BAB II

LANDASAN TEORI

2.1. Aplikasi

Menurut Daryanto (2004:347), aplikasi adalah software atau perangkat lunak yang dibuat untuk mengerjakan menyelesaikan masalah-masalah khusus.

Sedangkan menurut Jogiyanto (2004:4), aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jadi aplikasi secara umum adalah suatau proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal.

Dari uraian diatas dapat disimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang berisi perintah untuk menyelesaikan masalah dan pengolahan data.

2.1.1. Aplikasi Mobile

Arti aplikasi mobile adalah aplikasi yang melibatkan piranti bergerak dan melibatkan media komunikasi nirkabel. Oleh karena itu, peminatan ini menekuni sistem tersebar, keamanan jaringan, pemrograman jaringan, aplikasi mobile untuk bisnis, analisis dan perancangan jaringan, pemrograman web lanjut, sistem layanan berbasis lokasi, infrastruktur layanan mobile, teknologi nirkabel dan aplikasinya, sistem tertanam, serta hal-hal yang berkaitan dengan itu, termasuk di dalamnya aspek perangkat keras komputer.

2.2. Rukun Islam

2.2.1. Makna dan Hakikat Rukun Islam

Islam dibangun di atas lima dasar, yaitu Rukun Islam. Ibarat sebuah rumah, Rukun Islam merupakan tiang-tiang atau penyangga bangunan keislaman seseorang. Di dalamnya tercakup hukum-hukum Islam yang mengatur seluruh aspek kehidupan manusia. "Sesungguhnya Islam itu dibangun atas lima perkara: bersaksi sesungguhnya tidak ada Tuhan selain Allah dan Muhammad adalah utusan-Nya, mendirikan shalat, menunaikan zakat, haji ke Baitullah dan puasa di buIan Ramadhan" (HR. Bukhari Muslim). Bagi siapa saja yang telah mengerjakan Rukun Islam yang lima, belum berarti bahwa ia telah total masuk ke dalam Islam. Ia baru membangun landasan bagi amal-amalnya yang lain.

Rukun Islam merupakan landasan operasional dari Rukun Iman. Belum cukup dikatakan beriman hanya dengan megerjakan Rukun Islam tanpa ada upaya untuk menegakkannya. Rukun Islam merupakan training/pelatihan bagi orang mukmin menuju mardhotillah/keridhoan Allah.

- Syahadat adalah agreement (perjanjian) antara seorang muslim dengan Allah SWT [7.172]. Seseorang yang telah menyatakan Laa ilaaha ilallaah berarti telah siap untuk fight (bertarung) melawan segala bentuk ilah di luar Allah di da1am kehidupannya [29:2].
- Shalat adalah training: sebagai latihan agar setiap muslim di dalam kehidupannya adalah dalam rangka sujud (beribadah) kepada Allah [6:162]

- Zakat adalah training, yaitu sebagai latihan agar menginfakkan hartanya, karena setiap harta seorang muslim adalah milik Allah.[57:7, 59:7]. “Engkau ambil zakat itu dari orang-orang kaya mereka dan engkau kembalikan kepada orang-orang fakir mereka” (HR Mutafaqun ‘alahi).
- Shaum / Puasa adalah training, yaitu sebagai latihan pengendalian kebiasaan pada jasmani, yaitu makan dan minum dan ruhani, yaitu hawa nafsu. [2:185]
- Haji adalah training, yaitu sebagai latihan dalam pengorbanan jiwa dan harta di jalan Allah, mengamalkan persatuan dan persamaan derajat dengan sesama manusia. [22:27-28]

2.3. Rukun Iman

2.3.1. Hakikat Iman

Pengertian Iman menurut ahlussunah : Iman terdiri dari tiga unsur, yaitu pembenaran dengan hati, diikrarkan dengan lisan dan diamalkan dengan anggota badan, yang satu dengan lainnya tidak dapat dipisahkan. Jadi, Iman adalah keyakinan dan sekaligus juga amal [49:15].

2.3.2. Rukun Iman

Rukun Iman merupakan basis konsepsional atau landasan idiil yang mendasari pemikiran, ucapan dan tindakan seorang muslim. Artinya: seorang muslim yang beriman maka pemikiran, ucapan dan tindakannya tidak akan bertentangan dengan keimanannya kepada Allah, Malaikat, Kitab, Rasul, Taqdir dan Kiamat. Orang yang beriman haruslah beriman kepada enam Rukun iman

(2:285, 4:136) dan Hadits Ketika Nabi ditanya Malaikat Jibril tentang iman, maka jawab Nabi. ”Hendaklah engkau beriman kepada Allah, kepada Malaikat-Nya, kepada kitab-kitabNya, kepada Utusan-utusanNya, kepada Hari Kiamat dan hendaklah engkau beriman kepada Qodar yang baik dan yang buruk” (HR Muslim), barangsiapa yang mengingkari salah satunya maka ia telah mengingkari seluruh Rukun Iman.

1. Iman kepada Allah SWT . Konsekuensinya : mencintai Allah SWT [2:165]. Tanda-tandanya: lihat QS 8:2. Akibatnya: ikh1ash dalam menjalankan perintah-perintahNya.
2. Iman kepada Malaikat [50:16-18]. Konsekuensinya: tidak mungkin Seorang mu’min berbuat ma’siat karena selalu ditongkrongi Malaikat.
3. Iman kepada Kitab-Kitab [2:2, 20:1-3] Konsekuensinya: menjadikan Al-Qur’an sebagai pedoman hidup.
4. Iman kepada Nabi dan Rasul [33:40]. Konsekuensinya: mencintai dan mengikutinya [3:31-32].
5. Iman kepada Hari Akhir [3:185]. Konsekuensinya: mempersiapkan diri untuk menghadapiNya.
6. Iman kepada Takdir [22:7]. Konsekuensinya: berprinsip bahwa “Janganlah kita mempersoalkan apa-apa yang Allah ingin lakukan terhadap kita, tetapi kita harus melakukan apa-apa yang Allah ingin dari kita.”

2.4. Java

Bahasa Java dikembangkanoleh *Sun Microsystem* tahun 1991 sebagai bagian dari suatu proyek penelitian untuk mengembangkan software bagi konsumer barang – barang elektronik seperti televise, *VCR, toaster* dan mesin – mesin lainnya yang dapat dibeli di swalayan. Tujuan penciptaan Java pada waktu itu adalah menjadi suatu program yang berukuran kecil, efisien, dan *portable* di segala jenis *hardware*. Tujuan yang sama ini membuat Java menjadi satu bahasa yang ideal untuk mendistribusikan program – program yang dapat dijalankan melalui *www* dan juga suatu bahasa pemrograman untuk segala tujuan untuk mengembangkan program – program yang dapat digunakan dengan mudah dan *portable* di berbagai *platform* yang berbeda.

Sekarang, *Sun* telah mengeluarkan berbagai program Java yang dapat digunakan seperti *Java API,* atau JDK atau *JAVA Developer Kit* . Selain itu, banyak juga program-program lain yang dapat digunakan untuk membuat program Java, seperti *Eclipse, NetBeans, JBuilder, JCreator, J++,* dan sebagainya.

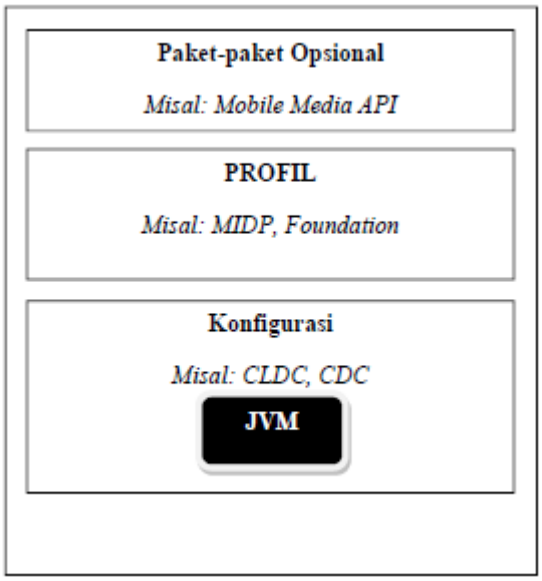
Saat ini distribusi Java dan kelas pendukungnya dibagi dalam tiga bagian yang masing-masing memiliki konsentrasi tersendiri yaitu:

1. *Java 2 Standart Edition* (J2SE), untuk aplikasi *desktop*
2. *Java 2 Enterprise Edition* (J2EE), untuk aplikasi *server*
3. *Java 2 Micro Edition* (J2ME), untuk piranti dengan kemampuan terbatas

2.4.1. J2ME

J2ME merupakan sebuah kombinasi yang terbentuk antara kumpulan *interface.* Java yang sering disebut dengan Java API (*Application programming interface*) dengan JVM (*Java Virtual Machine*) yang didesain khusus untuk alat, yaitu JVM dengan ruangan terbatas. Kombinasi tersebut kemudian digunakan untuk melakukan pembuatan aplikasi-aplikasi yang dapat berjalan di atas alat (dalam hal ini *mobile device*).

J2ME sendiri pada dasarnya terdiri dari tiga buah bagian, yaitu konfigurasi, profil, dan paket-paket opsional, seperti yang ditunjukkan oleh gambar 2.1.



Gambar 2.1. Bagian-bagian di dalam platform J2ME

2.4.1.1. Konfigurasi

Konfigurasi merupakan bagian yang berisi JVM dan beberapa *library* kelas lainnya. Perlu diperhatikan bahwa JVM yang dimaksud disini bukanlah JVM

tradisional seperti yang terdapat dalam J2SE, melainkan JVM yang sudahdidesain secara khusus untuk alat.

Terdapat dua buah konfigurasi yang disediakan oleh Sun Microsystems, yaitu CLDC (*Connected Limited Device Configuration*) dan CDC (*Connected Device Configuration*). Target alat dari konfigurasi CLDC adalah alat-alat kecil, seperti telepon selular, PDA, dan *pager*. Membahas lebih jauh mengenai konfigurasi CLDC pada bagian tersendiri dalam bab ini. Pada sisi yang lain, CDC merupakan *superset* dari CLDC sehinga semua yang kelas yang didefinisikan di dalam CLDC aka nada juga didalam CDC.

2.4.1.2. Profil

Profil merupakan bagian perluasan dari konfigurasi. Artinya, selain sekumpulan kelas yang terdapat pada konfigurasi, terdapat juga kelas-kelas spesifik yang didefinisikan lagi di dalam profil. Dengan kata lain, profil akan membantu secara fungsional yaitu dengan mehyediakan kelas-kelas yang tidak terdapat di level konfigurasi.

Adapun profil yang sangat popular penggunaannya adalah profile yang disediakan oleh Sun Microsystems, yaitu yang dinamakan MIDP (*Mobile Information Device Profile*). Berikut ini beberapa profil yang tersedia untuk kebutuhan-kebutuhan spesifik lainnya.

a. PDAP (*Personal DigitalAassistant Profile*) yaitu profil untuk PDA yang memperluas fungsi-fungsi pada konfigurasi CLDC dan digunakan khusus

untuk menambahkan kemampuan-kemampuan lebih apabila dibandingkan dengan penggunaan profil MIDP.\

b. *Foundation Profile*, yaitu profil yang digunakan untuk konfigurasi CDC. Profil ini menambahkan beberapa kelas dari J2SE ke dalam konfigurasi CDC, dan berperan juga sebagai pondasi untuk membentuk profil baru lainnya.

c. *Personal Profile*, yaitu profil yang mendefinisikan ulang *Personal* Java sebagai profil yang dapat digunakan sebagai profil dalam J2ME, Profil ini merupakan hasil perluasan dari *Foundation Profile.*

d. *RMI Profile*, yaitu profil yang menambahkan dukungan RMI (*Remote Method Invocation*) ke dalam konfigurasi CDC.

2.4.1.3. Paket-paket Opsional

Paket-paket opsional merupakan paket-paket tambahan yang dibutuhkan oleh aplikasi sehingga pada saat proses *deployment* paket-paket tersebut perlu didistribusikan juga sebagai bagian dari aplikasi bersangkutan. Sebagai catatan bahwa paket-paket opsional ini bukan merupakan paket yang dibuat oleh perusahaan alat yang digunakan.

2.5. Android

Pada bulan Juli 2000, Google mulai bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, nama-nama pendiri yang bekerja di google

antara lain: Andy Rubi, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc hanyalah sebagai perangkat lunak pada telepon seluler saja. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bekerja untuk menghadapi persaingan dalam pasar telepon seluler.

2.5.1. Awal Perkembangan Android

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar muncul yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan sudah tersedia di pasaran pada tanggal 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 (celah atau kelemahan) dan penambahan fitur-fitur baru.

Telepon seluler atau HP pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada tanggal 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

Di lihat dari pembehasan di atas perkembangan android sangatlah pesat, belum lama bermunculan sekarang sudah meledak sistem oprerasi ini. Apa keunggulan dari sistem operasi ini. Keunggulannya hanya satu This is open Source, jadi banyak orang yang berusaha mengembangkannya.

2.5.2. Versi Android

1. 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*.

2. 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.

3. Android Versi 1.6 (Donut)

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.

4. 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.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset* Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke

dalam sistem operasi Android adalah Shazam, *Backgrounds*, dan *WeatherBug*. Sistem operasi Android dalam situs internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

5. Android Versi 2.2 (Froyo: Frozen Yoghurt)

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*.

6. 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.

7. Android Versi 3.0 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga

berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.

2.5.3. Arsitektur Android

Secara garis besar arsitektur android dapat dijelas dan digambarkan sebagai berikut :

a. Application and Widgets

Application and widgets adalah layer dimana berhubungan dengan aplikasi dan biasanya download aplikasi kemudian lakukan instalasi dan jalankan aplikasi tersebut, delayer inilah terdapat seperti aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

b. Application Frameworks

Android adalah "*Open Development Platform*" yaitu android menawarkan kepada pengembang atau member kemampuan kepada pengembangan untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan serive *background*, mengatur alarm, dan menambahkan tambahan seperti status *notification*s dan masih banyak lagi. Pengembang memiliki akses penuh menuju API *Framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya dengan mudah dapat menggunakan komponen yang sudah digunakan (*reuse*). Sehingga bisa disimpulkan Application Framework

adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem android, karena pada layer inilah aplikasi dapat dirancang, seperti *content-providers* yang berupa sms dan lain sebagainya.

Komponen-komponen yang termasuk didalam *application Framework* adalah sebagai berikut :

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

c. Libraries

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

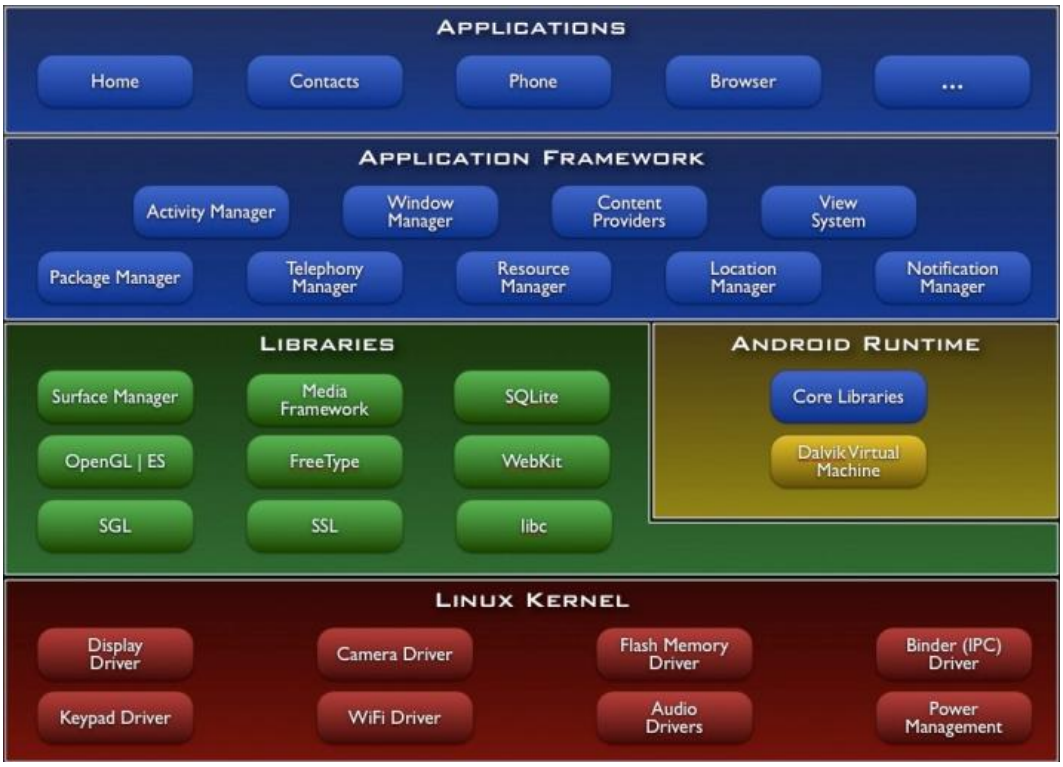
1. *Libraries* media untuk pemutar media audio dan video
2. *Libraries* untuk manajemen tampilan
3. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D
4. *Libraries SQLite* untuk dukungan database
5. *Libraries* SSL dan *WebKit* terintegrasi dengan web *browser* dan security
6. *Libraries Live Webcore* mencakup modern web *browser* dengan engine embedded web view

d. Android Run Time

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

1. *Core Libraries* : aplikasi android dibangun dalam bahasa java, sementara *Dalvik* sebagai virtual mesin bukan *Java Virtual Machine*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa Java/C yang dihandle oleh Core *Libraries*
2. *Dalvik Virtual Machine* : Virtual mesin yang berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien dimana merupakan pengembangan yang mampu membuat linux kernel untuk *threading* dan manajemen tingkat rendah.
3. Linux kernel

   Linux kernel adalah layer dimana inti dari operating sistem dari android itu sendiri, berisi file-file sistem yang mengatur sistem *processing*, *memory*, *resources*, *drivers*, dan sistem-sistem operanting andoroid lainnya. Linux kernel yang digunakan android adalah linux kernel *release* 2.2.



Gambar 2.2 Arsitektur Andoroid

2.5.4. Android SDK

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 sistem 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 member anda kesempatan unutk membuat

aplikasi yang kita butuhkan yang ubkan merupakan aplikasi bawaan Hadphone/*Smartphone*. Beberapa fitur-fitur android yang paling penting adalah :

a. *Framework* : aplikasi yang mendukung pengganti komponen dan reusable.
b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat *mobile*
c. Integrated *Browser* verdasarkan engine *open source* WebKit.
d. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional Ekselerasi *hardware*)
e. SQLite untuk penyimpanan data.
f. Media *Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, 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 termasuk pernagkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE Eclipse.

Untuk source SDK Android ini dapat dilihat dan didownload langsung di situs resmi pengembang SDK Android di http://www.developer.android.com

2.5.5. Komponen Aplikasi Android

Fitur penting android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur *scroller* dan aplikasi lain telah mengembangkan

fitur *scroller* yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan *scroller* yang telah ada.

Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi main() pada aplikasi lain. Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan.

2.5.5.1. *Activities*

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

Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack.* Keadaan suatu *activity* ditentukan oleh posisinya dalam tumpukan *acitivity,* LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back*, atau menutup

activity yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

Ketika *activity* diambil dan disimpan dalam tumpukkan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi :

1) ***Active,*** setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukkan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.
2) ***Paused,*** dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused.* Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat *active* namun tidak dapat menerima masukkan dari pengguna. Dalam kasus ekstrim, android akan menghentikan *activity* dalam keadaan *paused* ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.
3) ***Stopped,*** ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped. Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh

karenanya ketika suatu *activity* dalam kondisi *stopped* maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi *inactive*.

4) ***Inactive,*** kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah ditiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali.

Kondisi transisi ini sepenuhnya ditangani oleh manajer memori android. Android akan memulai menutup aplikasi yang mengandung *activity inactive*, kemudian *stopped activity*, dan dalam kasus luar biasa *paused activity* juga akan di tutup.

2.5.5.2. Services

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

2.5.5.3. *Intents*

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain sebagainya. *Intents* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. *Intents* dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin

menampilkan *URL*, sistem akan menentukan komponen apa yang dibutuhkan oleh *Intents* tersebut.

2.5.5.4. *Broadcast Receiver*

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

2.5.5.5. *Content Providers*

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

2.5.6. Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada android :

1. ***Foreground Activity***

Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini pasti mempertimbangkan siklus hidup *activity*, sehingga perpindahan antar *activity* dapat berlangsung dengan lancar.

2. ***Background Service***

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

3. ***Intermittent Activity***

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

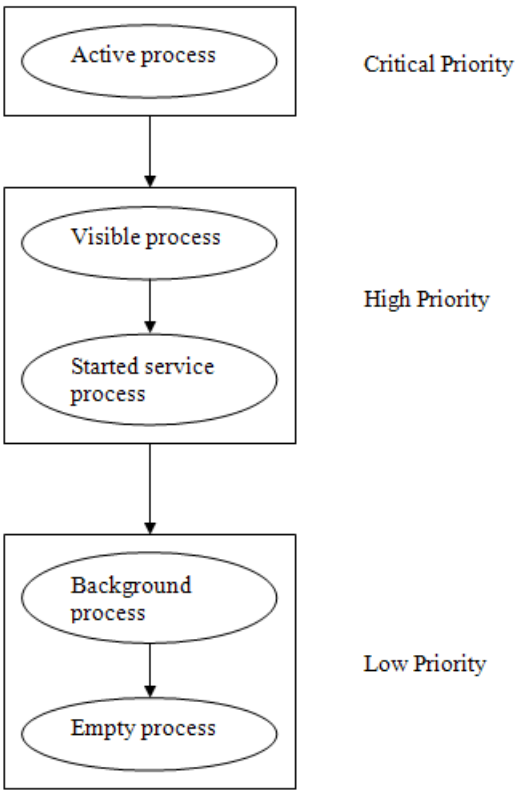
Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi aplikasi memiliki ciri-ciri dari semua kategori. Oleh karenanya perlu pertimbangan bagaimana aplikasi tersebut digunakan dan menentukan kategori aplikasi yang sesuai.

2.5.7. Siklus Hidup Android

Siklus hidup aplikasi android dikelola oleh sistem, berdasarkan kebutuhan pengguna, sumberdaya yang tersedia, dan sebagainya. Misalnya Pengguna ingin

menjalankan browser web, pada akhirnya sistem yang akan menentukan menjalankan aplikasi. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika pengguna ketika itu sedang menjalankan sebuah *Activity*, maka sistem akan memberikan perioritas utama untuk aplikasi yang tersebut. Sebaliknya, jika suatu *Activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *Activity* yang prioritas rendah akan ditutup.

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin virtual pengolah sendiri, dengan ini melindungi penggunaan memori pada aplikasi. Selain itu juga android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karenanya android sangat sensitive dengan siklus hidup aplikasi dan komponen-komponennya. Perlu adanya penanganan terhadap setiap kondisi agar aplikasi menjadi stabil. Gambar 2.3 menunjukkan prioritas dari aplikasi.



Gambar 2.3 Prioritas aplikasi berdasarkan *activity*

2.5.8. Kelebihan Android

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

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan

tanpa harus membayar royality. Sementara pengembang *software* menyukai karena android dapat digunakan diperangkat manapun dan tanpa terikat oleh *vendor* manapun.

2. Arsitektur komponen dasar android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya.

Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.6. Eclipse

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

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.

2.6.1. ADT plugin for Eclipse

Android Development Tools (ADT) adalah plugin untuk Eclipse IDE yang dirancang khusus untuk memberikan *integrated environment* yang kuat untuk membuat aplikasi Android. ADT memberikan kemampuan kepada Eclipse untuk membuat projek baru Android secara cepat, membuat aplikasi *User Interface*, menambahkan komponen berdasarkan Android *Framework* API, melakukan *debugging* aplikasi yang di buat dengan menggunakan Android SDK *tools* dan bahkan melakukan distribusi aplikasi yang dibuat.

Pembuatan aplikasi android dengan Eclipse beserta ADT sangat dianjurkan karena merupakan cara tercepat untuk memulai membuat projek Android. Dengan disediakannya *project setup*, serta *tools* yang sudah terintegrasi di dalamnya, *custom XML editor*, dan *debugging* beserta *output* dalam *emulator* yang sudah disediakan Android SDK, mempermudah para pengembang aplikasi Android dalam pembuatan aplikasinya.

2.6.2. Extensible Markup Language (XML)

Extensible Markup Language (XML) adalah sebuah format text yang sederhana yang berdasarkan SGML(ISO 8879), yang didesain untuk mempertemukan berbagai macam sumber informasi dalam dunia *web*

2.6.2.1. Fungsi dan Tujuan XML

XML sudah dikenal oleh banyak orang, dan adalah dasar untuk pengembangan Software yang meningkat dengan pesat. XML adalah dokumen yang menyimpan data dalam struktur-struktur yang dapat berubah-ubah, hal ini berbeda dengan html yang didesain untuk dokumen hypertext dengan struktur yang baku. Struktur XML yang baik menciptakan struktur yang berbentuk hirarki terstruktur yang memiliki pasangan *tags* awal dan akhir, yang dapat terdiri dari beberapa atribut yang berpasangan. Tidak ada aturan kosakata *tags* yang baku atau pasangan *tags* yang diperbolehkan, jadi hal ini diatur di setiap aplikasi.

Di dalam jurnal Bill Day (2004, Hal : 4) *WML (Wireless Markup Language)* — emerging as the standard for delivering content in most of the world, although North America is still predominantly HDML (Handheld Device Markup Language, the proprietary precursor to WML). Part of the Wireless Application Protocol (WAP) stack defined by the WAP Forum.

2.6.3. 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.

2.7. UML

Unfied Modelling Language (UML) menurut Martin Fowler (2005 : 1) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang

membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Company* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan.

2.7.1. Use Case Diagram

Use Case menurut Martin Fowler (2005 : 141) adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use Case Diagram* menampilkan aktor mana yang menggunakan *use case* mana, *uses case* mana yang memasukkan *use case* lain dan hubungan antara aktor dan *use case*.

2.7.2. Activity Diagram

Activity diagram menurut Martin Fowler (2005 : 163) adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, *activity diagram* memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah *activity diagram* mendukung *behavior paralel*. *Node* pada sebuah *activity diagram* disebut sebagai *action*, sehingga diagram tersebut menampilkan sebuah *activity* yang tersusun dari *action*.

2.7.3. Sequence Diagram

Sequence diagram menurut Munawar (2005 : 187) adalah grafik dua dimensi dimana obyek ditunjukkan dalam dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal.

2.7.4. Class Diagram

Class diagram menurut Munawar (2005 : 28) merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute/properties*. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi.

2.7.5. Collaboration Diagram

Collaboration diagram menurut Munawar (2005 : 101) adalah perluasan dari objek diagram. Objek diagram menunjukkan objek-objek yang hubungannya satu dengan yang lain. *Collaboration diagram* menunnjukkan *mesage-message* objek yang dikirim satu sama lain.

2.7.6. Component Diagram

Component diagram menurut Munawar (2005 : 119), mempresentasikan dunia riil item yaitu *component software*. *Component software* adalah bagian fisik dari sebuah sistem karena menetap di komputer. *Component diagram* mengandung *component*, *iterface* dan *relationship*.

2.7.7. Deployment Diagram

Deployment diagram menurut Munawar (2005 : 125), menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*.

2.7.8. Object Diagram

Object Diagram menurut Munawar (2005 : 135), gambaran obyek-obyek secara ringkas disebuah sistem pada suatu waktu. Obyek diagram sering disebut sebagai *instance diagram* karena menunjukan *instance-instance* dari *class*.