auto update* dalam aplikasi Android Market.

7. Android versi 2.3 (*Gingerbread*)

Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio

baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

8. Android bagi komunitas sumber terbuka (*open source*)

Android memiliki berbagai keunggulan sebagai software yang memakai basis kode komputer yang bisa didistribusikan secara terbuka (*open source*) sehingga pengguna bisa membuat aplikasi baru di dalamnya. Android memiliki aplikasi native Google yang terintegrasi seperti pushmail Gmail, Google Maps, dan Google Calendar.

Para penggemar open source kemudian membangun komunitas yang membangun dan berbagi Android berbasis *firmware* dengan sejumlah penyesuaian dan fitur-fitur tambahan, seperti FLAC lossless audio dan kemampuan untuk menyimpan download aplikasi pada microSD card.

(Anonim, 2010c)

2.4.2 Android

Android adalah *software* untuk perangkat mobile yang mencakup sistem operasi, *middleware* dan aplikasi kunci. Pengembangan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Serangkaian aplikasi inti Android antara lain klien email, program SMS, kalender, peta, *browser*, dan kontak.

Dengan menyediakan sebuah *platform* pengembangan yang terbuka, pengembang Android menawarkan kemampuan untuk membangun aplikasi yang sangat kaya dan inovatif. Pengembang bebas untuk mengambil keuntungan dari perangkat keras, akses informasi lokasi, menjalankan background services, mengatur alarm, tambahkan pemberitahuan ke status bar, dan banyak lagi.

Android bergantung pada versi Linux 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, network stack, dan model driver. Kernel juga bertindak sebagai lapisan abstraksi antara *hardware* dan seluruh *software* stack.

2.4.3 Features

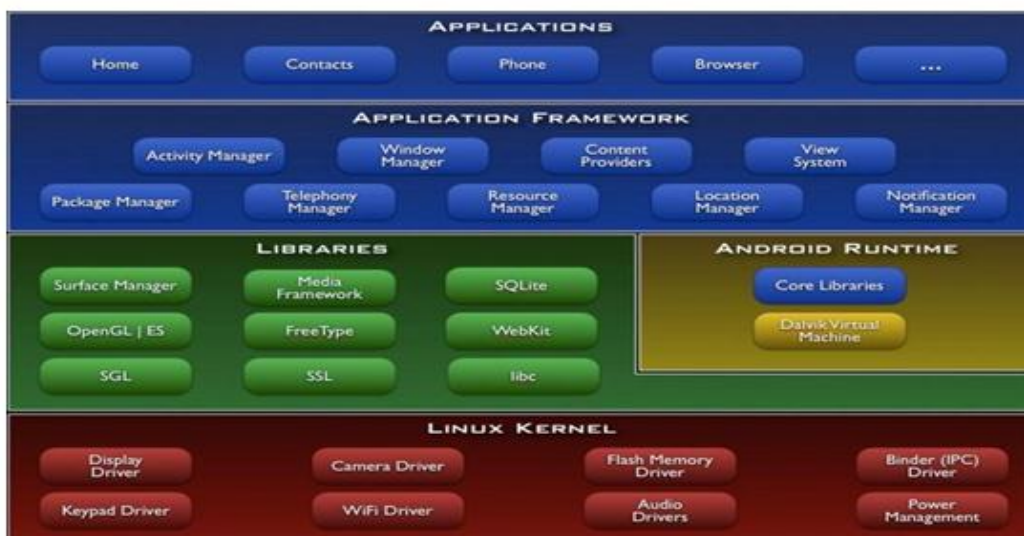
Features Aplikasi yang tersedia pada platform Android saat ini antara lain :

- *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*
- Mesin Virtual Davilk berjalan pada platform Linux kernel dan dioptimalkan untuk perangkat *mobile*
- *Integrated Browser* berdasarkan *open source* engine WebKit
- Grafis yang dioptimalkan dan didukung oleh library grafis 2D yang terkustomisasi, grafis 3D berdasarkan spesifikasi openGL ES 1,0 (*Opsioanal akselerasi hardware*)
- SQLite untuk menyimpan data
- *MediaSupport* yang mendukung audio, video, dan gambar (MPEG4, MP3, AAC, AMR, JPG, PNG, GIF)

- GSM Telephony (tergantung *hardware*)
- Bluetooth, EDGE, 3G, dan WIFI (tergantung *hardware*)
- Dukungan Perangkat Tambahan : Andorid dapat memanfaatkan kamera, layar sentuh, accelerometers, magnometers, GPS, akselerasi 2D (*dengan perangkat orientasi, scalling, konversi format piksel*) dan akselerasi grafis 3D
- *Multi-touch* : kemampuan layaknya *handset* modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat
- Lingkungan *development* yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk eclipse IDE
- Market : Seperti kebanyakan handphone yang memiliki tempat penjualan aplikasi, market pada android merupakan katalog aplikasi yang dapat di download dan di install pada handphone melalui internet.

2.4.4 Arsitektur Android

Diagram Gambar 2.1 menunjukkan komponen utama dari sistem operasi Android



Gambar 2.1 Arsitektur Platform Android (Mulyadi, 2010)

Berikut penjelasan komponen dari arsitektur *platform android*:

1. Linux Kernel

Android bukan Linux, akan tetapi dibangun pada platform Linux Kernel yaitu versi 2.6 sehingga keandalannya bisa dipercaya. Untuk inti sistem service Linux yang digunakan seperti keamanan, manajemen memori, proses manajemen, network, dan driver model. Seperti yang terlihat pada Gambar 2.1, Linux Kernel menyediakan Driver Layar, Kamera, Keypad, WiFi, Flash Memory, Audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan keamanan. Kernel juga bertindak sebagai lapisan abstrak antara *hardware* dan *software* stacknya.

2. Libraries

Android menyertakan libraries C / C + + yang digunakan oleh berbagai komponen dari sistem Android. Kemampuan ini disediakan kepada *Developer* aplikasi melalui *Framework* Aplikasi Android. Beberapa inti libraries tercantum di bawah ini :

- Sistem C library – Variasi dari Implementasi BSD-berasal pelaksanaan system standar C library (*libc*), sesuai untuk perangkat embedded berbasis Linux
- Media Library – Packet Video berdasarkan OpenCORE; library mendukung pemutaran rekaman dan populer banyak format audio dan video, serta file gambar, termasuk MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG
- *Surface Manager* – mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi

- LibWebCore – mesin web modern yang *powerfull* yang baik *browser* Android *embeddable web*
- SGL – mesin grafis 2D
- 3D libraries – sebuah pelaksanaan berdasarkan opengl ES 1,0 API; perpustakaan baik menggunakan perangkat keras akselerasi 3D (apabila tersedia) atau yang disertakan, sangat optimal 3D *software rasterizer*
- FreeType – bitmap dan vector font rendering
- SQLite – mesin *database* yang kuat dan ringan, dan penghubung tersedia untuk semua aplikasi

3. Android-runtime

Android terdiri dari satu set core libraries yang menyediakan sebagian besar fungsi yang sama dengan yang dapat pada core libraries bahasa pemrograman Java. Setiap aplikasi menjalankan prosesnya sendiri dalam Android, dengan masing-masing instan dari Mesin Virtual Davilk (*Davilk VM*). Davilk dirancang agar perangkat dapat menjalankan multiple VMs secara efisien. Mesin Virtual Davilk mengeksekusi file dalam executable (*.dex*), sebuah format yang dioptimalkan untuk memori yang kecil. Davilk VM berbasis, berjalan dan dikompilasi oleh compiler bahasa Java yang telah ditransformasikan ke dalam *.dex* format oleh tool “dx” yang telah disertakan. Davilk VM bergantung pada kernel Linux yang berfungsi, seperti threading dan manajemen memori tingkat rendahnya.

4. Framework-Aplikasi

Pengembang memiliki akses penuh menuju API framework yang sama, dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) dengan mudah.

Semua aplikasi merupakan rangkaian set layanan dan sistem, termasuk:

- *Views* yang kaya dan *extensible* yang dapat digunakan untuk membangun aplikasi, termasuk list, grids, kotak teks, tombol, dan bahkan sebuah *embeddable web*.
- Content Provider yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak telp) atau membagi data tersebut.
- *Resource Manager* yang memungkinkan semua aplikasi menampilkan alert yang bisa dikustomisasi didalam status bar.
- Activity Manager yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum *backstack*.

5. Applications

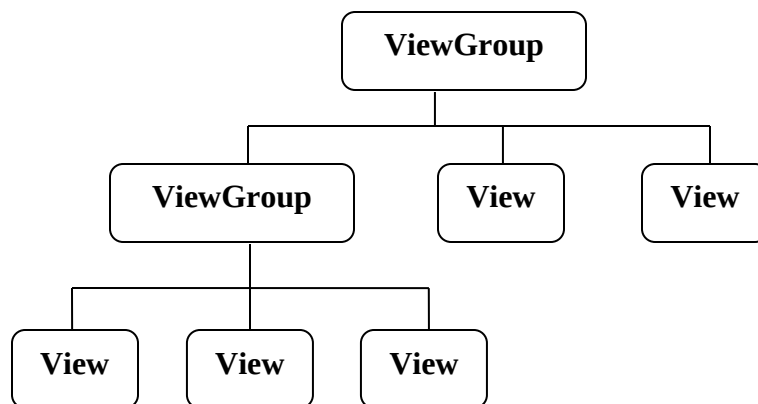
Android telah menyertakan aplikasi inti seperti *email client*, SMS, kalender, peta, *browser*, kontak, dan lain-nya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemograman Java. Pada layer inilah *developer* menempatkan aplikasi yang dibuat. Yang istimewa adalah pada android semua aplikasi baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan pada layer aplikasi dengan menggunakan librari API yang sama.

2.4.5 User Interface

Android *User Interface* bisa dibuat dengan dua cara, yaitu dengan menggunakan kode XML atau menulisnya melalui kode java. Membuatnya dengan XML sangat direkomendasikan, karena seperti yang kita tahu itu menggunakan konsep MVC (*Model-Viewer-Control*) yang memisahkan *user interface* dengan kode logika program (*Logic Program*). Dengan begitu, adaptasi program aplikasi dari satu format resolusi layar ke format resolusi yang lain menjadi mudah.

1. Hirarki Elemen Layar

Unit dasar dari aplikasi android adalah *activity*, yaitu sebuah objek dari kelas **android.app.Activity**. *Activity* bisa melakukan banyak hal, tetapi dia sendiri tidak akan terlihat secara langsung dilayar. Untuk membuat *activity* terlihat di layar adalah dengan membuatnya *User Interface*, dimana disini kita akan menggunakan *viewgroups* (Unit dasar *User Interface* dalam platform Andorid)



Gambar 2.2 Struktur Pohon *User Interface* (Raharjo dkk, 2007)

2. Resources

Resources adalah file eksternal (*non-code files*) yang digunakan oleh kode kita dan dikompil ke dalam aplikasi pada saat build. Android mendukung banyak macam file *resources*, termasuk file XML, PNG dan JPEG. File XML mempunyai perbedaan format tergantung bagaimana mereka deskripsikan. *Resources* ditempatkan di luar kode, dan file XML dikompil ke dalam binary, format yang bisa cepat diproses untuk alasan efisiensi. String dikompres ke dalam format penyimpanan yang lebih efisien.

3. R.java

R.java merupakan file yang *degenerate* secara otomatis, yang mengindex semua *resources* dari project. *User* bisa menggunakan kelas ini ke dalam kode sebagai cara untuk merefer ke *resources* yang dimasukkan ke dalam *project*.

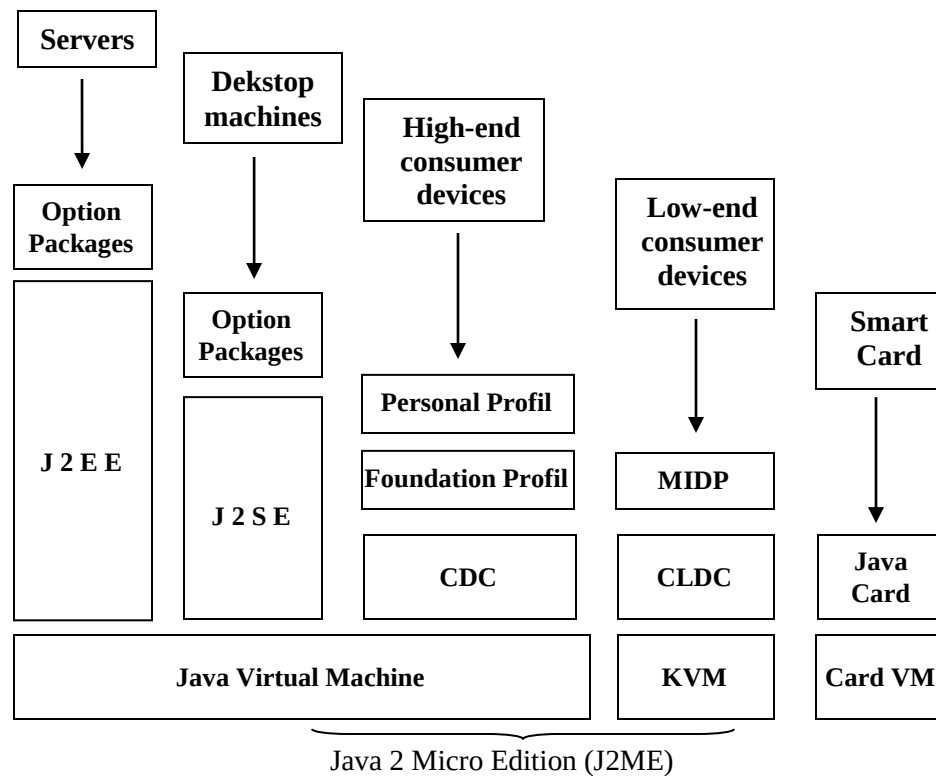
2.5 Definisi Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems sejak tahun 1991. Bahasa ini dikembangkan dengan model yang mirip dengan bahasa C++ dan Smalltalk, namun dirancang agar lebih mudah dipakai dan platform independent, yaitu dapat dijalankan di berbagai jenis sistem operasi dan arsitektur komputer. Bahasa ini juga dirancang untuk pemrograman di Internet sehingga dirancang agar aman dan portabel.

Alasan utama pembentukan bahasa java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan diberbagai macam perangkat elektronik, seperti *microwave oven* dan *remote control*, sehingga java harus bersifat *portable* atau sering disebut *platform independent* (tidak tergantung pada *platform*). Itulah yang menyebabkan dalam dunia pemograman Java, dikenal adanya istilah ‘*write once, run everywhere*’, yang berarti kode program hanya ditulis sekali, namun dapat dijalankan di *platform* manapun.

(Raharjo dkk, 2007)

Platform Java memiliki tiga edisi yang berbeda, yaitu J2EE (Java2 *Enterprise Edition*), J2ME (Java2 *Micro Edition*), dan J2SE (Java2 *Second Edition*). J2EE adalah kelompok dari beberapa API dari Java dan teknologi selain Java. J2EE dibuat untuk membuat aplikasi yang kompleks. J2EE sering dianggap sebagai *middldeware* atau teknologi yang berjalan di *server*, namun sebenarnya J2EE tidak hanya terbatas untuk itu. Faktanya J2EE juga mencakup teknologi yang dapat digunakan di semua lapisan dari sebuah sistem informasi. Implementasi J2EE menyediakan kelas dasar dan API dari Java yang mendukung pengembangan dan rutin standar untuk aplikasi klien maupun *server*, termasuk aplikasi yang berjalan di *web browser*. J2SE adalah lingkungan dasar dari Java, sedangkan J2ME digunakan untuk menangani pemograman pada perangkat-perangkat kecil, yang tidak memungkinkan untuk mendukung implementasi J2SE secara penuh.



Gambar 2.3 Ruang Lingkup J2EE, J2SE, dan J2ME (Shalahuddin dan Rosa, 2010)

2.6 UML

UML (*Unified Modelling Language*) adalah suatu "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan suatu standar untuk merancang model suatu sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih

cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET.

UML juga mendefinisikan diagram-diagram sebagai berikut :

1. *Use case diagram*
2. *Activity diagram*
3. *Sequence diagram*

2.6.1 Use Case Diagram

Use case diagram adalah diagram menggambarkan suatu fungsionalitas yang diharapkan dari sebuah sistem dan bagaimana sistem berinteraksi dengan dunia luar. Yang ditekankan dalam *use case diagram* adalah apa yang diperbuat sistem, dan bukan bagaimana sistem itu melakukannya.

Use case diagram sangat membantu bila sedang disusun *requirement* suatu sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua fitur yang ada pada sistem. Suatu *use case* dapat meng-include fungsionalitas *use case* lain sebagai bagian dari proses sistem tersebut.

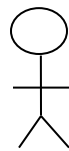
Use case diagram dapat digunakan selama proses analisis untuk menangkap *requirement* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case diagram* berperan untuk menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam suatu model mungkin terdapat satu atau beberapa *use case diagram*. Kebutuhan atau *requirement* sistem adalah fungsionalitas apa yang harus disediakan oleh sistem kemudian didokumentasikan

pada model *use case* yang menggambarkan fungsi sistem yang diharapkan (*use case*), dan yang mengelilinginya (*actor*), serta hubungan antara *actor* dengan *use case* (*use case diagram*) itu sendiri.

Notasi gambar yang dipakai *use case* :

1. *Actor*

Seorang aktor adalah suatu entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Bentuk aktor ditunjukkan pada Gambar 2.4.



Gambar 2.4 *Actor*

2. *Case*

Case menggambarkan deskripsi yang melibatkan *actor*. Bentuk *case* ditunjukkan pada Gambar 2.4.



Gambar 2.5 *Case*

3. *Extend*

Extend yaitu kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.

4. *Include*

Include yaitu aktivitas yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.

Cara menemukan *use case* :

- a. Pola perilaku perangkat lunak aplikasi.
- b. Gambaran tugas dari sebuah *actor*.
- c. Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada *actor*.
- d. Apa yang dikerjakan oleh suatu perangkat lunak (bukan bagaimana cara mengerjakannya).

2.6.2 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Suatu aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

2.6.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal.

Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

(Dharwiyanti, 2003)