# BAB II

# LANDASAN TEORI

Pada bab ini akan dijelaskan beberapa teori yang berkaitan dengan permasalahan yang dibahas sebagai dasar pemikiran.

## 2.1 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*. Program aplikasi merupakan program siap pakai. Program yang direka untuk melaksanakan

suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi ialah program pemproses kata dan *Web Browser*. Aplikasi akan menggunakan system operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app. Secara *historis*, aplikasi adalah *software* yang dikembangkan oleh sebuah perusahaan. App adalah *sofware* yang dibeliperusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah ini untuk merefleksikan medan pertempuran persaingan yang baru, yang *paralel* dengan yang terjadi antar sistem operasi yang dimunculkan.

### 2.1.1 Klasifikasi Aplikasi

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

1. Perangkat lunak perusahaan (*enterprise*)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

## 2.2 Sarana dan Prasarana

Sarana prasarana secara umum banyak diartikan menurut beberapa sumber. Sarana adalah perlengkapan yang dapat dipindah-pindahkan untuk mendukung fungsi kegiatan dan satuan pendidikan, yang meliputi : peralatan, perabotan, media pendidikan dan buku (Internet menurut Asep). Sarana adalah segala sesuatu yang dipakai sebagai alat dalam mencapai makana dan tujuan. Prasarana adalah segala sesuatu yang merupakan penunjang utama terselenggaranya suatu proses (Kamus Besar Bahasa Indonesia). Sarana prasarana adalah alat secara fisik untuk menyampaikan isi pembelajaran (Sagne dan Brigs dalam Latuheru, 1988:13). Dari berbagai definisi menurut para ahli dapat diartikan bahwa sarana prasarana adalah sumber daya pendukung yang terdiri dari segala bentuk jenis bangunan/tanpa bangunan beserta dengan perlengkapannya dan memenuhi persyaratan untuk pelaksanaan kegiatan.

### 2.2.1 Sarana Transportasi

Transportasi merupakan sarana pelayanan dalam pergerakan manusia dan barang dari suatu tempat ke tempat lainnya, serta dapat memberikan kemudahan untuk mencapai tempat yang dituju. Seperti yang dikemukakan oleh H.P White dan M.L Senior (1985 : 1) dalam Wulansari (2007), sebagai berikut :

*"…….transportasi is used as means to end: the movement of people and good from where they are to another place where, for the time being at any rete their satisfaction or value will be encanced. Transport creates utilities of place".*

Dalam geografi, studi transportasi menyangkut aspek perubahan ruang, sebagai akibat adanya interaksi antara aspek manusia dalam menggunakan ruang itu sendiri. Jadi, transportasi merupakan kunci dari berfungsinya suatu tempat, karena transportasi dalam kehidupan manusia memegang peranan yang sangat penting. Dengan demikian studi geografi mengenai aspek transportasi ini merupakan studi gejala, geografi yang dinamis karena mengungkapkan interaksi keruangan dan 24 mempelajari penyebaran gejala dan fakta tersebut dalam ruang. Dalam hal ini penyebaran kegiatan penduduk, baik antara pusat kegiatan maupun antara pusat dengan daerah lainnya. Transportasi sebagai dasar untuk pembangunan ekonomi dan perkembangan masyarakat serta pertumbuhan industrialisasi. Dalam perencanaan wilayah maupun perencanaan kota, masalah transportasi tidak dapat diabaikan.

Menurut Daldjoeni (1986 : 109) pentingnya transportasi, tercermin pada semakin meningkatnya kebutuhan akan jasa angkutan bagi mobilitas orang serta barang dari dan ke seluruh pelosok tanah air, bahkan dari dan keluar negeri. Di samping itu, transportasi juga berperan sebagai penunjang, pendorong, dan penggerak bagi pertumbuhan daerah yang berpotensi namun belum berkembang, dalam upaya peningkatan dan pemerataan pembangunan serta hasil-hasilnya. Pengelolaan dari sumber-sumber alam dan industry dari suatu daerah serta pengangkutan hasil-hasil produksinya, pertumbuhan penduduk serta mobilitasnya, kesemuanya itu mengakibatkan gerak. Hasil-hasil produksi dari daerah yang terkecil atau mempunyai spesialisasi dalam produksinya, sehingga terjadi suatu gerak pengangkutan. Demikian pula manusia selalu akan bergerak ke tempat

lainnya baik untuk bekerja maupun memenuhi kebutuhan yang tidak dapat di tempat asalnya, sehingga terjadi pula suatu gerak pengangkutan.

Menurut Morlok diterjemahkan oleh Hainim (1985 : 6) : "Suatu *system* transportasi perkotaan apakah itu berupa sarana jalan raya baru, sarana transit yang baik, menentukan cara terbaik dalam mengatur lalu lintas melalui *system* 25 tersebut, ataupun hanya sekedar mengatur rute-rute transit untuk melayani kebutuhan perjalanan saat ini dengan lebih baik".

Alexander dan Gibson (1980) dalam Kurniawati (2005), mengemukakan bahwa transportasi adalah komponen penting dalam geografi yang merupakan variabel dari suatu wilayah yang dapat diketahui karakteristik serta analisis keterhubungannya, seperti hubungan lokasi suatu tempat, arus lalu lintas dan fenomena-fenomena keterhubungann lainnya. Hal ini menunjukkan bahwa transportasi merupakan bagian integral dari suatu fungsi masyarakat dan berhubungan erat dengan gaya hidup, jangkauan dan lokasi kegiatan yang produktif dan tingkat pelayanan serta pendistribusian. Transportasi dapat dikatakan kegiatan yang kompleks dari berbagai aspek yang mempengaruhinya. Keadaan fisik dan sosial wilayah dengan komponen transportasi akan membentuk suatu sistem. Sistem transportasi akan berpengaruh pada struktur ruang suatu kota, dengan adanya sistem transportasi yang teratur maka kegiatan manusia pada ruang tersebut akan meningkat, oleh karena itu sistem transportasi tidak dapat dipisahkan dari infrastruktur kota baik di daerah perkotaan maupun pedesaan.

## 2.3 *Handphone*

Telepon selular (ponsel) atau telepon genggam (telgam) atau *handphone* (HP) atau disebut pula adalah perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon konvensional saluran tetap, namun dapat dibawa ke mana-mana (portabel, *mobile*) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (nirkabel; *wireless*). Saat ini Indonesia mempunyai dua jaringan telepon nirkabel yaitu sistem GSM (*Global System for Mobile Telecommunications*) dan sistem CDMA (*Code Division Multiple Access*). Badan yang mengatur telekomunikasi seluler Indonesia adalah Asosiasi Telekomunikasi Seluler Indonesia (ATSI).

## 2.4 Java

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

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

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

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

## 2.5 Android

Di kutip dari http://id.wikipedia.org/wiki/Android_(sistem_operasi) Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan untuk digunakan oleh bermacam peranti bergerak (*mobile device*). Hal ini memungkinkan para pengembang menulis kode terkelola (*managed code*) dalam bahasa pemrograman Java, mengontrol peranti via perpustakaan Java yang dikembangkan Google. Android merupakan *subset* perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, middleware, dan aplikasi inti yang di release oleh Google. Sedangkan Android SDK *(Software Development Kit)* menyediakan Tools dan API yang diperlukan untuk Aplikasi Android nantinya tidak akan berjalan langsung di atas *kernel*

sistem operasi namun berjalan di atas Dalvik, sebuah *virtual machine* yang khusus dirancang untuk digunakan pada sistem embedded mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemograman Java.

Pada Juli 2005, Google membeli Android, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, penggabungan dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan Nvidia. Jika sekarang seseorang mempunyai PC di rumah, dan notebook untuk ke kantor atau kuliah, serta berkomunikasi melalui telepon seluler. Maka pergerakkan yang kemudian terjadi sekarang adalah, orang mulai berpikir bagaimana menyatukan semuanya dalam satu genggaman. Sebenarnya hal tersebut telah mulai dipenuhi dengan munculnya PDA/*smartphone*, di mana seseorang dapat merangkum semua kebutuhan komputasinya dalam satu genggaman. Dan perkembangan inilah yang membuat Google berambisi untuk menguasai pangsa ini.

Android memiliki berbagai fitur seperti :

a. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*
b. Mesin virtual Dalvik dioptimalkan untuk perangkat *mobile*
c. *Integrated browser* berdasarkan *engine open source WebKit*
d. Grafts yang dioptimalkan dan didukung oleh *library* grafis 2D, grafis 3D berdasarkan spesifikasi OpenGL
e. *SQLite* untuk penyimpanan data

*f.* *Media Support* yang mendukung audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

*g.* GSM, *Bluetooth, EDGE,* 3G, *dn WiFi* (*hardware* dependent)

*h.* Kamera, GPS, kompas, dan *accelerometer*

*i.* Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator, tools* untuk *debugging*, serta plugin untuk Eclipse IDE.

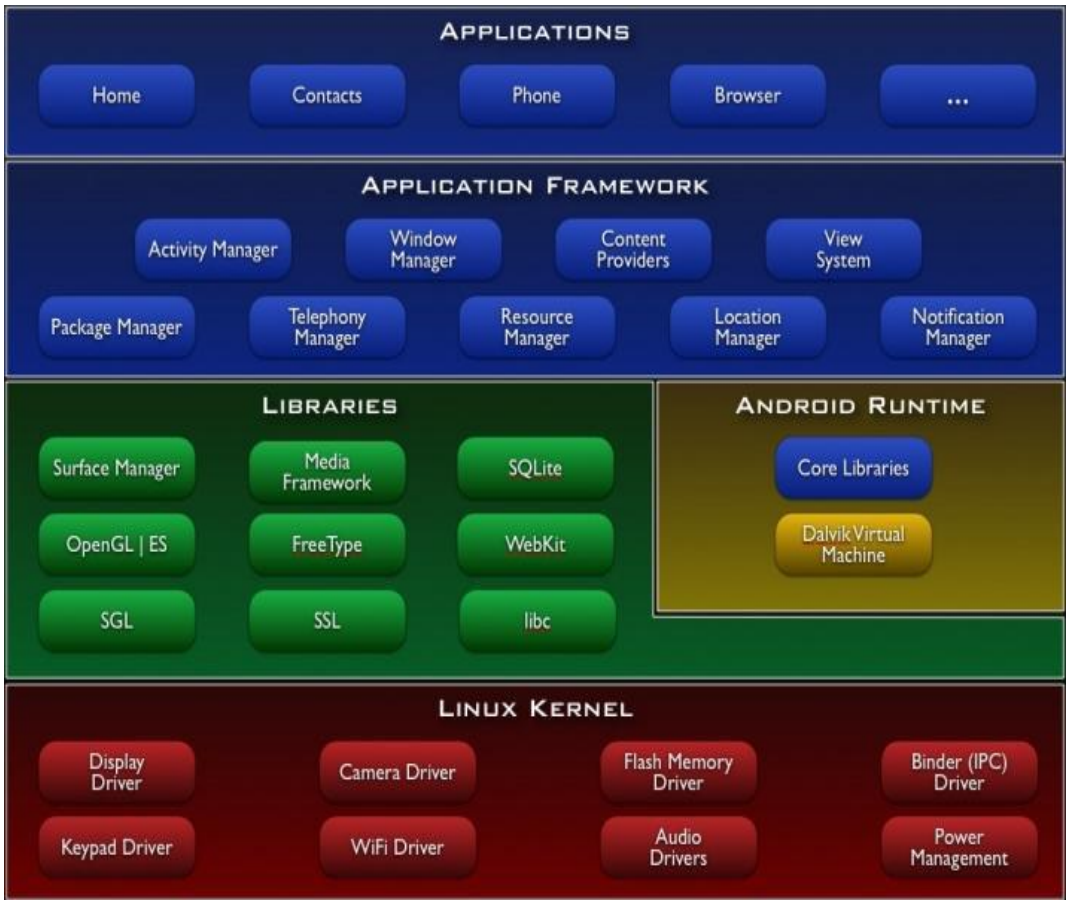
Selain itu menggunakan Android sebagai Operasi Sistemnya memiliki biaya lisensi lebih murah dan sifatnya yang semi *open source*. Tidak hanya itu, Android tentunya akan *support* dengan berbagai layanan dari Google.

### 2.5.1 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.

### 2.5.2 Arsitektur Android

Arsitektur Android menunjukkan komponen-komponen utama yang terdapat pada sistem operasi Android. Berikut ini arsitektur Android :




Gambar 2.1 Arsitektur Android

( Sumber http://www.android-indonesia.com )

**1. Linux *Kernel***

Android bukan Linux, akan tetapi Android dibangun diatas Linux *Kernel*, yaitu versi 2.6 sehingga kehandalannya bisa dipercaya. Untuk inti sistem *service* Linux yang digunakan seperti keamanan, manajemen memori, proses manajemen, *network*, dan *driver* model. Seperti yang terlihat digambar (Gambar II.3), Linux *Kernel* menyediakan *Driver* Layar, Kamera, *Keypad*, WiFi, *Flash Memory*, *Audio*,

dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan keamanan. *Kernel* juga bertindak sebagai lapisan abstrak antara *hardware* dan *software stack*-nya.

**2. *Libraries***

Android menyertakan *libraries* C/C++ yang digunakan oleh berbagai komponen dari sistem Android. Kemampuan ini disediakan kepada *Developer* aplikasi melalui *Framework* Aplikasi Android. Beberapa inti *libraries* tercantum dibawah ini:

1. *System C Library* – Variasi dari implementasi BSD-berasal pelaksana sistem standar C *library* (*libc*), sesuai untuk perangkat *embedded* berbasis Linux.
2. *Media Libraries – PacketVideo* berdasarkan OpenCORE; *library* medukung pemutaran rekaman dan populer banyak format audio dan video, serta *file* gambar, termasuk MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.
3. *Surface Manager* – mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi.
4. LibWebCore – mesin *web modern* yang *powerfull* yang baik *browser* Android *embedded web*.
5. 3D *Libraries* – sebuah pelaksana berdasarkan openGL ES 1,0 API; perpustakaan baik menggunakan perangkat keras akselerasi 3D (apabila tersedia) atau yang disertakan, sangat optimal 3D *software rasterizer*.
6. FreeType – bitmap dan vektor *font rendering*.
7. SQLite – mesin *database* yang kuat dan ringan, dan penghubung tersedia untuk semua aplikasi.

**3. *Android Runtime***

Android terdiri dari satu set *core libraries* yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada *core libraries* bahasa pemrograman Java. Setiap aplikasi menjalankan prosesnya sendiri dalam Android, dengan masing-masing instan dari mesin virtual Dalvik (*Dalvik VM*). Dalvik dirancang agar perangkat dapat menjalankan multiple VMs secara efisien. Mesin Virtual Dalvik mengeksekusi *file* dalam Dalvik *executable* (.dex), sebuah format yang dioptimalkan untuk memori yang kecil.

Dalvik VM berbasis, berjalan, dan dikompilasi oleh *compiler* bahasa Java yang telah ditransformasikan ke dalam .dex format oleh tool "dx" yang telah disertakan. Dalvik VM bergantung pada *kernel* Linux untuk berfungsi seperti *threading* dan manajemen memori tingkat rendahnya.

**4. *Framework* Aplikasi**

Pengembang memiliki akses penuh menuju API *framework* yang sama dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) dengan mudah. Setiap aplikasi dapat memanfaatkan kemampuan ini (sesuai dengan batasan keamanan yang didefinisikan oleh *framework*). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna.

Semua aplikasi merupakan rangkaian set layanan dan sistem, termasuk:

a. *Views* yang kaya dan extensible yang dapat digunakan untuk membangun aplikasi, termasuk list, grids, kotak teks, tombol, dan bahkan sebuah *embedded web*.

b. *Content Provider* yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak telepon) atau membagi data tersebut.

c. *Resource Manager*, yang menyediakan akses ke kode nonsumber lokal seperti string, gambar, dan tata letak *file*.

d. *Notifications Manager* yang memungkinkan semua aplikasi menampilkan *alert* yang bisa dikostumisasi di dalam status bar.

e. *Activity Manager* yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum *backstack*.

**5. *Applications***

Android telah menyertakan aplikasi inti seperti *email client*, SMS, kalender, peta, *browser*, kontak, dan lainnya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman Java. Pada *layer* inilah *developer* atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada Android semua aplikasi baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan pada *layer* aplikasi dengan menggunakan *library* API yang sama. Ini berarti semua aplikasi yang dibuat untuk Android akan memiliki akses yang setara dalam mengakses seluruh kemampuan *handset*, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan *platform* Android ini, *Programmer* atau *Developer* secara penuh akan bisa mengkostumisasi perangkat Androidnya.

### 2.5.3 Perkembangan Android

Dikutip dari http://id.wikipedia.org/wiki/Android_(sistem_operasi) Android telah banyak melakukan perkembangan, berikut adalah perkembangan dari android :

a. 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.

b. Android versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

c. Android versi 1.6 (*Donu*t)

*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.

d. 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.*

e. 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.*

f. 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.

g. 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.

h. 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.

### 2.5.4 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 system 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 system 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.

### 2.5.5. Kekurangan Android

Tak bisa dipungkiri disamping kelebihan tentu saja ada kekurangan dari sistem operasi ini.

1. Koneksi Internet yang terus menerus – Yups, kebanyakan ponsel berbasis system ini memerlukan koneksi internet yang simultan alias terus menerus aktif.Koneksi internet GPRS selalu aktif setiap waktu, itu artinya Anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan.
2. Iklan – Aplikasi di **Ponsel Android** memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut, akan selalu Iklan yang terpampang, entah itu bagian atas atau bawah aplikasi.

## 2.6 Eclipse

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

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

*b.* *Mulit-language:* Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

*c.* *Multi-role:* Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak,

seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

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

### 2.6.1. Arsitektur Eclipse

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

a. *Core platform*

b. OSGi

c. SWT (*Standard Widget Toolkit*)

d. JFace

e. *Eclipse Workbench*

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

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik[2]. 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.

### 2.6.2. Android SDK

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di *release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi-netral, android member anda kesempatan unutk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Hadphone/*Smartphone*. Beberapa fitur-fitur android yang paling penting adalah :

a. *Framework* : aplikasi yang mendukung pengganti komponen dan reusable.
b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat *mobile*

c. Integrated *Browser* verdasarkan engine *open source* WebKit.

d. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional Ekselerasi *hardware*)

e. SQLite untuk penyimpanan data.

f. Media *Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, GIF), GSM Telephony (tergantung *Hardware*)

g. *Bluetooth*, EDGE, 3G, dan WiFi (tergantung *hardware*)

h. Kamera, GPS, Kompas, dan *Accelerometer* (tergantung *hardware*)

i. Lingkungan *Development* yang lengkap dan termasuk pernagkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE Eclipse.

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

### 2.6.3. ADT plugin for Eclipse

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

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

## 2.7. SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat ACID-compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. Tidak seperti pada paradigma client-server umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead, latency times,* dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah *file.* Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

## 2.8. Extensible Markup Language (XML)

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

### 2.8.1. Fungsi dan Tujuan XML

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

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

## 2.9. Unfied Modeling Language (UML)

UML menurut Booch (1999: 14) dalam Jurnal Gintoro, Andreyus, Emilia dan Richard William (2010 , Hal : B-30) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk menvisualisasi, menspesifikasi, merancang dan mendokumentasi sistem piranti lunak.

Unified Modelling Language (UML) menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan UML adalah :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktik-praktik terbaik yang terdapat dalam pemodelan.

UML menyediakan beberapa *notasi dan artifact standar* yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. Artifact didalam UML didefinisikan sebagai informasi dalam bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat. Contohnya adalah source code yang dihasilkan oleh proses pemrograman. Diagram UML, diantaranya adalah :

### 2.9.1. Use Case Diagram

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". *Use Case* dilakukan oleh satu *actor* yang digambarkan dengan simbol orang yang dihubungkan dengan garis yang menunjukan hubungan komunikasi. Setiap *Use Case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksi dengan *Actor*. Nama *Use Case* boleh terdiri dari beberapa kata dan tidak boleh ada *Use Case* yang memiliki nama yang sama.

### 2.9.2. Class Diagram

*Diagram kelas* menggambarkan tipe-tipe objek dalam *system* dan berbagai jenis hubungan atau relasi statis yang ada diantara mereka. Diagram ini memberikan gambaran umum dari sistem. Seperti tipe-tipe dari objek dengan menunjukan kelasnya dan *relationship* yang diantara mereka, serat penjelasan detail tiap-tiap kelas ke dalam model suatu *system. Class diagram* bersifat *static* (tidak berubah) yang akan menunjukan apa itu interaksi tapi tidak menjelaskan apa yang terjadi ketika mereka melakukan interaksi.

### 2.9.3. Activity Diagram

*Activity diagram* memodelkan alur kerja *(work flow).* Sebuah proses bisnis dan urutan aktifitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari aktifitas keaktifitas lainnya atau dari suatu aktifitas kedalam keadaan sesaat (*state)*.

### 2.9.4 Sequence Diagram

*Diagram sequence* merupakan gambaran interaksi antar objek di dalam dan di sekitar sistem berupa *massage* yang digambarkan terhadap waktu. Diagram ini secara khusus berasosiasi dengan *use Case*. *Diagram sequence* juga digunakan untuk menggambarkan skenario atau rangkain langkah-langkah apa yang seharusnya terjadi sebagai respons dari sebuah *event* untuk menghasilkan sesuatu didalam *Use case* sebagai *output*. Untuk *massage* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* dipetakan menjadi operasi/metode dari *class*. *Activation bar* menunjukan lamanya eksekusi sebuat proses, biasanya diawali dengan sebuah *message*. Berikut ini adalah contoh dari *diagram sequence.*

### 2.9.5 *Componenet* Diagram

*Componen Diagram* dapat diartikan sebagai berikut :

1. *Component diagr*am menggambarkan struktur dan hubungan antar komponen peranti lunak, termasuk ketergantungan (*dependency*) diantaranya.

2. Komponen peranti lunak adalah modul berisi code, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, link time maupun *run time*.

3. Pada umumnya komponen terbentuk dari bebrapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil.

4. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

Tipe-tipe *Component* :

Bentuk-bentuk component ada 3 yaitu:

1. *Deployment Component*: Yang menjadi basis dari executable system. Contoh deployment component diantaranya: (*Dynamic Library Link*) file exe, *Active X Control*, *Java Bean* dan lain-lain

2. *Work Product Component*: Yaitu file-file yang dibutuhkan untuk pembuatan deployment component. Contoh untuk component kedua ini diantaranya file data, file *source code* dan lain-lain.

3. *Execution Component*: Yang dibuat sebagai hasil dari sistem yang akan dijalankan.

Menurut Fowler (2004) hal penting pada *component* adalah *component* mewakili potongan-potongan yang independen yang bisa dipesan dan diperbaharui sewaktu-waktu. Dengan demikian, pembagian sistem kedalam component-component lebih banyak didorong oleh kepentingan marketing dari pada kepentingan teknis. Meskipun demikian harus juga diingat bahwa terlalu banyak component juga kurang bagus, karena susah mengatur dan memeliharanya khususnya yang menyangkut masalah *versioning*.