BAB II

LANDASAN TEORI

2.1. Aplikasi

2.1.1. Pengertian Aplikasi

Menurut Rizky Dhanta (2009:32), aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel. Sedangkan menurut Anisyah Nur Suci (2000:30), aplikasi adalah penerapan, penggunaan atau penambahan.

Dari pengertian diatas, dapat disimpulkan bahwa aplikasi adalah *software* yang berfungsi untuk membantu atau melakukan tugas-tugas tertentu seperti penerapan, penggunaan atau penambahan.

2.2. *Mobile Learning* (*M-Learning*)

2.2.1. Pengertian *Mobile Learning*

Mobile learning didefinisikan oleh **Clark Quinn [Quinn 2000]** sebagai : *The intersection of mobile computing and e-learning: accessible resources wherever you are, strong search capabilities, rich interaction, powerful support for effective learning, and performance-based assessment. ELearning independent of location in time or space***.** Berdasarkan definisi tersebut maka *mobile learning* merupakan model pembelajaran yang memanfaatkan teknologi informasi dan komunikasi yang ada saat

ini. Pada konsep pembelajaran tersebut mobile learning membawa manfaat ketersediaan materi pembelajaran yang dapat di akses setiap saat dan visualisasi materi yang menarik. Hal penting yang perlu di perhatikan bahwa tidak setiap materi pengajaran cocok memanfaatkan *mobile learning*.

Mobile learning dapat didefinisikan sebagai suatu fasilitas atau layanan yang memberikan informasi elektronik secara umum kepada pembelajar dan *content* edukasional yang membantu pencapaian pengetahuan tanpa mempermasalahkan lokasi dan waktu. Sistem *m-learning* ini memanfaatkan mobilitas dari perangkat *handheld/mobile*, seperti *handphone* dan PDA, untuk memberikan suatu fungsi pembelajaran yang dapat dilakukan di mana pun dan kapan pun.

2.2.2. Penerapan Mobile Learning

Penerapan *mobile learning* memang sangat cocok untuk pembelajaran, namun ada juga Materi ajar yang tidak cocok mengadopsi konsep *mobile learning* antara lain : materi yang bersifat ”*hands on*”, keterampilan sebagai mana dokter gigi, *interview skills, team work* seperti marketing maupun materi yang membutuhkan pengungkapan ekspresi seperti tarian.

Pengukuran terhadap readiness atau kesiapan merupakan aktivitas yang perlu dilakukan. Hal ini disebabkan karena kesiapan terkait dengan keberhasilan penerapan mobile learning. Dalam konteks penerapan *mobile learning* kesiapan dapat dipahami sebagai kemauan dan kemampuan untuk menyelenggarakan dan berpartisipasi dalam

mobile learning. *Mobile learning readiness* menyangkut semua *stake holder* yang terkait dengan penerapan *mobile learning* antara lain pembimbing, siswa dan lembaga pendidikan.

Pembimbing diharapkan memiliki kemauan untuk memanfaatkan teknologi informasi dan komunikasi. Kemauan untuk menerima teknologi informasi dan komunikasi menjadi pintu awal yang mempengaruhi faktor kesiapan lain yaitu *ICT literacy*. Kemauan menerima teknologi akan mempengaruhi terhadap kemauan untuk menggunakan dan mempelajari teknologi informasi dan komunikasi untuk diterapkan dalam proses belajar mengajar. *ICT literacy* merupakan kemampuan teknis dan kognitif yang dimiliki guru untuk menggunakan teknologi informasi dan komunikasi dalam proses belajar mengajar.

Siswa berperan sama pentingnya dengan pembimbing dalam proses pembelajaran. Kemajuan siswa untuk menerima teknologi juga merupakan dimensi kesiapan yang perlu diukur. Sedangkan dimensi kemampuan meliputi *ICT literacy*, media akses, dan daya beli siswa dalam mengakses materi pembelajaran. *ICT literacy* terkait dengan kemampuan teknis dan kognitif siswa dalam menggunakan teknologi informasi dan komunikasi.

2.2.3. Keunggulan dan Kekurangan *Mobile Learning*

Mobile leraning memiliki kenggulan dan kekurangan, diantaranya :

a. Keunggulan *mobile leraning*.

Meskipun saat ini *m-learning* masih berada pada tahap awal pengembangan serta relatif belum begitu mapan, namun, *m-learning* diperkirakan akan menjadi cukup pesat dalam jangka waktu dekat. Hal ini didukung oleh beberapa faktor :

1. Sarana makin banyak, murah dan canggih.
2. Perkembangan tekhnologi wireless / seluler (2G, 2.5G, 3G, 3.5G).
3. Tuntutan kebutuhan.

Sebuah penelitian juga menunjukan bahwa pembelajar cukup nyaman menatap tampilan layar perangkat yang relatif kecil dalam waktu dibawah 5 menit, terdapat beberapa kelebihan dan kekuirangan *m-Learning* dibandingkan dengan pembelajaran lain adalah:

A. Kelebihan *mobile learning*

1. Dapat digunakan dimana-pun pada waktu kapan-pun.
2. Kebanyakan divais bergerak memiliki harga yang relatif lebih murah dibanding harga PC desktop.
3. Ukuran perangkat yang kecil dan ringan daripada PC desktop.
4. Diperkirakan dapat mengikutsertakan lebih banyak pembelajar karena *m-Learning* memanfaatkan teknologi yang biasa digunakan dalam kehidupan sehari-hari.

Dalam pembelajaran *e-Learning*, independensi waktu dan tempat menjadi faktor penting yang sering ditekankan. Namun, dalam *e-Learning* tradisional kebutuhan minimum tetap sebuah PC yang memiliki konsekuensi bahwa independensi waktu dan tempat tidak sepenuhnya terpenuhi. Independensi ini masih belum dapat dipenuhi dengan penggunaan *notebook* (komputer portabel), karena independensi waktu dan tempat yang sesungguhnya berarti seseorang dapat belajar dimana-pun kapan-pun dia membutuhkan akses pada materi pembelajaran.

b. Kekurangan *mobile learning*.

Mobile learning merupakan salah satu alternatif yang potensial untuk memperluas akses pendidikan. Namun, belum banyak informasi mengenai pemanfaatan divais bergerak, khususnya telepon seluler, sebagai media pembelajaran. Hal ini patut disayangkan mengingat tingkat kepemilikan dan tingkat pemakaian yang sudah cukup tinggi ini kurang dimanfaatkan untuk diarahkan bagi pendidikan.

Selain itu, saat ini masih sangat sedikit upaya pengembangan konten-konten pembelajaran berbasis divais bergerak yang dapat diakses secara luas. Kebanyakan konten yang beredar di pasaran masih didominasi konten hiburan yang memiliki aspek pendidikan yang kurang serta kebanyakan adalah hasil produksi dari luar negeri yang memiliki latar budaya yang berbeda dengan negera kita. Kenyataan ini memunculkan kebutuhan akan adanya pengembangan-pengembangan konten/aplikasi berbasis divais bergerak yang lebih banyak, beragam, murah dan mudah diakses.

Faktor yang menjadi keterbatasan pemanfaatan *m-learning* banyak terkait dengan keterbatasan pada divais. Saat ini kebanyakan divais bergerak memiliki keterbatasan layar tampilan, kapasitas penyimpan dan keterbatasan daya. *m-learning* juga memiliki lingkungan pembelajaran yang agak berbeda dengan *e-learning* atau pembelajaran konvensional. Dalam *m-learning* pembelajar lebih banyak memanfaatkan *m-learning* pada waktu luang *(spare time)* atau waktu idle *(idle time)* sehingga waktu untuk mengakses belajar juga terbatas.

Hal ini menyebabkan konten pembelajaran harus dirancang secara khusus dan tidak dapat dengan serta merta diadopsi dari modul pembelajaran *e-learning* atau pembelajaran tradisional. Penelitian yang saat ini ada masih banyak meng-eksplorasi kepada aspek-aspek teknis pengembangan *software* dan belum mendalami aspek lain berkait masalah usabilitas maupun aspek pedagogis dan aspek-aspek lainnya, sehingga diperlukan adanya penelitian-penelitian lebih lanjut yang lebih spesifik. Penelitian ini menunjukkan bahwa teknologi Java dapat dimanfaatkan sebagai salah satu enabler dari pemanfaatan *m-learning*. Java juga termasuk kategori *software* yang independen terhadap platform dan perangkat sehingga lebih banyak divais yang dapat menjalankan aplikasi Java.

Meski memiliki beberapa kelebihan, *m-Learning* tidak akan sepenuhnya menggantikan *e-learning* tradisional. Hal ini dikarenakan *m-Learning* memiliki keterbatasan-keterbatasan terutama dari sisi perangkat / media belajarnya. Kekurangan *m-Learning* sendiri sebenarnya lambat laun akan dapat teratasi

khususnya dengan perkembangan teknologi yang semakin maju. Kecepatan prosesor pada divais semakin lama semakin baik, sedangkan kapasitas memori, terutama memori eksternal, saat ini semakin besar dan murah. Layar tampilan yang relatif kecil akan dapat teratasi dengan adanya kemampuan *device* untuk menampilkan tampilan keluaran ke TV maupun ke proyektor. Masalah media input/output yang terbatas (hanya terdiri beberapa tombol) akan teratasi dengan adanya teknologi layar sentuh (*touchscreen*) maupun *virtual keyboard*. Keterbatasan dalam ketersediaan catu daya akan dapat teratasi dengan pemanfaatan sumber daya alternatif yang praktis, mudah didapat dan mudah dibawa, seperti baterai cair, tenaga gerak manusia, tenaga matahari dan lain-lain.

2.3. Android



Gambar 2.1 Logo Android
(Sumber : http://developer.android.com/distribute/googleplay/promote/brand.html diakses 15 Mei 2013)

Android merupakan sistem operasi mobile berbasis kernel Linux yang dikembangkan oleh Android Inc dan kemudian diakuisisi oleh Google. Sistem operasi

ini bersifat *open source* sehingga para programmer dapat membuat oplikasi secara mudah. Kemunculan Sistem Operasi Android ditandai dengan hadirnya Android seri pertama, yaitu Android 1.5 atau seri Cupcake. Setelah itu dengan berbagai pengembangan, muncullah Sistem Operasi Android seri terbaru dengan berbagai pengembangan yang lebih baik lagi.

2.3.1. Sejarah Android

Sebelum Android Inc, dibeli oleh Google Inc, Perusahaan *software* kecil ini didirikan oleh para senior lulusan IT dan *Comunications* tepatnya pada bulan Oktober 2003, para pendirinya antara lain adalah: Andy Rubin, Rich Minner, Chris White dan Nick Sears. Android Inc, yakin akan membuat suatu mobile device yang sangat memahami usernya, dari sinilah benih benih kesuksesan Android mulai terlihat. Karena konsep Android yang sangat menakjubkan ini, Google mulai melirik lalu mencoba mengenal konsep Android yang membuat pikirannya yakin akan menggamparkan dunia. Lalu pada akhirnya Google Inc mengakuisisi seluruh saham Android Inc.

Ternyata keyakinan yang sangat mengkuatkan Google ini makin menambahkan tingkat masa depan gemilang device mobile Android, sejak saat inilah Android membuat platform Linux untuk Mobile Phone. Pada tahun 2007, Dibentuklah Open Handset yang bertujuan untuk pengembangan, beberapa perusahaan yang mengkorsumsinya diantaranya adalah : *Broadcom, Intel, Marvell*

Techonology, Nvidia , Qualcomm, Sprint-nextex, Google, HTC, LG, Motorola dan T-Mobile. Di saat inilah realize Android pertama yang berbasis Linux kernel versi 2.6.

Pada Tahun 2008 inilah HTC Dream menjadi pencicip pertama Android versi 1.0 (Astro), dan pada Akhir tahun 2008 bergabunglah 14 perusahaan yaitu: *Vodafone, ARM holdings, Toshiba, Atheros, Asustek, Sony Ericsson, Garmin Ltd, PacketVideo dan Softbank.*

2.3.2. Versi Android

Android OS dirilis perdana pada Tanggal 5 November 2007, yang kemudian versi-versi berikutnya telah tersedia untuk pengguna. Sekedar informasi telepon yang pertama kali menggunakan sistem operasi ini adalah HTC Dream yang dirilis pada Bulan Oktober 2008 yang menggunakan Android versi pertama, yang diikuti oleh versi-versi berikut ini.

1) **Android Versi 1.0**

   Perangkat Android pertama yang berjalan pada Android 1.0 adalah HTC Dream (G1), memasukkan Android 1.0 fitur berikut :

a. Web Browser untuk menunjukkan, zoom dan pan full HTML dan halaman web XHTML, beberapa halaman menunjukkan sebagai jendela, mendukung POP3, IMAP4, dan SMTP.
b. Android Market download aplikasi dan update.
c. Multitasking, instant messaging, GPS, Wi-Fi merupakan Bluetooth.

d. Tidak ada Microsoft Exchange Server, camcorder tidak, youtube video pemutar.
e. Dukungan Kamera: kekurangan dalam perubahan, kualitas resolusi kamera, white balance, dan lain-lain.
f. Folder izin untuk pengelompokan sekelompok ikon aplikasi ke dalam folder icon tunggal pada layar awal.
g. Pemberitahuan peringatan di status bar optioned dengan ringtone, waspada getaran atau dengan LED.
h. Sinkronisasi Google untuk banyak aplikasi.
i. Google Maps dengan Latitude (Street View serta citra satelit), arah mengemudi menggunakan GPS.
j. Voice Dialer memungkinkan panggilan dan menempatkan panggilan telepon tanpa mengetik nama atau nomor.
k. Aplikasi lainnya termasuk: Jam Alarm, Calculator, Dialer (Phone), layar Home (peluncur) , Foto (Galeri), dan Pengaturan.

2) Android Versi 1.1

Google merilis Android versi 1.1 pada 9 Maret 2009, Android pada 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)

Dua bulan setelah peluncuran Android versi 1.1 tepatnya pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan mengunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Fitur-fitur barunya yakni kemampuan merekam dan menonton video dengan modus kamere, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan teknologi Bluetooth A2DP, kemampuan terhubung otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

4) Android Versi 1.6 (Donut)

Android 1.6 (Donut) di rilis pada bulan September dengan menampilkan proses pencarian yang lebih baik dibanding versi 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 diintegrasikan, mendukung teknologi CDMA/EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine, kemampuan dial kontak dan teknologi text to change speech.

5) Android Versi 2.0/2.1 (Éclair)

3 Desember 2009 Google kembali meluncurkan 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 teknologi Bluetooth 2.1.

6) 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, intergrasi 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 akhir tahun 2010, Android versi 2.3 (Gingerbread) diluncurkan. Versi Android ini paling banyak pemakainya. 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 Versi 3.0/3.1 (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. Tapi intinya Android versi ini tidak jauh beda dengan Android versi sebelumnya, cuma ada sedikit penanaman fitur-fitur baru.

9) Android Versi 4.0 (ICS: Ice Cream Sandwich)

Rilis 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

10) Android Versi 4.1 (Jelly Bean)

Android Jelly Bean yaang diluncurkan pada acara Google I/O tahun lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat. Tak ketinggalan Google Now juga menjadi bagian yang diperbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7.

11) Android Versi 4.2 (Jelly Bean)

Fitur photo sphere untuk panaroma, daydream sebagai screensaver, power control, lock screen widget, menjalankan banyak user (dalam tablet saja), widget terbaru. Android 4.2 Pertama kali dikenalkan melalui LG Google Nexus 4.

2.3.3. The Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di *Dalvik Virtual Machine* (DVM) bukan di Java *Virtual Machine* (JVM), sebernarnya banyak persamaan dengan Java *Virtual Machine* (JVM) seperti Java ME (*Java Mobile Edition*), tetapi android menggunakan *Virtual Machine* sendiri yang diskostumisasi dan dirancang untuk memastikan bahwa beberapa fitur-fitur berjalan lebih efisien pada perangkat *mobile*.

Dalvik Virtual Machine (DVM) adalah "*register bases*" sementara java *Virtual Machine* (JVM) bersifat "*stack based*", DVM didesain dan ditulis oleh Dan Bornsten dan beberapa enginers Google lainnya. Jadi bisa dikatakan "*Dalvik* equal (Java) == False" *Dalvik Virtual Machine* menggunakan kernel linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C/C++ sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dan proses sistem dari kernel linux yang digunakan dalam Android tersebut.

Semua *hardware* yang berbasis android dijalankan dengan menggunakan *Virtual Machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi pernagkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *Executable* file, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The Executable* file diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan *tools* yang disediakan dalam SDK Android.

2.3.4. Android SDK (Software Development Kit)

SDK Android adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java.

2.3.5. ADT (Android Development Tools)



Gambar 2.2 Android Development Tools (ADT)

Android Development Tools (ADT) adalah plugin yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi project android, membuat GUI aplikasi,

dan menambakan komponen-kompenen yang lainnya, begitu juga kita dapat melakukan running aplikasi menggunakan Android SDK melalui Eclipse. Dengan ADT juga kita dapat melakukan pembuatan package android (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang.

Mengembangkan aplikasi android dengan menggunakan ADT di eclipse sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi android.

Berikut adalah versi ADT untuk eclipse yang sudah dirilis :

a. ADT 12.0.0 (Juli 2011)
b. ADT 11.0.0 (June 2011)
c. ADT 10.0.1 (March 2011)
d. ADT 10.0.0 (February 2011)
e. ADT 9.0.0 (January 2011)
f. ADT 8.0.1 (Desember 2010)
g. ADT 8.0.0 (Desember 2010)
h. ADT 0.9.9 (September 2010)
i. ADT 0.9.8 (September 2010)
j. ADT 0.9.7 (May 2010)
k. ADT 0.9.6 (March 2010)
l. ADT 0.9.5 (Desember 2009)
m. ADT 0.9.4 (Oktober 2009)

Semakin tinggi platform android yang kita gunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya munculnya platform baru diikuti oleh munculnya versi ADT yang terbaru. Untuk melakukan instalasi ADT di-elipse dapat dilakukan secara on-line maupun offline. Untuk download ADT ini dapat di http://developer.android.com/sdk/eclipse-adt.html

2.3.6. Arsitektur Android

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

1. *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.

2. *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 :

a) *Views*
b) *Content Provider*
c) *Resource Manager*
d) *Notification Manager*
e) *Activity Manager*

3. *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 :

a) *Libraries* media untuk pemutar media audio dan video.
b) *Libraries* untuk manajemen tampilan.
c) *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
d) *Libraries SQLite* untuk dukungan database.

e) *Libraries* SSL dan *WebKit* terintegrasi dengan web *browser* dan security.

f) *Libraries Live Webcore* mencakup modern web *browser* dengan engine embedded web view.

g) *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's.

4. *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:

a) *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.*

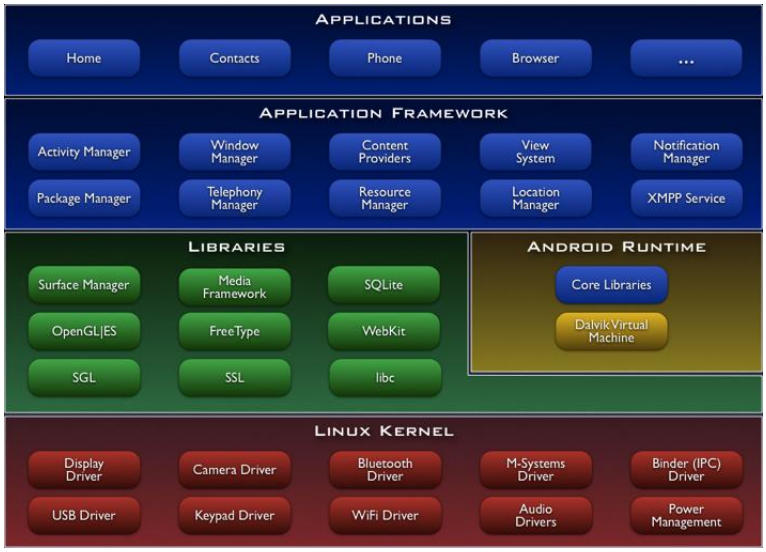
b) *Dalvik Virtual Machine*

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.

5. 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* 2.6.



Gambar 2.3 Arsitektur Android
(Sumber : http://www.candra.web.id/2011/01/30/arsitektur-android-os/ diakses 15 Mei 2013)

2.3.7. Fundamental Aplikasi

Aplikasi android ditulis dalam bahasa pemrograman java, kode java dikompilasi bersama data file *resource* yang dibutuhkan oleh aplikasi dimana prosesnya dipaket oleh *tools* yang dinamakan "apt *tools*" ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang sebenernya disebut dengan aplikasi yang dapat diinstal di perangkat *mobile* nantinya.

Ada empat jenis komponen pada aplikasi android yaitu :

a) Activites

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain dapat dilakukan dengan satu even misalnya click tombol, memilih opesi atau menggunakan triggers tertentu. Secara hirarki sebuah window *activity* dinyatakan dengan method *Activity*.setContentView(). ContentView adalah objek yang berada pada root hirarki.

b) Service

Service tidak memiliki visual *user interface* (UI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan music, *service* mungkin memainkan music atau mengambil data dari jaringan, tetapi setiap *service* haruslah berada dalam kelas induknya. Misalnya media player sedang memutar lagu dari *list* yang ada. Aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu atau menulis sms sambil *player* sedang jalan untuk menjaga *music* tetapi dijalankan, *activity player* dapat menjalankan *service* untuk membuat

aplikasi tetap berjalan. *Service* dijalankan pada *thread* utama dari proses aplikasi.

c) Broadcast Receiver

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *broadcast* seperti notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh kamera, atau pengubah referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi *broadcast* misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah didownload ke perangkat dan siap untuk digunakan *Broadcast Receiver* tidak memiliki *user interface* (UI) tetapi memiliki sebuah *activity* untuk merespon informasil yang mereka terima atau mungkin menggunakan *notification* manager untuk memberitahu kepada pengguna seperti lampu latar atau *viberating* (getaran) perangkat dan lain sebagainya.

d) Content Receiver

Content Provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti database SQLite.*Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika menggunakan aplikasi yang membutuhkan peta (MAP) atau aplikasi yang membutuhkan untuk mengkases data kontak dan navigasi, maka disinilah fungsi *content provider*.

2.4. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat 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 ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

2.4.1. Diagram UML

UML merupakan diagram yang saling terkait oleh karena itu perlu adanya kekonsistenan rancangan diagram yang satu dengan lainnya, bukan asal menggambar. Berikut akan dijelaskan 4 macam diagram yang sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu *use case* diagram, *sequence* diagram, *collaboration* diagram, dan *class* diagram.

1) *Use Case* Diagram

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan didalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.

2) *Sequence* Diagram

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

3) *Collaboration* Diagram

Collaboration diagram dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan *sequence* diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration* diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.

4) *Class* Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

2.5. Tinjauan Perangkat Lunak Pendukung

Perangkat pendukung yang penulis pergunakan adalah Java Platform Android, Eclipse, XML, Rational Rose, SQLite, Adobe Photoshop.

2.5.1. *Java Platform* Android

Aplikasi yang dibuat dan dikembangkan dalam penulisan ini menggunakan bahasa pemrograman berbasis java untuk platform android. Platform java ini mirip dengan Java 2 Standard Edition (J2SE) dikarenakan struktur perintah untuk kondisi dan perulangan, tipe variabel yang dapat digunakan, dan operator yang digunakan. Platform ini dapat dikatakan pengembangan dari J2SE yang diterapkan untuk aplikasi

mobile yang berbeda dari J2ME (Java 2 Micro Edition), karena file yang dihasilkan berekestensi .apk bukan .jar.

2.5.1.1. Gambaran *Java Platform* Android

Pada android, penggunaan bahasa pemrograman java tidak berbeda dengan java untuk platform lainya, yaitu membutuhkan dua buah produk yang dikeluarkan untuk membantu dalam membuat aplikasi, yaitu :

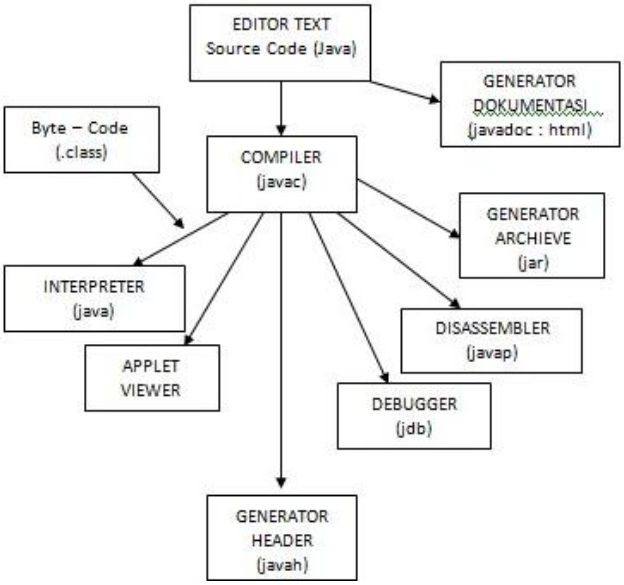
1. *Java SE Runtime Environment* (JRE)

Java Runtime Environment (JRE) menyediakan perpustakaan, Java Virtual Machine (JVM), dan komponen lain untuk menjalankan applet dan aplikasi yang ditulis dengan bahasa pemrograman java. Selain itu, terdapat dua buah kunci teknologi yang merupakan bagian dari JRE, yaitu: Java Plug-in, yang memungkinkan menjalankan applet di browser populer dan Java Web Start, yang menyebarkan aplikasi mandiri melalui jaringan. JRE tidak mengandung utilitas seperti compiler atau debugger untuk mengembangkan applet dan aplikasi. JRE yang digunakan adalah JRE6.

2. *Java Development Kit* (JDK)

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk menajemen dan membangun berbagai aplikasi java. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE ditambahkan compiler dan debugger yang diperlukan untuk mengembangkan applet dan aplikasi. JDK yang mendukung aplikasi android adalah JDK1.6.0_24 (JDK 1.6 update 24). Pada JDK,

terdapat 25 berbagai tools yang digunakan untuk membangun aplikasi java. Tools tersebut digambarkan dengan diagram berikut :



Gambar 2.4 Tools Dasar JDK
(Sumber : http://automotivehunter.blogspot.com/2012/08/java-development-kit jdk.html di akses 15 Mei 2013)

2.5.2. 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 :

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

c. 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.

2.5.2.1. Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM *Visual Age for Java* 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta untuk pengembangannya. Semenjak itu konsursium Eclipse *Foundation* mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

"The future of mobile computing is becoming even more exciting. Mobile devices are continually growing more capable, especially with the advent of cleverly integrated phone capabilities. With better and better wireless networks capable of transferring media in real time, an entirely new breed of applications is now possible". (Ashwin Kumar:2012:1).

2.5.2.2. Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform* (RCP). Berikut ini adalah komponen yang membentuk RCP :

a. Core platform.

b. OSGi.

c. SWT (*Standard Widget Toolkit*).

d. JFace.

e. Eclipse *Workbench.*

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools), plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan plug-in baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

2.5.3. Adobe Photoshop

Adobe Photoshop, atau biasa disebut Photoshop, adalah perangkat lunak editor citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (market leader) untuk perangkat lunak pengolah gambar/foto, dan, bersama Adobe Acrobat, dianggap sebagai produk terbaik yang pernah diproduksi oleh Adobe Systems. Versi kedelapan aplikasi ini disebut dengan nama Photoshop CS (Creative Suite), versi sembilan disebut Adobe Photoshop CS2, versi sepuluh disebut Adobe Photoshop CS3 , versi kesebelas adalah Adobe Photoshop CS4, versi keduabelas adalah Adobe Photoshop CS5 dan versi yang terakhir (ketigabelas) adalah Adobe Photoshop CS6.

2.5.4. XML

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet. Teknologi XML adalah teknologi keturunan dari SGML (*Standard Generalized Markup Language, ISO 8879*) yang dikembangkan pada tahun 1980-an. XML adalah suatu bahasa Markup, yaitu bahasa yang berisikan kode-kode berupa tanda-tanda tertentu dengan aturan tertentu untuk memformat dokumen teks dengan tag sendiri agar dapat dimengerti. XML adalah language untuk mengidentifikasi dan menandai data terstruktur.

XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang view database, tetapi dengan suatu cara yang standar. XML memiliki tiga tipe file, diantaranya :

1. XML, merupakan standar format dari struktur berkas (file).
2. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. XSD, merupakan standar yang mendefinisikan struktur database dalam XML.

Keunggulan XML bisa diringkas sebagai berikut :

1. Pintar *(Intelligence)*. XML dapat menangani berbagai tingkat (level) kompleksitas.
2. Dapat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri. Seperti *Microsoft* membuaat bahasa MSXML atau Macromedia mengembangkan MXML.
3. Mudah pemeliharaannya.
4. Sederhana. XML lebih sederhana.
5. Mudah dipindah-pindahkan *(Portability)*. XML mempunyai kemudahan perpindahan *(portabilitas)* yang lebih bagus.

2.5.5. Rational Rose

Rational Rose berfungsi sebagai *tool* untuk pemodelan sistem yang menggambarkan proses-proses yang ada pada sistem ini. *Rational Rose* adalah kakas (*tools)* pemodelan visual untuk pengembangan system berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem. *Rational Rose* digunakan untuk melakukan pemodelan sistem sebelum pengembang menulis kode-kode dalam bahasa pemrograman tertentu. Ia juga membantu analisis system dengan cara pengembang membuat diagram *use case* untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna. Kemudian, ia juga menuntut pengembang untuk mengembangkan *Interaction* diagram untuk melihat bagaimana objek-objek saling bekerjasama dalam menyediakan fungsionalitas yang diperlukan. (Adi Nugroho, 2005:20).

2.5.6. SQLite

Menurut Nazaruddin Safaat android memiliki fasilitas untuk membuat *database* yang dikenal dengan *SQLite* yaitu salah satu *software* yang *embedded* yang sangat popular, kombinasi *SQL interface* dan penggunaan memory yang sangat sedikit dengan kecepatan yang sangat cepat. *SQLite* di android termasuk dalam *Android runtime*, sehingga setiap versi dari android dapat membuat *database* dengan *SQLite* (Nazaruddin:2011).

Dalam sistem android terdapat beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai berikut :

1. *Shared prefences* yaitu menyimpan data beberapa nilai (*value*) dalam bentuk *groups key* yang dikenal dengan *prefences*.
2. *Files* yaitu menyimpan data dalam *file*, dapat berupa menulis ke *file* atau membaca dari file.
3. *SQLite Database*, yaitu menyimpan data dalam bentuk *database.*
4. *Content Providers*, yaitu menyimpan data dalam bentuk *content providers service.*