BAB II

LANDASAN TEORI

2.1 Perangkat Lunak Aplikasi

2.1.1 Pengertian 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*).

2.1.2 Aplikasi Mobile

Aplikasi *mobile* adalah sebuah aplikasi yang memungkinkan Anda melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau *handphone*. Dengan menggunakan aplikasi *mobile*, Anda dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya. Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemari oleh hampir 70% pengguna telepon seluler, karena dengan memanfaatkan adanya fitur *game*,

music player, sampai *video player* membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun.

2.2 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Android pertama kali dikembangkan oleh perusahaan bernama Android Inc yang kemudian pada tahun 2005 diakuisisi oleh raksasa Internet Google. Android dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap release-nya diberi kode nama berdasarkan nama hidangan makanan. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-*Mobile*, dan Nvidia.

Keunggulan utama Android adalah gratis dan *open source*, yang membuat smartphone Android dijual lebih murah dibandingkan dengan Blackberry atau iPhone meski fitur (*hardware*) yang ditawarkan Android lebih baik. Beberapa fitur utama dari Android antara lain WiFi hotspot, Multi-touch, Multitasking, GPS, support java, mendukung banyak jaringan (GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, and WiMAX) dan juga kemampuan dasar *handphone* pada umumnya

2.2.1 Android

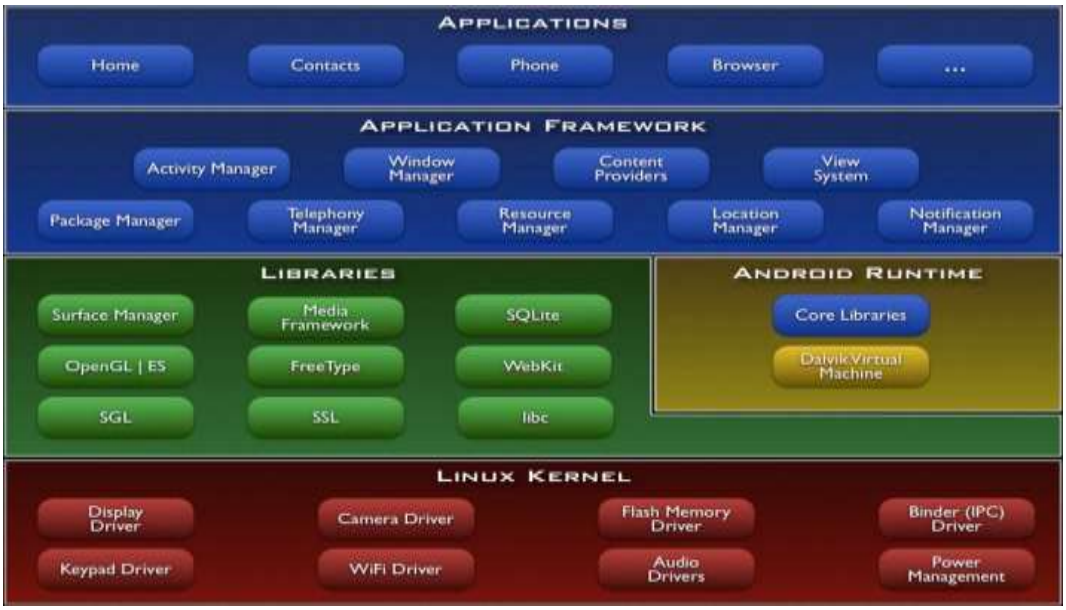
Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc.

Bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. 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 bersiap menghadapi persaingan dalam pasar telepon seluler. versi android terbaru yaitu versi 3.0. Android juga sudah bergabung dengan beberapa smart *Mobile* seperti Nokia, Sony Ericsson, dan lainnya.

1. Android Versi 1.1
2. Android Versi 1.5 (Cupcake)
3. Android Versi 1.6 (Donut)
4. Android Versi 2.0/2.1 (Eclair)
5. Android Versi 2.2 (Froyo: Frozen Yoghurt)
6. Android Versi 2.3 (Gingerbread)
7. Android Versi 3.0 (Honeycomb)
8. Android Versi 4.0 Ice Cream Sandwich
9. Android Versi 4.1 (Jelly Bean)

2.2.2 Anatomi Aplikasi Android

Dalam paket sistem operasi Android tediri dari beberapa unsur seperti tampak pada gambar 2.1. Secara sederhana arsitektur android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *Framework* yang menyediakan dan mengatur alur proses aplikasi.



Gambar 2.1 *Detail* Anatomi Android

2.2.2.1 *Linux Kernel*

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

2.2.2.2 *Libraries*

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

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

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

2.2.2.3 Android *Runtime*

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

1. Pustaka Inti, android dikembangkan melalui bahasa pemrograman Java, tapi *Android Runtime* bukanlah mesin virtual Java. Pustaka inti android menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus Android.

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

2.2.2.4 *Application Framework*

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generic untuk mengakses perangkat, serta mengatur tampilan *User Interface* dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut:

1. *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”*Backstack“* untuk navigasi penggunaan
2. *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
3. *Resuource Manager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan *file layout*.
4. *Location Manager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.

5. *Notification Manager*, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada *status bar.*

2.2.2.5 *Application Layer*

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

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

2.2.3 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.

2.2.3.1 *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.

2.2.3.2 *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.

2.2.3.3 *Intens*

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.

2.2.3.4 *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.

2.2.3.5 *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.

2.2.4 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.

2.2.5 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.

2.3 Multimedia

2.3.1 Pengertian Multimedia

Multimedia adalah kombinasi dari komputer dan video atau multimedia secara umum merupakan kombinasi tiga lapisan, yaitu: suara, gambar dan teks, atau multimedia adalah kombinasi dari paling sedikit dua menjadi input atau

output dari data, media ini dapat audio (suara, musik), animasi, video, teks, grafis dan gambar.

Multimedia merupakan alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafis, animasi, audio, dan gambar video. Definisi yang lain dari multimedia, yaitu dengan menempatkan dalam konteks, seperti yang dilakukan oleh Hofstetter, multimedia adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, grafis, audio, gambar bergerak (video dan animasi) dengan menggabungkan link dan tool yang memungkinkan pemakai melakukan navigasi, berinteraksi, berkreasi dan berkomunikasi.

Dalam definisi ini terkandung empat komponen penting multimedia. Pertama harus ada komputer yang mengkoordinasikan apa yang dilihat dan didengar. Yang berinteraksi dengan kedua, harus ada link yang menghubungkan kita dengan informasi. Ketiga, harus ada alat navigasi yang memandu kita, menjelajah jaringan informasi yang saling terhubung. Kempat, multimedia menyediakan tempat kepada kita untuk mengumpulkan, memproses, dan mengkomunikasikan informasi dan ide kita sendiri, jika salah satu komponen tidak ada. Maka bukan multimedia dalam arti yang luas namanya. Misalnya, jika tidak ada komputer untuk berinteraksi, maka itu namanya media campuran, bukan multimedia. Jika tidak ada link yang menghadirkan sebuah struktur dan dimensi, maka namanya rak buku, bukan multimedia.

Kalau tidak ada alat navigasi yang memungkinkan kita memilih jalannya suatu tindakan maka itu namanya film, bukan multimedia. Demikian juga jika kita

tidak mempunyai ruang untuk berkreasi dan menyumbangkan ide sendiri, maka namanya televisi, bukan multimedia. Dari beberapa definisi diatas, maka multimedia ada yang online (internet) dan multimedia ada yang offline (tradisional).

Beberapa definisi menurut beberapa ahli:

1. Kombinasi dari komputer dan video (Rosch, 1996)
2. Kombinasi dari tiga elemen: suara, gambar, dan teks (McComick, 1996)
3. Kombinasi dari paling sedikit dua media input atau output. Media ini dapat berupa audio (suara, musik), animasi, video, teks, grafik dan gambar (Turban dan kawan-kawan, 2002)
4. Alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio dan video (Robin dan Linda, 2001)
5. Multimedia dalam konteks komputer menurut Hofstetter 2001 adalah: pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, audio, video, dengan menggunakan tool yang memungkinkan pemakai berinteraksi, berkreasi, dan berkomunikasi.

2.3.2 Multimedia Interaktif

Jenis multimedia dapat dibedakan menjadi dua, yaitu : multimedia linier dan multimedia interaktif. Multimedia linier adalah suatu multimedia yang tidak dilengkapi dengan alat pengontrol apapun yang dapat dioperasikan oleh user. Multimedia ini berjalan sekuensial (berurutan), contohnya : TV dan film. Sedangkan multimedia interaktif adalah suatu multimedia yang dilengkapi dengan

alat pengontrol yang dapat dioperasikan oleh user, sehingga user dapat memilih apa yang dikehendaki untuk proses selanjutnya. Contoh multimedia interaktif adalah multimedia pembelajaran interaktif, aplikasi game, dan lain-lain.

2.3.3 Kelebihan Multimedia Dalam Pembelajaran

Berbagai kelebihan multimedia pembelajaran berikut mampu mengatasi kelemahan-kelemahan yang terdapat di dalam pengajaran konvensional :

1. Tahap interaktif dengan bantuan *hardware*, multimedia dapat diprogram sesuai dengan kebutuhan dan prinsip-prinsip suatu media dalam penyampaian suatu pesan sehingga pesan yang terkandung dapat tercapai. Dalam pemakaian suatu multimedia, pebelajar dapat mengawal sendiri arah dan alur pelajaran yang hendak dipelajarinya. Dari sini akan tercipta interaksi pebelajar dengan media yang dipergunakan.
2. Pengajaran berdasarkan kemampuan pebelajar materi yang dipergunakan bisa untuk melayani setiap individu berdasarkan kemampuan peserta didik. Pebelajar akan mengalami pengalaman pembelajaran yang berbeda karena pebelajar dapat mengatur kapan suatu materi isi dari multimedia akan dipelajarinya.
3. Menarik dan menambah minat terhadap materi multimedia mempunyai kandungan berupa gabungan dari elemen-elemen seperti teks, grafik, animasi, bunyi, dan video sebagai rangsangan supaya menarik perhatian dan minat pelajar terhadap isi pelajaran yang disampaikan. pengguna atau users dapat melihat, mendengar, berinteraksi dengan cara bernavigasi pada tombol, hyperlink dan tools navigasi lainnya yang terdapat dalam program Dengan

pemakaian media yang cocok dan menarik maka pebelajar dapat memiliki minat untuk mempelajari isi atau pesan dari multimedia tersebut.

2.4 *Mobile Learning*

Andreas H, N Alexander, dan M Matthias (2004), menyatakan *bahwa Mobile Learning (M-Learning)* adalah generasi berikutnya *E-Learning* dan berdasarkan pada perangkat mobile. Satu keuntungan adalah ketersediaan tinggi dari perangkat tersebut: penetrasi pasar ponsel di Austria saat ini pada tingkat 81% dan jumlahnya terus bertambah. Hal ini dapat ditekankan bahwa mayoritas penduduk memiliki ponsel yang mereka miliki di tangan sebagian besar waktu. Akibatnya, *M-Learning* akan menjadi instrumen penting untuk belajar sepanjang masa.

Seperti halnya dengan pelaksanaan skema inovatif, tantangan teknis dan administrasi yang signifikan akan dihadapi. Ini akan bertemu bersama dengan tantangan yang lebih tak jelas: 'Bagaimana penggunaan teknologi mobile saat ini membantu pendidik untuk merangkul pendekatan yang berpusat pada pembelajar yang benar-benar belajar?". Transfer dari *E-learning* untuk revolusi *M-learning* telah disertai dengan perubahan terminologi: 'multimedia' sekarang memberikan cara belajar objek untuk", 'interaktif' menjadi 'spontan', sebagaimana digambarkan dalam tabel.

Tabel 2.1 Perbandingan Terminology Antara *E-Leraning* dan *M-Learning*

E-Learning	**M-Learning**
Computer	Mobile
Bandwidth	GPRS, G3, Bluetooth

Multimedia	Objects
Interactive	Spontaneous
Hyperlinked	Connected
Collaborative	Networked
Media-rich	Lightweight
Distance learning	Situated learning
More formal	Informal
Simulated situation	Realistic situation
Hyperlearning	Constructivism,

Perbedaan utama pedagogis antara kedua disiplin ini perbedaan antara teks dan instruksi berbasis grafis untuk lebih suara dan grafis dan petunjuk berbasis animasi. Dimana belajar sebelumnya terjadi di depan komputer, di kelas, laboratorium, atau di rumah, sekarang diaktifkan untuk terjadi di lapangan, atau pada lokasi di mana perangkat mobile yang berfungsi penuh.

Dalam pengembangan m-learning, terdapat lima kategori dasar teknologi yang harus dipertimbangkan ketika mengimplementasi pengembangan m-learning yaitu, transportasi, platform, pengiriman, media teknologi, dan pengembangan bahasa. Sebuah model untuk penggunaan teknologi yang efektif untuk mengajar di perguruan tinggi terdapat delapan kriteria yang akan digunakan dalam menentukan pilihan teknologi. Sebuah penelitian apakah teknologi yang tepat

telah dipilih ini bisa dibilang merupakan aspek penting dari sebuah evaluasi menyeluruh pembelajaran mobile.

2.5 Konsep Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah mempermudah *programmer* didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Suatu perusahaan *software* yaitu *Rational Software*, telah membentuk konsarium dengan berbagai organisasi untuk meresmikan pemakaian *Unifed Modelling Language (UML)* sebagai bahasa standar dalam *Object Oriented Analysist Design (OOAD).*

2.5.1 Unified Modelling Language (UML)

UML dalam sebuah bahasa untuk menentukan visualisasi, konstruksi, dan mendokumentasikan artifacts dari sistem software, untuk memodelkan bisnis, dan sistem non-software lainnya. UML merupakan sistem arsitektur yang bekerja dalam OOAD dengan satu bahasa yang konsisten untuk menentukan, visualisasi, konstruksi dan mendokumentasikan artifact yang terdapat dalam sistem. Artifact adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa software. Artifact dapat berupa model, deskripsi atau software.

2.5.1.1 Kelebihan UML

Kelebihan UML dibandingkan dengan bahasa permodelan yang lain antara lain:

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk bahasa pemodelan.
5. Memadukan praktek-praktek terbaik di industri perangkat lunak menjadi terminologi dan notasi yang diterima luas.
6. Menyediakan kemampuan merepresentasikan semua konsep yang relevan untuk sistem perangkat lunak.
7. Menyediakan fleksibilitas yang diperlukan bagi konsep-konsep perangkat lunak yang baru.

2.6 Alat Musik

2.6.1 Pengertian Alat Musik

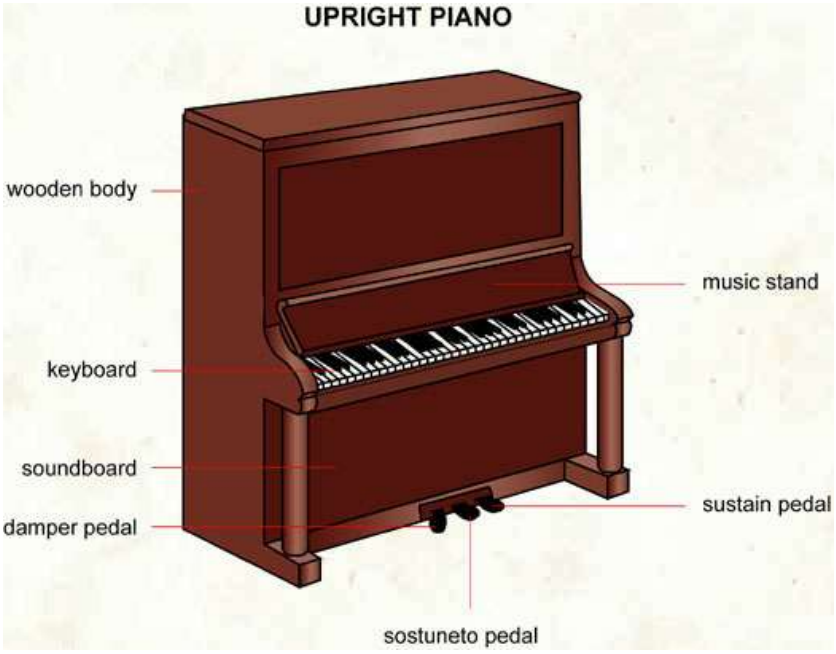
Alat musik merupakan suatu instrumen yang dibuat atau dimodifikasi untuk tujuan menghasilkan musik. Pada prinsipnya, segala sesuatu yang memproduksi suara, dan dengan cara tertentu bisa diatur oleh musisi, dapat disebut sebagai alat musik. Walaupun demikian, istilah ini umumnya diperuntukkan bagi alat yang

khusus ditujukan untuk musik. Bidang ilmu yang mempelajari alat musik disebut organologi.

2.6.2 Piano

Piano adalah sebuah alat musik yang banyak digemari. Ada berbagai alasan mengapa orang menyukainya. Melodi, suara, lagu yang dimainkan di piano, mungkin itu adalah beberapa alasan yang ada di benak seseorang.

2.6.2.1 Bagian-Bagian Piano



Gambar 2.2 Bagian-Bagian Piano

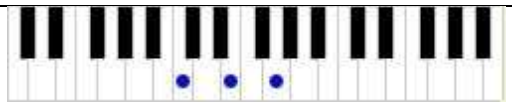
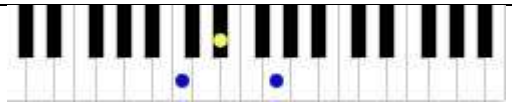
Bagian-bagian alat musik piano diantaranya yaitu:

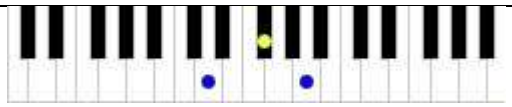
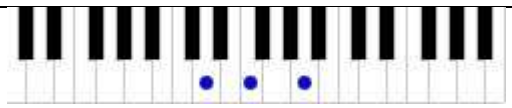
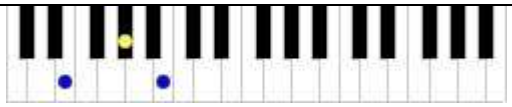
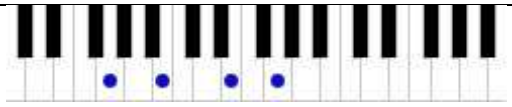
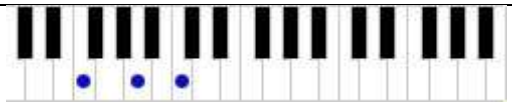
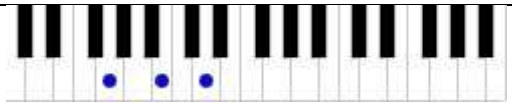
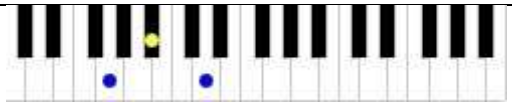
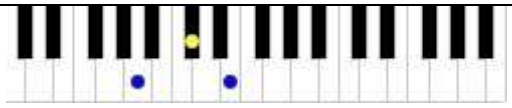
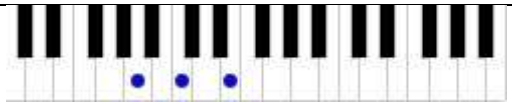
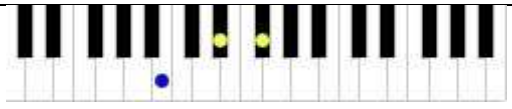
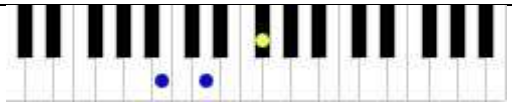
1. Tuts Piano / Keyboard: tuts piano terdiri dari dua bagian, yaitu tuts hitam dan tuts putih. pada awalnya, piano hanya memiliki satu tuts saja yaitu tuts putih namun karena kurang peraktis maka tuts hitampun ditambahkan menjadi bagian utama piano. Perbedaan tuts hitam dan tuts putih hanya pada posisinya saja.

2. Martil: bagian martil berfungsi untuk memukul dawai pada piano ketika anda menekan tuts piano untuk menghasilkan suara.
3. Dawai Piano: dawai inilah yang menghasilkan suara ketika dipukul oleh martil. Dawai piano harus sering distem agar tinggi-rendahnya nada tetap standar dan suaranya tidak fals.
4. Soundboard: bagian ini adalah bagian yang menyebabkan gema pada piano. Jika martil memukul dawai piano maka akan dihasilkan suara, suara tersebut akan diteruskan keruang kecil yang disediakan. Ruang gema inilah yang menentukan keras lembutnya suara serta jenis suara yang dihasilkan.
5. Pedal: Pada piano biasanya terdapat 2 hingga 3 pedal. Pedal tersebut diantaranya yaitu:
   a) Pedal sustain: untuk membuat nada bergema lebih lama.
   b) Pedal sustuneto: melembutkan suara.
   c) Pedal damper: mendekatkan jarak martil dengan dawai.
6. Music Stand: Berfungsi untuk menempatkan not balok.

2.6.2.2 Chord Dasar Piano

Tabel 2.2 Chord Dasar Piano (Major dan Minor)

Chord	Fringer Chart
C	
Cm	
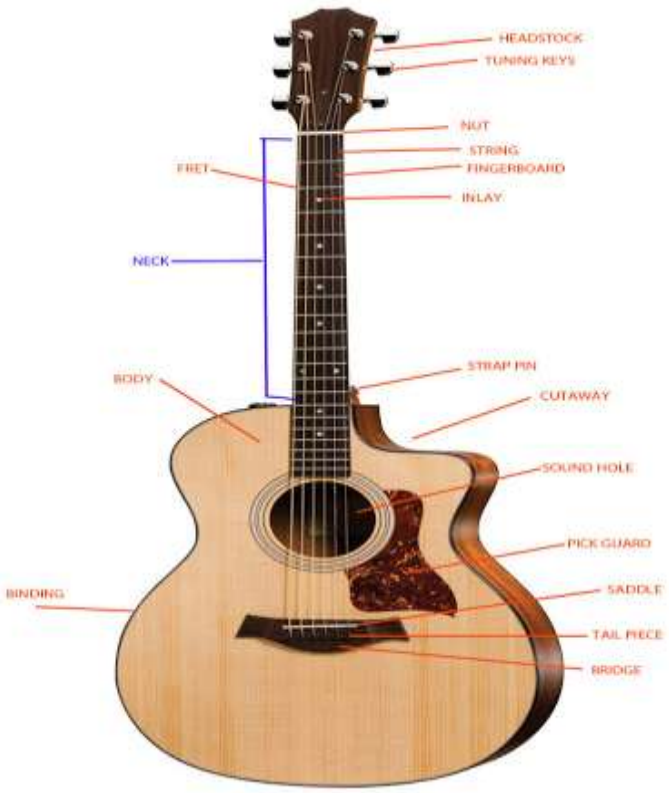
D	
Dm	
E	
Em	
F	
Fm	
G	
Gm	
A	
Am	
B	
Bm	

2.6.3 Gitar

Gitar adalah sebuah alat musik berdawai yang dimainkan dengan cara dipetik, umumnya menggunakan jari maupun plektrum. Gitar terbentuk atas sebuah bagian tubuh pokok dengan bagian leher yang padat sebagai tempat senar yang umumnya berjumlah enam didempetkan. Gitar secara tradisional dibentuk dari berbagai jenis kayu dengan senar yang terbuat dari nilon maupun baja. Beberapa gitar modern dibuat dari material polikarbonat.

2.6.3.1 Bagian-Bagian Gitar

Pada dasarnya sebuah gitar mempunyai 3 anatomi dasar hampir menyerupai anatomi utama tubuh manusia, yaitu head (bagian kepala dari gitar), neck (bagian leher dari gitar) dan body (bagian tubuh dari gitar).

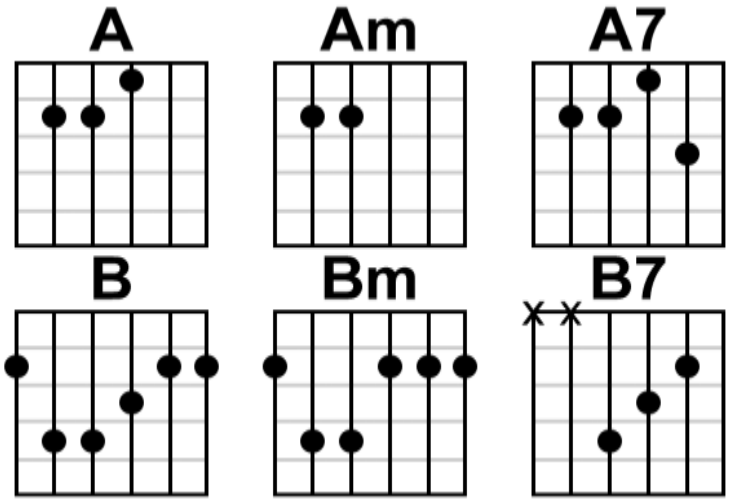


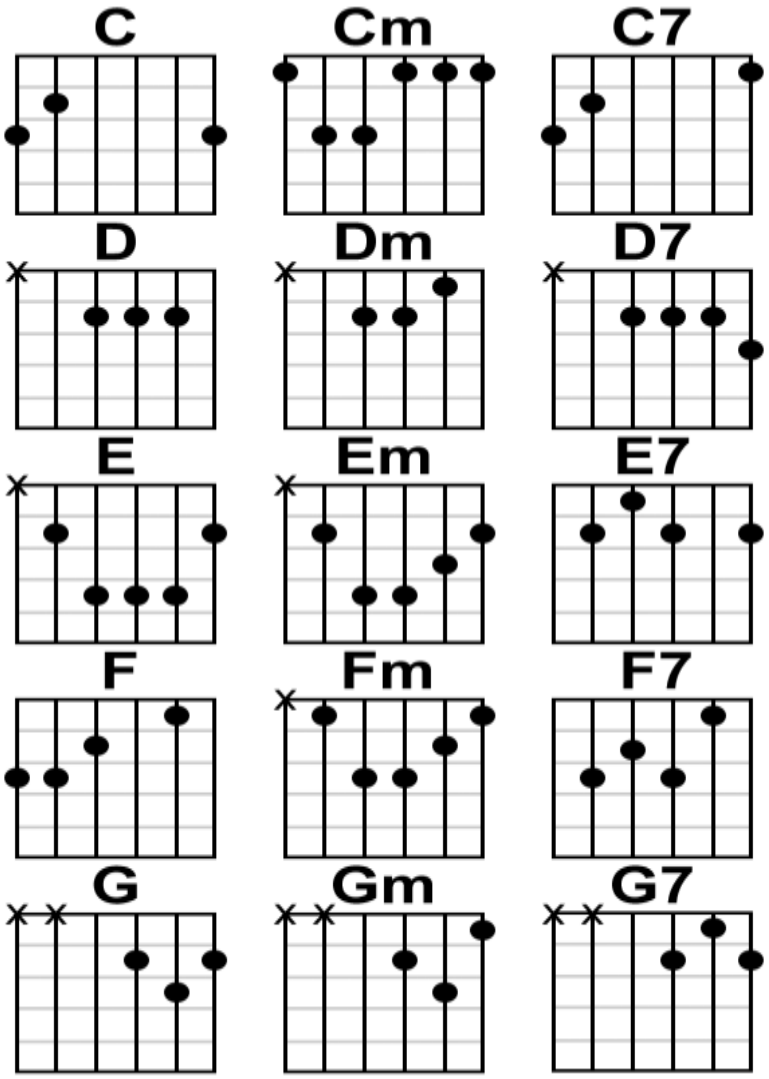
Gambar 2.3 Bagian-Bagian Gitar

1. Headstock, adalah "kepala gitar" yang berfungsi untuk tempat pemasangan tuning key.
2. Tuning keys/tuning machine/tuner, sering juga disebut dryer, adalah unit yang berfungsi untuk mengatur ketegangan dawai sehingga sesuai dengan nada yang seharusnya (untuk men-stem).
3. Nut, yaitu unit yang berfungsi untuk bantalan dan pengatur ketinggian dawai pada neck agar dawai tidak menyentuh fret, sehingga dawai dapat bergetar.
4. Neck, atau leher gitar, merupakan bagian gitar yang berfungsi sebagai tempat dipasangnya fretboard.
5. Fingerboard/Fretboard, yaitu papan tipis (kurang lebih 5 mm) selebar neck yang ditempelkan pada neck, berfungsi sebagai tempat untuk memasang fret.
6. Fret, adalah kawat khusus (biasanya terbuat dari tembaga atau stainless steel) yang dipasang pada fretboard, fungsinya adalah untuk menghaslkan tingkat ketinggian nada yang berbeda dengan cara menempatkan jari-jari pada ruang-ruang di antara logam-logam fret.
7. Inlay/fret marker, yaitu tanda yang ditempatkan pada fretboard yang berfungsi sebagai alat bantu gitaris untuk menentukan fret yang harus ia tekan untuk menghasilkan nada tertentu.
8. Dawai/string, sering disebut senar, yaitu kawat/nilon tipis yang direntangkan dari headstock sampai bridge gitar, yang berfungsi untuk menghasilkan suara.
9. Body, atau badan gitar, pada gitar akustik body gitar berfungsi sebagai penguat getaran dawai.

10. Binding, adalah lis kecil yang mengitari body, neck, atau headstock gitar, fungsinya untuk memperindah tampilan gitar
11. Pick Guard/Scratch Guard, yaitu papan plastik yang berfungsi untuk melindungi body gitar dari goresan-goresan yang timbul saat kita memainkan gitar
12. Sound hole, adalah lubang pada body gitar akustik yang befungsi sebagai tempat keluarnya suara
13. Saddle, adalah plastik (bisa juga logam atau tulang) yang fungsinya sama dengan nut namun terlketak pada bridge gitar
14. Bridge, yaitu tempat untuk memasang saddle
15. Tailpiece, adalah bagian gitar yang berfungsi untuk memegang ujung senar
16. Cutaway, adalah coakan yang terdapat pada ujung body gitar yang berfungsi untuk mempermudah gitaris bermain pada fret-fret yang berada di pangkal neck
17. Strap pin, adalah tempat untuk mengaitkan strap (sabuk menggantung gitar)

2.6.3.2 Kord Dasar Gitar





Gambar 2.4 Chord Dasar Gitar (Major dan Minor)

2.6.4 Drum

Drum adalah kelompok alat musik perkusi yang terdiri dari kulit yang direntangkan dan dipukul dengan tangan atau sebuah batang. Selain kulit, drum juga digunakan dari bahan lain, misalnya plastik. Drum terdapat di seluruh dunia dan memiliki banyak jenis, misalnya kendang, timpani, Bodhrán, Ashiko, snare drum, bass drum, tom-tom, beduk, dan lain-lain.

Dalam musik pop, rock, dan jazz, drums biasanya mengacu kepada drum kit atau drum set, yaitu sekelompok drum yang biasanya terdiri dari snare drum, tom-tom, bass drum, cymbal, hi-hat, dan kadang ditambah berbagai alat musik drum listrik. Orang yang memainkan drum set disebut "drummer".

2.6.5.1 Bagian-Bagian Drum

Drum sebenarnya bermacam-macam diantaranya : snare, tom-tom, bass, conga, tymbal, mondo, bedug, tabla dan lain-lain. mereka sebenarnya adalah drum, karena memainkannya dengan cara dipukul. Tetapi yang kita bahas adalah drumset, yang bisa dibilang bentuk drum modern. Drumset itu sendiri sebenarnya terdiri atas 3 bagian, yaitu:

1. Tom-tom terdiri atas berbagai macam ukuran baik dalam kedalamannya dan diameternya. Ukuran suatu drum biasnya ditulis 12x10 yang maksudnya adalah kedalamannya 12 inchi dan diameternya 10 inchi. Diameter tom-tom bervariasi, biasanya tom-tom paling kecil berdiameter 6", dan berlanjut ke 8", 10", 12", 13", 14", 15", 16", 18" dan 20".
2. Bass drum tidak terlalu berbeda dengan tom-tom, hanya bass drum mempunyai diameter yang lebih besar, 16", 18", 20", 22", 24" dan bahkan 26" atau lebih. Dan bass drum dipukul dengan menggunakan pedal dan ditaruh dibawah. Tetapi suara bass drum tidak seperti tom-tom yang bersuara "Dung..." tetapi cenderung bersuara "Dug..." (lebih mati suaranya). Kayu bass drum cenderung lebih tebal untuk menghasilkan suara yang lebih keras dan untuk ketahanan drum itu sendiri.

3. Snare drum adalah drum yang paling berbeda diantara lainnya (dari bentuk dan suaranya). Dan snare drum merupakan unsur utama dari drumset (yang paling sering dipukul). Drum ini biasanya berukuran 10" sampai 15", tetapi yang paling biasa digunakan adalah ukuran 14". Yang membuat perbedaan pada snare drum yaitu pada bagian bawah drum tersebut. Di bawahnya menggunakan kawat-kawat yang berbentuk spiral atau yang sebenarnya dinamakan Snare Wire /Strainer.



Gambar 2.5 Snare Drum

4. Cymbal, lagi-lagi merupakan 'nyawa' bagi drumset, karena hampir tidak mungkin bermain drum tanpa cymbal (ibaratnya seperti makan nasi tanpa nasi, nggak makan donk...). Cymbal terdiri atas 4 jenis mereka yaitu:



Gambar 2.6 Cymbal Drum

a) Hihat cymbal: 'Jantungnya' cymbal dan drum. Berguna untuk menjaga waktu/tempo. terdiri atas sepasang cymbal. berukuran 8" sampai 15". Ukuran standart 14"

b) Ride cymbal: Sama fungsinya dengan hihat tetapi dengan bentuk dan suara yang berbeda. Hanya terdiri dari satu cymbal tetapi berukuran besar 18" sampai 22". ukuran standar 20"

c) Crash cymbal: Berguna untuk memberi phrase/nada pada suatu lagu. Berukuran 13" sampai 22" tergantung dari selera pemain.

d) Efek cymbal: Efek cymbal terdiri atas Splash, bell, china dan swiss. Berguna untuk memberi 'warna' khusus pada suatu lagu. Splash dan bell biasanya berukuran 6" sampai 12" dan untuk china dan swiss biasanya berukuran 16" sampai 22".

Hardware terdiri atas berbagai macam bentuk dan fungsi:

1. Pedal: Berguna untuk memukul bass drum, juga tersedia double pedal, yaitu pedal yang menggunakan 2 pedal dan 2 pemukul atau beater untuk mendapatkan suara yang lebih pada bass drum.



Gambar 2.7 Pedal Drum

2. Hihat stand: Untuk menempatkan hihat cymbal yang terdiri atas 2 buah cymbal sehingga anda dapat membuka dan menutup kedua cymbal itu dengan kaki kiri anda.
3. Cymbal stand: Untuk menempatkan segala macam jenis cymbal kecuali hihat.

4. Snare stand: Untuk menempatkan Snare drum dan anda dapat merubah posisinya sesuka anda.
5. Tom holder/tom stand: Berguna untuk memasang tom-tom.

Drumhead mempunyai ukuran, jenis, fungsi dan ketebalan yang berbeda. Drumhead terdiri atas 3 bagian; Pertama Batter head, yaitu drumhead yang dirancang khusus untuk dipukul. Kedua, Resonant hanya ditaruh pada bagian bawah tom-tom dan bagian depan bass drum. Head ini tidak untuk dipukul, head ini berguna untuk memberi 'hidup' pada tom-tom dan bass drum. Dan terakhir adalah snare side, khusus hanya untuk ditaruh dibagian bawah snare untuk mendapatkan suara snare wirenya. Snare side merupakan head yang paling tipis. Ingat, tidak untuk dipukul.

2.7 Tangga Nada

2.7.1 Tangga Nada Major

Dalam teori musik, skala mayor atau tangga nada mayor adalah salah satu Tangga nada diatonik. Skala ini tersusun oleh delapan not. Interval antara not yang berurutan dalam skala mayor adalah: 1, 1, 1/2, 1, 1, 1, 1/2. Sebagai contoh, tangga nada A minor adalah C, D, E, F, G, A, B, C'. Adapun ciri-ciri tangga nada major, sebagai berikut:

1. Bersifat riang gembira
2. Bersemangat
3. Biasanya diawali dan diakhiri dengan nada Do
4. Mempunyai pola interval : 1 , 1 , ½, 1 , 1 , 1, ½

2.7.2 Tangga Nada Minor

Dalam teori musik, tangga nada minor atau tangga nada minor adalah salah satu tangga nada diatonik. Tangga nada ini tersusun oleh delapan not. Interval antara not yang berurutan dalam tangga nada minor (asli) adalah: 1, 1/2, 1, 1, 1/2, 1, 1. Sebagai contoh, tangga nada A minor adalah A, B, C, D, E, F, G, A'. Adapun ciri-ciri tangga nada major, sebagai berikut:

1. Bersifat sedih
2. Kurang Bersemangat
3. Biasanya diawali dan diakhiri dengan nada La = A
4. Mempunyai pola interval : 1 , ½ , 1 , 1 , ½ , 1 , 1

2.8 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 popular adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.

2.8.1 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.

2.8.2 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:

1. Core platform
2. OSGi
3. SWT (Standard Widget Toolkit)
4. JFace
5. 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.

Konsep Eclipse adalah IDE yang terbuka (open), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya.

2.8.3 Perkembangan Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah deployment dan maintenance untuk sistem enterprise. Adapun versi eclipse yang telah diluncurkan yaitu :

1. Eclipse 3.0
2. Eclipse 3.1
3. Callisto
4. Europa
5. Ganymede
6. Galileo
7. Helios
8. Juno

2.9 Software Development Kit (SDK)

Software Development Kit (SDK) adalah suatu kit atau library dari bahasa pemrograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya SDK terdiri dari kumpulan tools yang dibutuhkan. Misalnya bahasa pemrograman java, mempunyai SDK yang berisi suatu library yang dapat digunakan untuk membuat suatu aplikasi berbasis java.

2.10 Java Development Kit

Java Development Kit (JDK) adalah sekumpulan perangkat lunak yang dapat kamu gunakan untuk mengembangkan perangkat lunak yang berbasis Java, Sedangkan JRE adalah sebuah implementasi dari Java Virtual Machine yang benar-benar digunakan untuk menjalankan program java.Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber kompiler java, bundling, debuggers, development libraries dan lain sebagainya. Perbedaan JDK dengan SDK (Software Development Kit) yaitu JDK adalah sebuah SDK tetapi sebuah SDK tidak harus menjadi sebuah JDK.

2.11 Android Development Tools (ADT)

Android Development Tools (ADT) adalah plugin untuk Eclipse Intergrated Development Environment (IDE) yang dirancang untuk memberikan lingkungan yang terpadu di mana untuk membangun aplikasi Android. ADT memperluas kemampuan Eclipse untuk membiarkan para developer lebih cepat dalam membuat proyek baru Android, membuat aplikasi UI, menambahkan komponen berdasarkan Android Framework API, debug aplikasi dalam pengunaan Android SDK, dan membuat file APK untuk mendistribusikan aplikasi.

Mengembangkan aplikasi di Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi android, karena banyak kemudahan-kemudahan sebagai tools yang terintegrasi seperti, custom XML editor, dan debug panel ouput. Selain itu ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.