BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi *Mobile*

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.

Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalkan telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi.

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 Pengertian Peta

Pengertian peta menurut F.J Mounkhous dan H.R Wilkinson: Peta ialah suatu perakitan terpadu atau suatu sintesa dari empat kelompok infomasi yaitu titik, garis, wilayah dan nama yang dikemukakan dalam istilah : liputan, ciri, pola, bentuk, ukuran, ketebalan simbul dan lain-lain. Batasan tersebut di atas langsung menunjuk ke pada segi teknik penetapan simbul dan analisis keruangan aspek persebaran data dalam jenis dan besaran serta penamaan geografiknya (*toponimy*).

2.3 Pengertian Kereta Api

Kereta api adalah sarana transportasi berupa kendaraan dengan tenaga gerak, baik berjalan sendiri maupun dirangkaikan dengan kendaraan lainnya, yang akan ataupun sedang bergerak di rel. Kereta api merupakan alat transportasi massal yang umumnya terdiri dari lokomotif (kendaraan dengan tenaga gerak yang berjalan sendiri) dan rangkaian kereta atau gerbong (dirangkaikan dengan kendaraan lainnya). Rangkaian kereta atau gerbong tersebut berukuran relatif luas sehingga mampu memuat penumpang maupun barang dalam skala besar. Karena sifatnya sebagai angkutan massal efektif, beberapa negara berusaha memanfaatkannya secara maksimal sebagai alat transportasi utama angkutan darat baik di dalam kota, antarkota, maupun antarnegara

2.4 Pengertian Informasi

Menurut Fatta (2007:9), informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.

Menurut Rochim (2002:3), informasi menyandang arti manfaat, bila kita bisa memanfaatkanya, informasi mengandung makna usaha, untuk mendapatkannya, memahaminya, menggunakannya, menyebarkannya, menyimpannya dan memadukannya dengan informasi lain menjadi suatu bentuk informasi baru.

Menurut Kusrini dan Koniyo (2007:7), informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengembalian keputusan saat ini atau mendukung sumber informasi.

Menurut Wahyono (2004:3), informasi adalah data yang telah diolah menjadi bentuk yang berguna bagi penerimanya dan nyata, berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan.

Berdasarkan definisi diatas dapat disimpulkan bahwa informasi adalah data yang telah diolah menjadi sesuatu yang bermanfaat bagi penggunanya itu sendiri.

2.4.1 Ciri-Ciri Informasi

Ciri-ciri informasi menurut Azhar Susanto (2008 : 38) menyatakan bahwa:

1. Akurat

   Informasi harus mencerminkan keadaan yang sebenarnya

2. Tepat waktu

   Informasi harus tersedia atau ada pada saat informasi tersebut diperlukan

3. Relevan

   Informasi yang diberikan harus sesuai dengan yang dibutuhkan oleh suatu individu

4. Lengkap

Informasi harus diberikan secara lengkap

2.4.2 Fungsi Informasi

Selain itu informasi mempunya berbagai fungsi, antara lain sebagai berikut:

1. Menambah pengetahuan.

   Adanya informasi yang akan menambah pengetahuan bagi penerimanya yang dapat digunakan sebagai baha pertimbangan yang mendukung proses pengambilan keputusan.

2. Mengurangi ketidakpastian.

   Adanya informasi yang akan mengurangi ketidakpastian karena apa yang akan terjadi dapat diketahui sebelumnya, sehingga menghindari keraguan pada saat pengambilan keputusan

3. Mengurangi resiko kegagalan.

   Adanya informasi akan mengurangi resiko kegagalan karena apa yang akan terjadi dapat diantisipasi dengan baik, sehingga kemungkinan terjadinya kegagalan akan dapat dikurangi dengan pengambilan keputusan yang tepat.

4. Mengurangi keanekaragaman/variasi yang tidak diperlukan.

   Adanya informasi akan mengurangi keanekaragaman yang tidak diperlukan, karena keputusan yang di ambil lebih terarah.

5. Memberi standar, aturan-aturan, ukuran dan keputusan keputusan yang menentukan pencapaian sasaran dan tujuan. Adanya informasi

yang akan memberikan standar, aturan, ukuran, dan keputusan yang lebih terarah untuk mencapai sasara dan tujuan yang telah ditetapkan secara baik berdasarkan informasi yang diperoleh.

2.5 Java

Java menurut definisi dari *sun* adalah nama untuk sekumpulan tekhnologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java adalah generasi kedua dari java *platform* (generasi awalnya adalah *Java Development Kit*).

2.5.1 Tentang Java

Java berdiri di atas sebuah mesin interpreter yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu , bahasa Java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Platform Java terdiri dari sekumpulan *library*, JVM, kelas-kelas *loader* yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah kompiler, debuger dan kakas lain yang dipaket dalam *Java Development Kit* (JDK). Java adalah generasi yang sekarang sedang berkembang dari platform Java. Agar sebuah program Java dapat dijalankan, maka file dengan ekstensi .java harus dikompilasi menjadi file *bytecode*. Untuk menjalankan *bytecode* tersebut dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi.

Platform Java memiliki tiga buah edisi yang berbeda, yaitu J2EE (*Java Enterprise Edition*), J2ME (*Java2 Micro Edition*), dan J2SE (*Java2 Second Edition*).

2.5.1.1 J2EE (Java Enterprise Edition)

J2EE adalah kelompok dari beberapa API dari Java dan tekhnologi selain Java. J2EE dibuat untuk membuat aplikasi yang kompleks. J2EE sering dianggap sebagai *middleware* atau teknologi yang berjalan di *server*, namun sebenarnya j2EE tidak hanya terbatas untuk itu.

2.5.1.2 J2SE (Java2 Second Edition)

J2SE adalah inti dari bahasa pemrograman Java. JDK (*Java Development Kit*) adalah salah satu tool dari J2SE untuk mengkompilasi dan menjalankan program Java. Di dalamnya terdapat tool untuk mengkompilasi program Java dan JRE. Tool J2SE yang salah satunya adalah jdk1.5, jdk merupakan tool *open source* dari Sun.

2.5.1.3 J2ME (Java2 Micro Edition)

Java Micro Edition atau yang disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer desktop yang biasanya lebih kecil dibandingkan perangkat komputer desktop. J2ME biasa digunakan pada telepon selular, pager, personal digital assistants (PDA’s) dan sejenisnya.

Teknologi J2ME juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada ponsel. J2ME sangat tergantung pada perangkat (*device*) yang digunakan, bisa dari segi merk ponsel, maupun kemampuan ponsel, dan dukungan terhadap teknologi J2ME. Misalnya, jika sebuah ponsel tidak memiliki kamera maka jelas J2ME pada ponsel tersebut tidak bisa mengakses kamera. Keterbatasan lainnya adalah padaukuran aplikasi, karena memori pada ponsel sangat terbatas. Sebagian besar ponsel tidak mengijinkan aplikasi J2ME menulis pada file karena alasan keamanan.

2.6 Android

Android merupakan suatu software stack untuk *mobile device.* Di dalamnya terdapat sistem operasi, *middleware,* dan key application. Aplikasi pada platform ini dikembangkan dalam bahasa pemrograman java. Android sendiri memiliki banyak fitur diantaranya adalah:

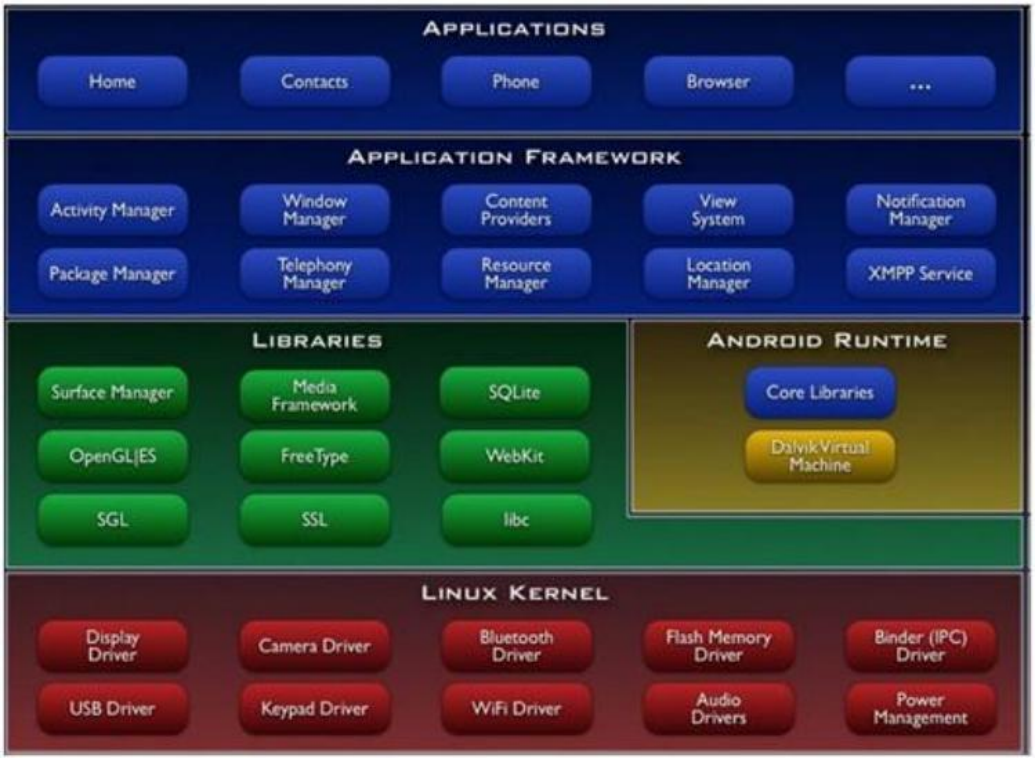
Merupakan sebuah *Application Framework* sehingga programmer dapat menggunakan beberapa fungsi yang telah disediakan.

1. *Dalvik virtual machine*. Tiap aplikasi dalam Android memiliki *instance virtual machine* yang dapat bekerja secara efisien dalam lingkungan memoriyang terbatas.
2. *Integrated browser. Web browser* berbasis *WebKit engine* terdapat pada *browser default Android* atau pun dapat diintegrasikan dengan aplikasi lain.
3. *Optimized graphics*. *Library* grafis 2D yang kaya dan 3D berbasis *OpenGL* ES 1.0 yang mendukung akselerasi *hardware*.

4. *SQLite*. Basis data relasional yang ringan namun sangat *powerful*.
5. Media Support. Mendukung berbagai format audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
6. *GSM Telephony*. Mendukung fungsi komunikasi GSM.
7. *Bluetooth,EDGE,3G* dan WiFi. Mendukung komunikasi pada jaringan (tergantung hardware).
8. Kamera, GPS, kompas dan accelerometer. Mendukung berbagai fitur yang disediakan oleh hardware.
9. Kakas pengembangan yang lengkap. Termasuk *device emulator*, tools untuk debugging, profiling memori dan performa, plugin untuk *Eclipse IDE*.

2.6.1 Arsitektur Android

Android terdiri dari beberapa stack software yang terdiri dari: *Applications, Application Framework, Libraries, Android Runtime* dan *Kernel Linux*. Arsitektur lengkap platform ini dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur Android

2.6.2 *Aplication*

Application merupakan program yang langsung berhubungan dengan user. Baik program yang merupakan bawaan dari *Android* sendiri maupun program yang dibuat oleh developer menggunakan bahasa pemrograman java. Contoh program bawaan dari platform Android sendiri adalah *email client*, program SMS, calendar, maps, *web browser, contact* dan sebagainya.

2.6.3 *Aplication Framework*

Lapisan ini berisi sekumpulan API yang dapat digunakan oleh programmer maupun *core application* dari *Android*. Lapisan ini dirancang untuk memudahkan penggunaan komponen dari Android sendiri. Aplikasi manapun

dalam *Android* dapat berbagi fungsi sehingga aplikasi lain dapat memanfaatkannya. Aplikasi pada *Android* disusun atas beberapa komponen:

1. Sekumpulan Views. Digunakan untuk mengatur tampilan pada aplikasi. Contohnya adalah lists, grids, text box, button, bahkan embeddable webbrowser.
2. *Content providers.* Komponen yang mengatur agar aplikasi dapat mengakses resources dari aplikasi lain (seperti Contacts), atau berbagi data dengan aplikasi lain.
3. *Resource Manager*. Menyediakan akses ke pada resource non-code seperti localized string, grafik dan *file layout.*
4. *Notification Manager.* Memungkinkan agar suatu aplikasi dapat menampilkan peringatan yang dapat di kostumasi pada status bar.
5. *Manager.* Mengatur siklus aplikasi dan navigasi antar aplikasi yang sedang berjalan.

2.6.4 *Libraries*

Android mendukung beberapa library C/C++ yang digunakan pada berbagai komponen *Android.* Kemampuan ini dapat diakses oleh developer melalui *Android application framework*. Beberapa library diantaranya adalah:

1. *System C library*. Implementasi library C standar (libc).
2. Media Libraries. Mendukung berbagai format multimedia (termasuk MPEG4,H.264, MP3, AAC, AMR, JPG, PNG).
3. *Surface Manager*. Mengatur akses ke subsistem display.
4. LibWebCore. *Engine web browser* modern.
5. SGL. Engine grafis 2D.

6. 3D Library. Implementasi OpenGL ES 1.0 yang mendukung akselerasi *hardware.*
7. *FreeType*. Rendering untuk bitmap dan vector font.
8. *SQLite.* Basis data relasional yang kecil namun sangat ampuh.

2.6.5 Android *Runtime*

Tiap aplikasi pada Android memiliki proses-nya masing-masing. Tiap aplikasi tersebut memiliki instans dari *Dalvik virtual machine* (VM) *Dalvik virtual machine* dirancang agar suatu device dapat menjalankan beberapa VM secara efisien. Dalvik VM mengeksekusi file dengan format *Dalvik Executable format* (.dex) yang dirancang untuk meminimalkan memory footprint . Dalvik VM berbasis register, dan dapat menjalankan kelas-kelas yang dikompilasi dengan bahasa pemrograman java dan ditransformasikan menjadi format .dex. Dalvik VM sendiri bergantung pada Kernel Linux untuk fungsi dasarnya, seperti threading dan manajemen memori secara low-level.

2.6.6 Anatomi Aplikasi Android

Terdapat empat building blocks pada aplikasi Android yaitu:

1. Activity
2. Intent Receiver
3. Service
4. Content Provider

Tidak semua aplikasi membutuhkan keempat blok ini, tetapi suatu aplikasi dibuat menggunakan kombinasi beberapa blok ini. Setelah memutuskan

blok komponen yang akan digunakan, blok tersebut didaftarkan pada suatu file yang disebut dengan *AndroidManifest.xml*. File XML ini digunakan untuk menyatakan komponen apa saja yang dibutuhkan oleh sebuah aplikasi dan kemampuan serta kebutuhan aplikasi tersebut.

2.6.6.1 *Activity*

Activity paling umum diantara keempat *building block*. *Activity* biasanya suatu tampilan (screen) dalam suatu aplikasi. Tiap *activity* diimplementasikan sebagai satu kelas yang meng-extend base class *Activity*. Kelas yang dibuat akan menampilkan user interface yang terdiri dari beberapa views dan akan merespon suatu event. Kebanyakan, aplikasi terdiri dari beberapa screen. Contohnya aplikasi pengiriman pesan akan memiliki screen untuk mendaftarkan pesan yang masuk, menulis pesan dan screen lainnya untuk melakukan pengaturan. Tiap screen pada contoh ini akan diimplementasi sebagai suatu *activity*. Aplikasi dapat berpindah ke screen lain dengan cara memulai suatu *activity* baru. Pada suatu kasus tertentu, *activity* akan mengembalikan nilai kepada activity sebelumnya. Contohnya *activity* untuk memilih foto akan mengembalikan foto yang dipilih pada *activity* yang memanggilnya. Pada saat *screen* baru terbuka, screen sebelumnya akan dihentikan dan state nya akan disimpan pada history *stack*. *User* dapat melakukan navigasi ke activity sebelumnya melalui *history* tersebut. *Scree*n akan di hapus dari *history stack* apabila sistem memutuskan bahwa screen tersebut tidak layak untuk berada pada memori. *Androi*d menyimpan history stack untuk setiap aplikasi yang dijalankan dari tampilan awal (*home screen*).

2.6.6.2 *Intent* dan *Intent Filter*

Android menggunakan kelas khusus bernama intent untuk berpindah dari screen ke screen lainnya. Sebuah intent mendeskripsikan apa yang ingin aplikasi penuhi. Bagian struktur data dari intent yang paling penting adalah action dan data dari aksi yang dilakukan. Nilai action yang biasanya adalah MAIN, VIEW, PICK, EDIT dan sebagainya. Data diekspresikan sebagai URI. Sebagai contoh, untuk melihat informasi suatu contact seseorang, kita membuat intent dengan action VIEW dan data sebagai URI yang merepresentasikan orang tersebut. Terdapat suatu kelas yang dinamakan *IntentFilter*. Apabila intent adalah suatu permintaan untuk melakukan sesuatu, *IntentFilter* adalah deskripsi yang dapat dilakukan oleh intent activity. Activity yang dapat menampilkan informasi contact dari seseorang akan mempublikasikan *IntentFilter* yang menyatakan bahwa ia dapat menangani action VIEW apabila dipakai pada data yang merepresentasikan seseorang. Suatu aplikasi mempublikasikan IntentFilter pada file *AndroidManifest.xml.* Navigasi dari suatu screen ke screen lainnya dipenuhi dengan cara me-resolve intent. Untuk melakukan navigasi ke depan, activity memanggil *startActivity(myIntent).* Sistem akan mencari IntentFilter untuk semua aplikasi dan memilih activity yang IntentFilter-nya yang paling mendekati dengan myIntent. activity baru akan mendapat informasi tentang intent yang menyebabkan activity tersebut dijalankan. Proses pencarian intent terjadi sewaktu runtime pada saat *startActivity* dipanggil.

2.6.6.3 *Intent Recivier*

Intent Receiver dapat digunakan pada saat kita menginginkan kode kita dijalankan akibat reaksi dari eksternal event tertentu. Contohnya, pada saat telepon berbunyi, atau pada saat ada data pada jaringan, atau pada suatu waktu tertentu. *IntentReceiver* tidak menampilkan user interface, meskipun ia dapat menggunakan *Notification Manager* untuk memperingatkan user apabila sesuatu terjadi. *IntentReceiver* didaftarkan pada file *AndroidManifest.xml,* tetapi secara manual juga bisa ditambahkan dengan fungsi *Context.registerReceiver()*.

2.6.6.4 *Service*

Service adalah sebuah kode yang hidup terus-menerus dan berjalan tanpa memiliki tampilan. Contoh yang sederhana adalah suatu aplikasi media player yang sedang memainkan lagu dari *play list*. Pada aplikasi media player, kemungkinan terdapat satu atau lebih *activity* yang memperbolehkan user memilih lagu dan memainkannya. Tetapi musik yang sedang diputar tidak boleh ditangani oleh *activity* karena user akan menginginkan musik tetap diputar meskipun ia berpindah ke screen baru. Pada kasus ini activity *media player* dapat menjalankan *service* menggunakan fungsi *Context.startService()* untuk menjalankan *service* di *background* yang memutar musik. Sistem akan tetap memutar musik sampai selesai. Selain itu activity dapat terhubung dengan service menggunakan fungsi *Context.bindService()*. Pada saat terhubung, aplikasi dapat berkomunikasi melalui *interface* yang diberikan oleh *service* yang bersangkutan. Pada contoh *media player*, *interface service* akan memberikan *fungsi pause*, *rewind* dan sebagainya.

2.6.6.5 *Provider*

Content provider digunakan untuk pertukaran data antara aplikasi. Data yang dipertukarkan bersifat shared. *Content provider* adalah sebuah kelas yang berisi method standar yang memungkinkan aplikasi lain dapat menyimpan dan mengambil suatu data yang ditangani oleh *content provider*.

2.6.7 Perkembangan Operasi Sistem Android

Android mengalami perkembangan yang baik berikut adalah tahapan perkembangan *android* :

2.6.7.1 Android Beta

Pertama kali dirilis pada 5 November 2007, kemudian pada 12 November 2007 Software Development Kit (SDK) dirilis oleh Google.

2.6.7.2 Android 1.0 (Astro)

Pertama kali dirilis pada 23 Spetember 2008. Sebenarnya *Android* versi pertama ini akan dinamai dengan nama "Astro" tapi karena alasan hak cipta dan trademark nama"Astro" tidak jadi disematkan pada versi pertama dari *OS Android* ini. HTC Dream adalah ponsel pertama yang menggunakan OS ini. Berikut penampakan HTC Dream.

Pada android versi pertama ini sudah mulai diperkenalkan *Android* Market, tempat untuk mendownload berbagai macam aplikasi dan update untuk Android. Kemudian terdapat fitur *syncronisasi* dengan berbagai layanan *Google* lainnya seperti Google Contact, Calender, Maps, Talk, dan lainnya.

2.6.7.3 Android 1.1 (Bender)

Pertama kali dirilis pada 9 Februari 2009. Versi Android kedua ini juga mengalami masalah penamaan yang sama dengan versi pertamanya. Pada awalnya Android ini akan diberi nama "Bender" akan tetapi karena alasan melanggar trademark, nama "Bender" tidak jadi disematkan pada versi Android ini. Awalnya versi OS Android ini dirilis untuk perangkat T-Mobile G1 saja. Versi ini merupakan *update* untuk memperbaiki beberapa *bugs,* mengganti API dan menambahkan beberapa fitur. Berikut penampakan *home screen OS Android* v.1.1.

2.6.7.4 Android 1.5 Cupcake

Pertama kali dirilis pada 30 April 2009. Nah, mulai versi Android ini penamaan menggunakan nama makan pencuci mulut (dessert) mulai digunakan, karena ini merupakan versi yang ketiga maka penamaan diawali dengan huruf "C" dan jadilah "Cupcake" menjadi nama resmi dari versi OS Android ketiga ini. OS ini berbasiskan pada kernel Linux 2.6.27 dan menambahkan beberapa update serta UI baru dari versi Android sebelumnya. Mulai terdapat "widget" yang dapat dibesar kecilkan. Kemudian ditambah kemampuan untuk meng-upload video dan gambar ke Youtube dan Picasa.

2.6.7.5 Android 1.6 Donut

Dirilis pertama kali pada 15 September 2009. Terdapat peningkatan pada fitur pencarian dan UI yang lebih user friendly. Pada versi ini juga sudah

mendukung teknologi CDMA/EVDO, 802.1x, VPNs. Kemudian support layar dengan resolusi WVGA. Berikut penampakan Android v1.6 Donut.

2.6.7.6 Android 2.0/2.1 Enclair

Dirilis pertama kali pada 9 Desember 2009. Terjadi penambahan fitur untuk 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. Beberapa versi updatenya antara Android v.2.0 kemudian v2.0.2 dan terakhir v.2.1.

2.6.7.7 Android 2.2 Frozen Yoghurt (Froyo)

Dirilis pertamakali pada 20 Mei 2010 pada smartphone Google Nexus One. Pada versi ini sudah support terhadap Adobe Flash Player 10.1. Peningkatan pada kecepatan membuka dan menutup aplikasi, serta penggunaan SD Card sebagai tempat penyimpanan aplikasi. Ketika Android Froyo hadir mulai muncul banyak diskusi yang membahas mengenai persaingan antara Android dengan iOS yang akan semakin ketat di masa yang akan datang. Beberapa versi *update* yang dirilis antara lain Android v.2.2.1 hingga v.2.2.3.

2.6.7.8 Android 2.3 Gingerbread

Pertama kali diperkenalkan pada 6 Desember 2010. Terjadi banyak peningkatan pada versi Android yang satu ini dibandingkan dengan versi sebelumnya. Dirancang untuk memaksimalakan kemampuan aplikasi dan game. Serta mulai digunakannya Near Field Communication (NFC). Perbaikan terhadap

dukungan layar resolusi WXGA dan diatasnya. Beberapa versi update yang dirilis antara lain v.2.3.3 hingga v.2.3.7. Sampai saat ini Android Gingerbread merupakan versi Android yang memiliki pengguna terbanyak dibandingkan dengan seri Android lainnya, yaitu mencapai 65% dari seluruh versi Android yang dirilis.

2.6.7.9 Android 3.0/3.1 Honeycomb

Pertama kali diperkenalkan pada 22 Februari 2011 dan Motorola Xoom adalah yang pertama kali menggunakannya. Android versi ini merupakan OS yang didesain khusus untuk pengoptimalan pengunaan pada tablet PC.

2.6.7.10 Android 4.0 Ice Cream Sandwich

Sampai tulisan ini ditulis ICS merupakan versi Android yang paling anyar. Pertama kali dirilis pada 19 Oktober 2011. Smartphone yang pertama kali mengunakan OS Android ini adalah Samsung Galaxy Nexus. Gingerbread, dapat di-update ke Android Ice Cream Sandwich. Namun sayangnnya sampai saat ini kebanyak smartphone yang menggunakan Android ICS merupakan smartphone kelas high-end yang dijual dengan harga cukup mahal. Mungkin karena alasan inilah distribusi versi Android satu ini tidak lebih dari 8% sampai pertehangahan tahun 2012 ini.

2.6.7.11 Android 4.1 Jelly Beans

Google secara resmi mengumumkan sistem operasi mobile Android versi 4.1, yang diberi nama Jelly Bean, dalam konferensi Google I/O 2012 di San Fransisco, Amerika Serikat, Rabu (27/6/2012). Jelly Bean diumumkan dua pekan

setelah Apple mengumumkan sistem operasi mobile iOS versi 6. Jelly Bean mengalami banyak pembaruan dari Android sebelumnya, *versi 4.0 Ice Cream Sandwich.*

2.7 Perangkat Lunak yang Digunakan

Untuk memulai pemrograman mobile, memerlukan suatu tool pengembangan khusus. Salah satu tool yang paling populer untuk pemrograman berbasis android adalah *eclips* Indigo IDE 3.7.1.

2.7.1 Eclipse

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

1. *Multi-platform:* Target sistem operasi Eclipse adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX* dan *Mac OS X*.
2. *Mulit-language*: *Eclipse* dikembangkan dengan bahasa pemrograman *Java,* akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++, Cobol, Python, Perl, PHP,* dan lain sebagainya.
3. Multi-role: Selain sebagai *IDE* untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu *IDE* favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari *Eclipse* yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.

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. Salah satu situs yang menawarkan plug-in secara gratis seperti *Eclipse downloads by project*.