BAB II

LANDASAN TEORI

2.1 Marawis

Marawis adalah salah satu jenis "band tepuk" dengan perkusi sebagai alat musik utamanya. Musik ini merupakan kolaborasi antara kesenian Timur Tengah dan Betawi, dan memiliki unsur keagamaan yang kental. Itu tercermin dari berbagai lirik lagu yang dibawakan yang merupakan pujian dan kecintaan kepada Sang Pencipta.

2.1.1 Macam-macam alat musik marawis

1. Marawis merupakan gendang kecil berdiameter 20 Cm dengan tinggi 19 Cm. Alat ini terbuat dari kayu yang bagian tengahnya dilubangi dalat inilah yang menjadi ciri khas dari musik jenis ini, sehingga musik jenis ini pun disebut dengan Marawis.
2. Hajir disebut juga hajir marawis merupakan sebuah Gendang yang berukuran diameter 45 Cm dengan tinggi 60-70 cm, alat ini terbuat dari kayu yang bagian tengahnya dilubangi sehingga berbentuk mirip sebuah tabung. Kedua bagian ujungnya ditutup dengan kulit binatang yang berfungsi sebagai selaput / memberan. Adapun kulit binatang yang biasa digunakan adalah kulit kambing atau domba.

3. Dumbuk Pinggang merupakan alat musik sejenis gendang yang berbentuk mirip dandang, bagian tengah dan kedua ujungnya memliki diameter yang berbeda - beda, diameter terbesar pada ujung yang detutup dengan selaput/membrean dari mika, kemudian disusul bagian ujung yang terbuka, sedangkan pada bagian tengah memiliki diameter terkecil. adapun disebut dumbuk pinggang karena dalam penggunaannya alat ini diletakkan di pinggang.
4. Dumbuk Batu, bentuk alat ini mirip dengan dumbuk pinggang, hanya saja mempunyai ukuran yang sedikit lebih besar. adapun disebut dumbuk batu karena konon pada awalnya terbuat dari batu.
5. Simbal dan Tamborin, kadang kala musik marawis dilengkapi dengan tamborin atau krecek dan [Symbal] yang berdiameter kecil dimana kedua alat ini digabungkan menjadi satu kesatuan

(Sumber: http://forum.viva.co.id/musik/332286-sejarah-musik-marawis.html)

2.2 Aplikasi

Aplikasi menurut Jogiyanto *(2001:12)* adalah penggunaan dalam suatu komputer,instruksi *(instructiom)* atau pernyataan *(statement)* yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output.* Menurut Kamus *Kamus Besar Bahasa Indonesia (2001 : 52)*, "Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atauketentuan bahasa pemrograman tertentu". Aplikasi adalah suatu program komputer yang dibuat untuk

mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa software. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini. Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instructionset*.

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan

setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu

lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.3 Konsep Aplikasi *Mobile*

Mobile Application adalah perangkat lunak berbasis mobile (*PDA, handphone, smartphone, Pocket PC*).

Aplikasi *software mobile* dapat terintegrasi dengan perangkat *handphone* dan dipakai klien sebagai aplikasi mini yang dapat dibawa kemanapun untuk membantu berbagai aktifitas klien.

(http://www.e-scapemedia.co.id/mobile-application.html)

2.4 JAVA

JAVATM yang dikembangkan di Sun Microsystem berawal dari gagasan untuk menciptakan suatu bahasa, perangkat lunak yang bebas dan tidak bergantung pada platform atau sistem operasi tertentu (tidak hanya bekerja pada sistem operasi tertentu). Tujuan awalnya adalah dengan menggunakan bahasa yang sudah ada, yaitu C++ namun seiring dengan kemajuan yang dicapai, para pencipta JAVA menyadari bahwa akan lebih baik bila mereka menemukan (menciptakan) bahasa mereka sendiri daripada mengembangkan C++(sunmicrosystems).

Tidak seperti bahasa-bahasa compiler tradisional, yang mengubah kode (*source code*) menjadi perintah-perintah tingkat mesin (bahasa mesin), kompiler Java mengubah (menterjemahkan) kode-kode sumber Java menjadi perintah-perintah yang akan di- interpretasi (dibaca) oleh *runtime* Mesin *Virtual Java* (*Java Virtual Machine*).

Java dapat digunakan untuk membuat dua jenis program, yaitu applet dan aplikasi mandiri (*stand alone application*). Secara sederhana, sebuah applet adalah bagian dari halaman web entah itu berupa animasi, gambar sederhana (*image*) atau hanya sebuah garis atau sekumpulan teks.

Para pencipta Java di Sun Microsystem mendefinisikan Java sebagai bahasa yang sederhana, berorientasi object, terdistribusi, terinterpretasi, kokoh, aman, netral arsitektur, akrab, berkinerja tinggi, multi jalinan (*multithreaded*) dan dinamis (sunmicrosystems).

2.4.1 Sejarah JAVA

Java adalah bahasa pemograman tingkat tinggi yang berorientasi objek, diedarkan oleh Sun Microsystem1 pada awal tahun 1996. Sejarah awal Java berawal di tahun 1991 ketika satu group insinyur-insinyur Sun, yang dipimpin oleh *Patrick Naughton* dan *James Gosling*, ingin mendesain sebuah bahasa pemograman komputer yang berukuran kecil yang dapat digunakan untuk peralatan elektronika konsumen sepertu *switchboxes* TV kabel.

Dikarenakan peralatan-peralatan ini menggunakan konsumsi daya dan

memory yang rendah, maka bahasa pemograman tersebut harus berukuran sangat kecil. Juga karena setiap vendor menggunakan CPUs (*Central Processing Unit*) yang berbeda, maka bahasa tersebut harus bersifat multiplatfrom, tidak terikat hanya pada satu arsitektur (*Architecture Neutral*). Proyek ini diberi nama *Green Project.*

Karena harus bersifat *Architecture* Neutral, maka Green Project menggunakan *Virtual Machine* (atau dikenal dengan *Java Virtual Machine*) yang berasal dari model implementasi bahasa Pascal di awal-awal perkembangan PC. Dikarenakan insinyur insinyur Sun berlatar belakang Unix2, jadi mereka mendasari bahasa pemograman mereka dengan C++ dari pada Pascal. Secara khusus mereka membuat bahasa mereka berorientasi obyek (*object oriented*), bukan berorientasi prosedur (*procedural oriented*) seperti model bahasa Pascal. Bahasa pemograman tersebut dinamakan Oak, kemudian diubah menjadi Java (*sunmicrosystems*).

Karena pada awalnya ditujuan untuk pemograman device kecil, Java memiliki karakteristik berukuran kecil, efisien, dan *portable* untuk berbagai hardware. Perkembangannya sempat terhenti karena tidak ada yang tertarik dan tidak memiliki pasar seperti yang diramalkan. Ketika teknologi internet berkembang, Java diarahkan untuk menjadi bahasa pemograman internet karena fitur-fitur Java seperti *Architecture Neutral*, *real time*, *reliable* dan *secure* sangat sesuai untuk pengembangan internet.

2.4.2 Karakteristik JAVA

1. Sederhana

Inti dari Java terutama terletak pada kemiripannya dengan C dan C++. Karena programmer-programmer masa kini khususnya yang tertarik menggunakan Java telah memiliki pengalaman setidaknya dengan C dan mungkin dengan C++, Java tentu saja terlihat sederhana dan akrab bagi programmer- programmer ini.

Java menyederhanakan bahasa C++ dengan menambahkan fitur-fitur pendukung yang belum terdapat dalam C++ dan membuang beberapa fitur yang membuat C++ menjadi bahasa yang rumit dan sulit untuk dikuasai. Java sederhana karena hanya ia memiliki tiga tipe angka data primitif, tipe *Boolean* dan *array*. Selebihnya, semua dalam Java adalah sebuah kelas. Salah satu fitur penting yang ditawarkan Java (yang belum ada pada C++) adalah pengumpulan sampah (*garbage collection*). Dengan mekanisme ini, user tidak perlu membebaskan memori yang dialokasikan, karena semua dilakukan oleh Mesin *Virtual Java*. Java juga mendukung penulisan program multi jalinan, yaitu suatu program yang dapat melakukan lebih dari satu pekerjaan dalam waktu yang bersamaan.

2. Berorientasi Obyek

Dalam pendekatannya pada orientasi obyek, Java lebih merujuk pada SmallTalk daripada C++. Selain tipe data primitivenya, semua yang ada pada Java adalah kelas. Sebaliknya dalam C++ semuanya serba campur, dimana programmer dapat secara bebas (benar-benar bebas)dalam mencampur kode-kode orientasi obyek (kelas) (kelas) dengan kode prosedural (fungsi). Dalam Java semua ini tidak

diperbolehkan. Tidak ada fungsi global dalam Java, semua fungsi harus dipanggil melalui sebuah obyek.

Dukungan Java terhadap orientasi-obyek tidak termasuk penurunan (*inheritansi*) ganda. Para perancang Java merasa kerumitan yang dimunculkan *inheritansi* ganda tidak sebanding dengan keuntungan yang dihasilkan. Sebagai ganti dari inheritansi ganda, Java kemudian menyediakan antarmuka (*interface*).

3. Terdistribusi

Java memudahkan pembuatan aplikasi terdistribusi dengan sekumpulan kelas yang digunakan pada aplikasi-aplikasi jaringan. Dengan menggunakan kelas URL (*Uniform Resource Locator*) Java, suatu aplikasi dapat dengan mudah mengakses server-jauh (remote server).

4. Terinterpretasi

Karena Java adalah bahasa interpretasi, sekali Interpreter Java terpasang, terinstal pada suatu mesin tertentu, mesin tersebut dapat secara langsung menjalankan aplikasi-aplikasi Java (tidak peduli platform atau sistem operasi apa yang terpasang pada mesin tersebut). Ketika menggunakan bahasa interpreter, programmer juga terbebas dari kekhawatiran berkaitan dengan ketergantungan antar modul.

Keuntungan lain adalah waktu yang digunakan untuk lingkaran *edit-compile-link-test* dapat dipotong. Tanpa adanya langkah kompile dan link, maka bekerja dalam lingkungan interpreter lebih sederhana dan lebih hemat waktu karena hanya melalui lingkaran edit-test.

5. Kokoh

Membuat suatu program yang terdistribusi, mendukung multi-jalinan yang dapat dijalankan pada berbagai sistem operasi dan berbagai prosesor bukanlah suatu pekerjaan yang mudah. Berdasarkan pemikiran ini, maka Java diciptakan sebagai bahasa yang sangat ketat dalam penulisan *(strongly typed language*). Dalam Java management memori telah disederhanakan dengan dua cara. Pertama Java tidak mendukung manipulasi *pointer* atau aritmatik secara langsung, sehingga mustahil bagi program Java untuk menumpuk (meng-*overwrite*) memori atau mengkorupsi data. Kedua Java menggunakan mekanisme pengumpulan sampah saat program berjalan (*runtime*) daripada pembebasan memori secara *eksplisit*.

6. Aman

Karena Java tidak menggunakan pointer yang secara langsung merujuk pada lokasi memori seperti yang terdapat pada C atau C++, Java memiliki kendali penuh terhadap semua kode yang ada pada lingkungan Java. Sebelumnya telah diantisipasi bahwa aplikasi-aplikasi Java akan dijalankan di internet dan akan secara dinamis dapat dijalankan bersama atau mengeksekusi program lain di tempat yang berlainan melalui internet, maka para pengembang Java berpendapat tentang perlu adanya kompiler Java yang akan menghasilkan kode bite Java yang akan melewati standar keamanan *runtime* Java

7. Arsitektur Netral

Sebelum dijalankan, program Java harus dikompile terlebih dulu menggunakan kompiler Java. Proses kompilasi ini kemudian akan menghasilkan

suatu kode bite tertentu yang serupa dengan *file* kode (bahasa) mesin yang dapat dieksekusi pada mesin apapun yang memiliki interpreter Java.

8. *Portable*

Salah satu tujuan penting pembuatan Java adalah kode-kode Java haruslah *Portable*, sehingga ketika arsitektur baru (baik itu perangkat keras ataupun sistem operasi atau keduanya) berkembang, lingkungan Java dapat diterapkan dan dipindahkan pada mereka.

Pada Java, semua tipe data *primitif* (*integer, long, float, double* dan sebagainya) memiliki ukuran tertentu, tidak bergantung pada mesin atau sistem operasi dimana program Java dijalankan. Hal ini sangat berlawanan dengan bahasa seperti C atau C++ yang menyerahkan ukuran tipe data primitif pada kompiler dan mesin (serta sistem operasi). Java *portable* karena kompiler Java sendiri ditulis menggunakan Java.

9. Berkinerja Tinggi

Sebuah aplikasi Java tidak akan menyamai kinerja aplikasi bahasa terkompilasi penuh seperti C atau C++. Namun demikian untuk sebagian besar aplikasi termasuk pengolahan grafis, satu dari berbagai macam hal yang dapat ditemui secara umum pada *World Wide Web,* kinerja Java lebih dari cukup.

Banyak pengadopsi awal C++ khawatir aanya kemungkinan penurunan kinerja saat mereka mengubah program mereka dari C ke C++. Namun para pengadopsi (pengguna) awal C++ menemukan walaupun program dalam C akan melebihi suatu

program C++ dalam banyak kasus, waktu untuk pengembangan lebih lanjut dan biaya tidak sesuai dengan kinerja yang dihasilkan.

Walupun aplikasi Java mungkin tidak dapat menyamai aplikasi C++, namun secara normal aplikasi Java akan cukup cepat, dan Java memungkinkan user melakukan banyak hal yang tidak dapat dilakukan dengan C++.

10. Multi-Jalinan

Menulis suatu bahasa yang hanya bisa melakukan satu perkerjaan adalah hal yang biasa dalam bahasa pemrograman. Program-program Java dapat terdiri atas beberapa jalinan yang memungkinkan program untuk melakukan beberapa pekerjaan sekaligus. Suatu contoh, sebuah program multi jalinan dapat me-render sebuah image di layar pada satu jalinan disamping menerima masukan dari keyboard user pada jalinan utama-nya.

Dukungan terhadap beberapa jalinan tersinkronisasi sekaligus telah terbangun (secara internal) pada Java dan lingkungan *runtime*-nya. Jalinan tersinkronisasi sangat berguna dalam membuat aplikasi yang terdistribusi dan berbasis jaringan. Aplikasi semacam ini dapat berkomunikasi dengan server jauh pada satu jalinan dan berinteraksi dengan user pada jalinan yang lain.

11. Dinamis

Karena terinterpretasi, Java adalah bahasa yang benar-benar dinamis. Saat *runtime*, lingkungan Java dapat mengembangkan dirinya dengan terhubung pada kelas-kelas yang mungkin terletak pada server-jauh pada suatu jaringan (misalnya, lewat internet). Pada C++ setiap kali anggota variabel atau fungsi ditambahkan pada

suatu kelas, maka kelas tersebut dan semua kode tambahan yang merujuk pada kelas tersebut perlu dikompile ulang. Java menyederhanakan masalah ini dengan menyerahkan pada *runtime*. Saat *runtime* interpreter Java melakukan resolusi nama ketika terhubung dengan kelas-kelas yang bersangkutan. Interpreter Java juga bertanggung-jawab dalam menentukan penempatan obyek dalam memori. Dua fitur pada interpreter Java ini memecahkan masalah berkaitan degan perubahan definisi kelas ketika digunakan kelas-kelas yang lain.

(Sumber : Ivan Michael Siregar. 2011. *Membongkar source code berbagai Aplikasi* Android.Gava Media.Yogyakarta.)

2.5 Android

2.5.1 Pengertian Android

Android adalah kumpulan perangkat lunak yang ditujukan bagi perangkat bergerak mencakup sistem operasi, *middleware*, dan aplikasi kunci. Android *Standart Development Kid* (SDK) menyediakan perlengkapan dan *Application Programming Interface* (API) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Jeff Lessard (2010; 1-2) *Android is an Operation System (OS) developed by Open Handset Alliance (OHA).*android adalah sistem operasi yang di kembangkan oleh *Open Hadset Alliance*. Android dikembangkan oleh Google bersama *Open Handset Allience* (OHA) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan *Hardware, Software* dan

perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

(sumber : http://id.wikipedia.org/wiki/Android_(sistem_operasi)/27-juni-2012)

2.5.2 Sejarah perkembangan Android

Pada mulanya terdapat berbagai macam sistem operasi pada perangkat selular, diantaranya sistem operasi Symbian, Microsoft Windsos Mobile, Mobile Linux, iPhone, dan sistem operasi lainnya. Namun diantara sistem operasi yang ada belum mendukung standar dan penerbitan API yang dapat dimanfaatkan secara keseluruhan dan dengan biaya yang murah. Kemudian Google ikut berkecimpung didalamnya dengan platform Android, yang menjanjikan keterbukaan, keterjangkauan, *open source*, dan *framework* berkualitas.

Pada tahun 2005, Google mengakuisisi perusahaan Android Inc. untuk memulai pengembangan platform Android. Dimana terlibat dalam pengembangan ini Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada pertengahan 2007 sekelompok pemimpin industri bersama-sama membentuk aliansi perangkat selular terbuka, *Open Handset Alliance* (OHA). Bagian dari tujuan aliansi ini adalah berinovasi dengan cepat dan menanggapi kebutuhan konsumen dengan lebih baik, dengan produk awalnya adalah platform Android. Dimana Android dirancang untuk melayani kebutuhan operator telekomunikasi, manufaktur *handset*, dan pengembang aplikasi. OHA berkomitmen untuk membuat Android *open source* dengan lisensi Apache versi 2.0. Android pertama kali diluncurkan pada 5 November 2007, dan *smartphone*

pertama yang menggunakan sistem operasi Android dikeluarkan oleh T-Mobile dengan sebutan G1 pada bulan September 2008. Hingga saat ini Android telah merilis beberapa versi Android untuk menyempurnakan versi sebelumnya. Selain berdasarkan penomoran, pada setiap versi Android terdapat kode nama berdasarkan nama-nama kue. Hingga saat ini sudah terdapat beberapa versi yang telah diluncurkan, diantaranya: versi 1.5 dirilis pada 30 April 2009 diberi nama *Cupcake*, versi 1.6 dirilis pada 15 September 2009 diberi nama *Donut*, versi 2.0 dirilis pada 26 Oktober 2009 diberi nama *Éclair*, versi 2.2 diberinama *froyo*, 2.3 diberi nama *gingerbread* dan versi 4 diberinama *ice cream sandwich* .

2.5.3 Kelebihan Android

Sudah banyak *platform* untuk perangkat selular saat ini, termasuk didalamnya Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM0), 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.

2.5.4 Karakteristik Android

Andorid merupakan subset perangkat lunak untuk perangkat mobile yang meliputi sistem operasi, middleware, dan aplikasi init yuang di-release oleh Google. Pada tulisan sebelumnya, kita mengenal SDK (Software Development Kit). SDK adalah suatu tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform atau linkungan Android. Pengembangan aplikasi Android menggunaka bahasa pemrograman Java. Seperti kita ketahui, SDK ini dikembangkan oleh OHA (Open Handsate Alliance). Organisasi OHA ini terdiri atas Googl, Intel, Motorola, Qualcomm, T-Mobile, dan NVIDIA. SDK dapat kita gunakan pada beberapa IDE (Integrated Development and Environment) – software untuk membuat suatu program – akan tetapi, pada tulisan selanjutnya, saya akan menggunakan Eclipse karena OHA secara resmi membuat plugin untuk IDE Eclipse.

2.5.5 Perkembangan Android

Android telah banyak melakukan perkembangan, berikut adalah perkembangan dari 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, dukunganBluetooth 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, 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.

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 3.0/3.1 (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. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

8. Android versi 4.0 (ICS :Ice Cream Sandwich)

   Diumumkan pada tanggal 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.

(Sumber : http://developer.android.com/guide/index.html / *Android SDK Developer Guide*,/24 Februari 2012.)

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 *plugin.*

(Sumber *:* http://bayduaenam.blogspot.com/2011/06/eclipse.html *Masih Belajar Eclipse* 20 Maret 2012)

2.6.1 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.

2.6.2 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.

2.6.3 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

(Sumber :http://bayduaenam.blogspot.com/2011/06/eclipse.html *Masih Belajar Eclipse* 20 Maret 2012)

2.7 *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.

(Sumber : http://developer.android.com/guide/index.html / *Android SDK Developer Guide*,/24 Februari 2012.)

2.8 *Java Development Kit* (*JDK*)

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.

(Sumber **:** http://ikc.dinus.ac.id/*Menginstall Java Development Kit*/ 26 April 2012)

2.9 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.

(Sumber : http://opensource.telkomspeedy.com/wiki/index.php/Eclipse:*_ADT plugin untuk Eclipse* 26 April 2012).