BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi

Menurut Whitten Perancangan Sistem adalah “Proses dimana keperluan pengguna dirubah ke dalam bentuk paket perangkat lunak dan atau kedalam spesifikasi pada komputer yang berdasarkan pada sistem informasi.” (Kristanto, 1994, hal 60).

2.2 E-learning

Pada paradigma tradisional proses belajar mengajar pada umumnya berlangsung di ruang kelas dan ditandai dengan kehadiran pendidik di muka kelas. Pendidik memiliki tanggungjawab penuh terhadap jalannya proses belajar mengajar dan bisa dianggap sebagai sumberdaya paling penting dari sebuah proses belajar mengajar. Sebaliknya pada paradigma baru, peserta didik harus difasilitasi sesuai kebutuhannya masing-masing. Setiap peserta didik adalah spesifik dan memiliki kebutuhan belajar yang berbeda-beda. Proses belajar mengajar harus berfokus pada aktifitas ”belajar” dan bukan pada aktifitas ”mengajar” seperti pada paradigma lama. Dengan paradigma seperti ini maka keberadaan pendidik tidak lagi menjadi satu-satunya faktor penting dalam proses pembelajaran. Keberadaan pendidik bisa digantikan oleh bahan belajar berupa modul, diktat, perangkat lunak edukasi yang bisa digunakan untuk belajar secara mandiri oleh peserta didik. Paradigma baru yang menjadikan peserta didik

sebagai active learner tersebut saat ini mendapatkan sarana yang sesuai untuk diimplementasikan pada sistem pendidikan di Indonesia dengan keberadaan Teknologi Informasi dan Komunikasi (TIK). TIK mampu berperan dalam menghasilkan berbagai produk bahan belajar yang jauh lebih menarik untuk dipelajari, memiliki unsur interaktif yang tinggi, dan mudah dipahami oleh peserta didik. Segala kelebihan tersebut dapat mempercepat proses belajar mereka. Lebih dari itu TIK juga mampu mengantarkan berbagai bahan belajar tersebut ke 8 hadapan peserta didik tanpa batasan jarak dan waktu dengan adanya internet sebagai medianya.Dengan adanya TIK maka telah muncul berbagai model pembelajaran baru dalam dua dekade terakhir.*E-Learning* merupakan suatu jenis belajar mengajar yang memungkinkan tersampaikannya bahan ajar ke siswa dengan menggunakan media *internet*, *intranet*, atau media jaringan *computer* lain (Hartley, 2001). Contoh model tersebut antara lain:

1. Computer Based Learning/Training (CBL/ CBT)

Model CBL/CBT berkembang sekitar pertengahan tahun 1990-an. Saat itu berbagai pelatihan atau kelas menyediakan berbagai bahan belajar berupa modul elektronik baik berupa perangkat lunak edukasi maupun softcopy dari berbagai modul cetak yang sudah ada sebelumnya. Bentuk ini di kemudian hari dikenal sebagai e-book dan berkembang semakin pesat berkat adanya format file pdf dari Adobe. Pada era tersebut CBL/CBT sendiri berkembang pada komputer standalone dan belum terhubung dengan internet. Biasanya pembelajaran dengan model CBL/CBT adalah untuk penyiapan

tenaga ahli pada suatu bidang yang memerlukan pelatihan terlebih dahulu sebelum menempati posisinya.

Perangkat lunak simulasi membantu peserta didik melakukan simulasi atas pekerjaan yang hendak dilakukan. Dengan simulasi maka proses belajar menjadi lebih mudah dan biaya pun bisa ditekan lebih murah dibandingkan apabila mereka harus mempraktekkan sendiri pada peralatan yang sebenarnya.

Modul elektronik mempermudah peserta untuk mempelajari secara mandiri materi yang harus dipelajari dan tidak memerlukan biaya cetak yang tinggi.

2. Web-based Learning

Dengan semakin luasnya perkembangan internet maka perkembangan selanjutnya adalah terjadinya perluasan akses terhadap bahan-bahan belajar CBL/CBT di atas. Berbagai perangkat lunak edukasi ataupun softcopy dari modul, diktat, dan berbagai buku elektronik (e-book) lainnya yang semula didistribusikan dalam bentuk disket atau CD mulai membanjiri internet. Dengan melakukan upload berbagai referensi dan bahan belajar di internet berarti membuka akses dari seluruh penjuru dunia terhadap berbagai bahan belajar tersebut.

Para pengguna internet pun bisa mempelajari apa saja dari berbagai situs web yang tersedia. Demikian pula para penyelenggara pendidikan mulai memanfaatkan internet untuk memperluas layanan mereka pada siapapun yang ingin menjadi peserta didiknya. Berbagai kelas dan pelatihan bisa diikuti

hanya dengan melakukan berbagai download terhadap bahan belajar elektronik, berdiskusi dengan dosen melalui email atau forum-forum diskusi online, dan mengikuti ujian secara online di internet. Setelah lulus sang peserta didik tinggal menunggu ijazah atau sertifikat yang terkirim ke alamatnya. Model inilah yang dikenal sebagai Web-based learning, sebuah model pembelajaran jarak jauh (distance learning) yang menggunakan internet sebagai sarananya.

3. Mobile Learning

TIK tidak hanya terbatas pada penggunaan komputer saja. Berbagai model pembelajaran yang menggunakan peralatan TIK lainnya seperti misalnya telepon genggam pun saat ini telah mulai berkembang. Dengan berbagai fitur dan teknologi yang dimiliki telepon genggam saat ini telah melahirkan sebuah model pembelajaran baru yang dikenal sebagai mobile learning (m-learning).

Aktifitas utama pada M-learning adalah mendistribusikan bahan belajar kepada peserta didik agar dapat diakses menggunakan perangkat komunikasi portabel semacam telepon genggam atau PDA Sedangkan Dong (dalam Kamarga, 2002) 10 mendefinisikan e-learning sebagai kegiatan belajar asynchronous melalui perangkat elektronik komputer yang memperoleh bahan belajar yang sesuai dengan kebutuhannya. Atau e-learning didefinisikan sebagai berikut :

> *e-Learning is a generic term for all technologically supported learning using an array of teaching and learning tools as phone bridging, audio and videotapes, teleconferencing, satellite transmissions, and the more recognized web-based training or computer aided instruction also commonly referred to as online courses*
>
> (Soekartawi, Haryono dan Librero, 2002). Rosenberg (2001) menekankan bahwa e-learning merujuk pada penggunaan teknologi internet untuk mengirimkan serangkaian solusi yang dapat meningkatkan pengetahuan dan keterampilan. Hal ini senada dengan Cambell (2002), Kamarga (2002) yang intinya menekankan penggunaan internet dalam pendidikan sebagai hakekat elearning.
>
> Bahkan Onno W. Purbo (2002) menjelaskan bahwa istilah "e" atau singkatan dari elektronik dalam e-learning digunakan sebagai istilah untuk segala teknologi yang digunakan untuk mendukung usaha-usaha pengajaran lewat teknologi elektronik internet.

E-Learning adalah sistem pendidikan yang menggunakan aplikasi elektronik untuk mendukung belajar mengajar dengan media *internet*, jaringan *computer*, maupun *computer standalone* (Glossary, 2001).

E-Learning sebagai sembarang pengajaran dan pembelajaran yang menggunakan rangkaian elektronik (LAN, WAN, atau *internet*) untuk

menyampaikan isi pembelajaran , interaksi, atau bimbingan (Koran, 2002).

2.2.1 Sejarah E-Learning

E-learning atau *pembelajaran elektronik* pertama kali diperkenalkan oleh universitas llionis di Urbana-Champaign dengan menggunakan sistem instruksi berbasis komputer (*computer assisted instruktion*) dan komputer bernama PLATO. Sejak saat itu, perkembangan *E-Learning* berkembang sejalan dengan perkembangan dan kemajuan teknologi. Berikut perkembangan *E-Learning* dari masa ke masa :

2.2. Tahun 1990 : Pada masa CBT (Computer-Based Training) di mana mulai bermunculan aplikasi *E-Learning* yang berjalan dalam PC *standlone* ataupun berbentuk kemasan CD-ROM. Isi materi dalam bentuk tulisan maupun multimedia (Video dan Audio) dalam format mov, mpeg-1, atau avi.

2.3. Tahun 1994 : Seiring dengan diterimanya CBT oleh masyarakat sejak tahun 1994 CBT muncul dalam bentuk paket-paket yang lebih menarik dan diproduksi secara masal.

2.4. Tahun 1997 : LMS (*Learning Management System*). Seiring dengan perkembangan teknologi *internet*, masyarakat di dunia mulai terkoneksi dengan *internet*. Kebutuhan akan informasi yang dapat diperoleh dengan cepat mulai dirasakan sebagai kebutuhan mutlak dan jarak serta lokasi bukanlah halangan lagi. Dari sinilah muncul LMS. Perkembangan LMS yang makin pesat membuat pemikiran baru untuk mengatasi masalah *interoperability* antar LMS yang satu dengan lainnya secara standar. Bentuk standar yang muncul

misalnya standar yang dikeluarkan oleh AICC (*Airline Industry CBT Commettee*), IMS, IEEE LOM, ARIADNE, dsb.

2.5. Tahun 1999 sebagai tahun aplikasi *E-learning* berbasis *Web*. Perkembangan LMS menuju aplikasi *e-learning* berbasis *Web* berkembang secara total, baik untuk pembelajar (*learner*) maupun administrasi belajar mengajarnya. LMS mulai digabungkan dengan situs-situs informasi, majalah dan surat kabar. Isinya juga semakin kaya dengan perpaduan *multimedia, video streaming* serta penampilan interaktif dalam berbagai pilihan format data yang lebih standar dan berukuran kecil.

2.2.2. Fungsi E-Learning

Ada 3 (tiga) fungsi pembelajaran elektronik terhadap kegiatan pembelajaran di dalam kelas (*Classroom instruction)*, yaitu sebagai suplemen yang sifatnya pilihan / optional, pelengkap (komplemen), atau pengganti (substitusi) (Siahaan, 2002).

a. Suplemen

   Dikatakan berfungsi sebagai supplemen (tambahan), apabila peserta didik mempunyai kebebasan memilih, apakah akan memanfaatkan materi pembelajaran elektronik atau tidak. Dalam hal ini, tidak ada kewajiban / keharusan bagi peserta didik untuk mengakses materi pembelajaran elektronik. Sekalipun sifatnya opsional, peserta didik yang memanfaatkannya tentu akan memiliki tambahan pengetahuan atau wawasan.

b. Komplemen (Pelengkap)

Dikatakan berfungsi sebagai komplemen (pelengkap) apabila materi pembelajaran elektronik diprogramkan untuk melangkapi materi pembelajaran yang diterima siswa di dalam kelas (Lewis, 2002). Sebagai Komplemen berarti materi pembelajaran elektronik diprogramkan utnuk menjadi materi *reinforcement* (pengayaan) atau remedial bagi peserta didik di dalam mengikuti kegiatan pembelajaran konvensional. Materi pembelajaran elektronik dikatakan sebagai *enrichment*, apabila kepada peserta didik yang dapat dengan cepat menguasai / memahami materi pelajaran yang disampaikan guru secara tatap muka (*fast leaners*) diberikan kesempatan untuk mengakses materi pembelajaran elektronik yang memang secara khusus dikembangkan untuk mereka. Tujuannya agar semakin memantapkan tingkat penguasaan peserta didik terhadap materi pelajaran yang disajikan guru didalam kelas. Dikatakan sebagai program remedial, apabila kepada peserta didik yang mengalami kesulitan memahami materi pelajaran yang disajikan guru secara tatap muka di kelas (*Slow learners*) diberikan kesempatan untuk memanfaatkan materi pembelajaran elektronik yang memang secara khusus dirancang untuk mereka.

c. Pengganti (Substitusi)

Beberapa perguruan tinggi di Negara-negara maju memberikan beberapa alternatif model kegiatan pembelajaran / perkuliahan kepada para

mahasiswanya. Tujuannya agar para mahasiswa dapat secara fleksibel mengelola kegiatan perkuliahannya sesuai dengan waktu dan aktivitas lain sehari-hari mahasiswa.

2.2.3. Keunggulan

Keuntungan menggunakan *E-Learning* adalah sebagai berikut :

a. Menghemat waktu proses belajar mengajar.

b. Mengurangi biaya perjalanan.

c. Menghemat biaya pendidikan secara keseluruhan (infrastruktur, peralatan, buku-buku).

d. Menjangkau wilayah geografis yang lebih luas.

e. Melatih pembelajaran lebih mandiri dalam mendapatkan ilmu pengetahuan.

2.2.4. Kekurangan

Pemanfaatan *internet* untuk pembelajaran atau *e-learning* juga tidak terlepas dari berbagai kekurangan. Berbagai keritik (Bullen,2001 dan Beam,1997), antara lain :

1. Kurangnya interaksi antara guru dan siswa atau bahkan antar siswa itu sendiri.
2. Kecenderungan mengabaikan aspek akademik atau aspek social dan sebaliknya mendorong tumbuhnya aspek bisnis / komersial.
3. Proses belajar dan mengajarnya cenderung kearah pelatihan daripada pendidikan.
4. Berubahnya peran guru dari yang semula menguasai teknik pembelajaran

konvensional, juga dituntut mengetahui teknik pembelajaran menggunakan ICT.

5. Siswa yang tidak mempunyai motivasi belajar yang tinggi cenderung gagal.
6. tidak semua tempat tersedia fasilitas *internet*.
7. kurangnya tenaga yang mengtahui dan memiliki keterampilan internet. kurangnya penguasaan bahasa komputer.

2.3 Matematika

Pengertian Matematika (Indonesia) mathematics (Inggris), mathematik (Jerman), mathematique (Perancis), matematico (Itali), matematiceski (Rusia), mathematick atau wiskunde (Belanda) berasal dari bahasa Yunani: mathematikos yaitu ilmu pasti, dari kata mathema atau mathesis yang berarti ajaran, pengetahuan, atau ilmu pengetahuan.

Istilah Matematika menurut bahasa Latin (manthanein atau mathema) yang berarti belajar atau hal yang dipelajari, yang kesemuanya berkaitan dengan penalaran. Matematika adalah salah satu pengetahuan tertua dan dianggap sebagai induk atau alat dan bahasa dasar banyak ilmu. Matematika terbentuk dari penelitian bilangan dan ruang yang merupakan suatu disiplin ilmu yang berdiri sendiri dan tidak merupakan cabang dari ilmu pengetahuan alam.

Pengertian Matematika menurut Roy Hollands, ”matematika adalah suatu sistem yang rumit tetapi tersusun sangat baik yang mempunyai banyak cabang.”

Matematika pada suatu tingkat rendah terdapat ilmu hitung, ilmu ukur dan aljabar (bagian dari matematika dan perluasan dari ilmu hitung, yang banyak digunakan diberbagai bidang disiplin lain, misal fisika, kimia, biologi, teknik, komputer, industri, ekonomi, kedokteran dan pertanian).

Banyak cabang Matematika baru yang bertambah seperti:

- Topologi (cabang-cabang matematika yang mempelajari posisi dan posisi relatif unsur-unsur dalam himpunan),
- Mekanika (suatu cabang ilmu yang mempelajari kerjagayaterhadap benda, kesetimbangan dan gerakan),
- Dinamika (mempelajari penyebab dan sebab benda-benda nyata bergerak),
- Statistika (cabang matematika yang menangani segala macam data numeris yang penting bagi masalah dalam berbagai cabang kehidupan manusia, misal cacah jiwa, angka kematian, angka produktivitas, pertanian, angka perdagangan),
- Peluang (kebolehjadian atau angka banding banyaknya cara suatu kejadian dapat muncul dan jumlah banyaknya semua kejadian yang dapat muncul),
- Analisis (cara memeriksa suatu masalah, untuk menemukan semua unsur dasar dan hubungan antara unsur-unsur yang bersangkutan),
- Serta logika, ilmu ukur segitiga, dan banyak lagi yang lainnya.

Pengertian Matematika tidak hanya berhubungan dengan bilangan-bilangan tetapi lebih luas berhubungan dengan alam semesta. The Liang Gie mengutip

pendapat seorang ahli matematika bernama Charles Edwar Jeanneret yang mengatakan: ”Mathematics is the majestic structure by man to grant him comprehension of the universe”, yang artinya matematika adalah struktur besar yang dibangun oleh manusia untuk memberikan pemahaman mengenai jagat raya.

Dalam belajar matematika diperlukan pemahaman dan penguasaan materi terutama dalam membaca simbol, tabel dan diagram yang sering digunakan dalam matematika serta struktur matematika yang kompleks, dari yang konkret sampai yang abstrak, apalagi jika yang diberikan adalah soal dalam bentuk cerita yang memerlukan kemampuan penerjemahan soal ke dalam kalimat matematika dengan memperhatikan maksud dari pertanyaan soal tersebut.

Belajar matematika merupakan belajar bermakna, dalam arti setiap konsep yang dipelajari harus benar-benar dimengerti/dipahami sebelum sampai pada latihan yang aplikasinya pada materi dan kehidupan sehari-hari. Selain itu ada faktor-faktor yang dapat mempengaruhi proses belajar Matematika.

2.4 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam.Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke

dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM).

Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun".Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

2.5 J2ME

Java 2 Micro Edition atau yang biasa disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak JAVA pada barang elektronik berserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi dengan baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lain. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi yang lebih kecil dibandingkan dengan computer desktop. J2ME biasa digunakan pada telepon selular, pager, PDA, dan sejenisnya.

Teknologi J2ME juga memiliki beberapa keterbatasan jika diaplikasikan pada ponsel. J2ME sangat bergantung pada device yang digunakan, bisa dari merk ponsel, maupun kemampuan ponsel, dan dukungan terhadap teknologi J2ME. Misalnya, jika

sebuah ponsel tidak memiliki kamera maka jelas J2ME pada ponsel tersebut tidak dapat mengakses kamera.

2.6 ANDROID

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-*Mobile*, dan Nvidia.

2.6.1 Sejarah Android

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. versi android terbaru yaitu versi 3.0. Android

juga sudah bergabung dengan beberapa smart *Mobile* seperti Nokia, Sony Ericsson, dan lainnya.

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*Mobile*) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan 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, dukungan Bluetooth 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 indicator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, camcorder dan galeri yang dintegrasikan pada CDMA / EVDO, 802.1x, VPN, Gestures, dan *Textto- speech engine*. Kemampuan dial kontak teknologi *text to change speech* tidak tersedia pada semua ponsel.

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,

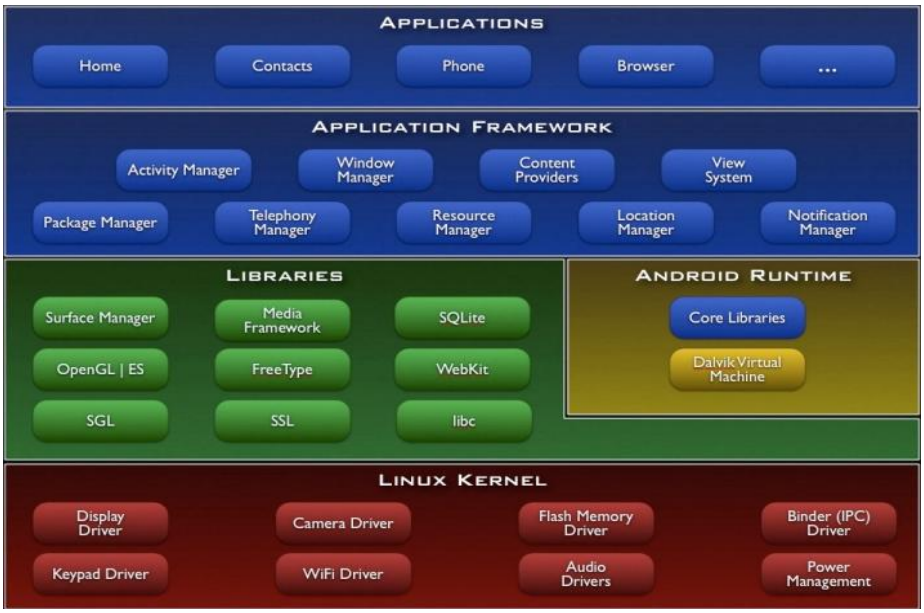
efek audio baru *(reverb, equalization, headphone virtualization, dan bass boost)*, dukungan kemampuan *Near Field Communication (NFC)*, dan dukungan jumlah kamera yang lebih dari satu.

7. *Android* versi 3.0 (*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.

2.6.2 Anatomi Aplikasi Android

Dalam paket sistem operasi android tediri dari beberapa unsur seperti tampak pada gambar 2.8. Secara sederhana arsitektur android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *Framework* yang menyediakan dan mengatur alur proses aplikasi.



(Mulyadi,2010,http://ayobelajarilmukomputer.wordpress.com/2011/04/16/Karakteristik-android/)

Gambar 2.1 *Detail* Anatomi Android

2.6.2.1 Linux Kernel

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

2.6.2.2 Libraries

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

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

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

2.6.2.3 Android Runtime

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

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

2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable* (*.dex).

Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

2.6.2.4 Aplication Framework

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

1. *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”*Backstack“* untuk navigasi penggunaan.

2. *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.

3. *Resuource Manager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan *file layout*.

4. *Location Manager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.

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

2.6.2.5 Aplication Layer

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

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

2.6.3 Komponen Aplikasi Android

Fitur penting android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur *scroller* dan aplikasi lain telah mengembangkan fitur *scroller* yang

baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan *scroller* yang telah ada.

Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi main() pada aplikasi lain.

Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan system untuk memanggil dan menjalankan ketika dibutuhkan.

2.6.3.1 Activities

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *Activity* menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap *Event*. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan.

Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack.* Keadaan suatu *activity* ditentukan oleh posisinya dalam tumpukan *acitivity,* LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back,* atau menutup *activity* yang sedang

digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

Ketika *activity* diambil dan disimpan dalam tumpukkan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi :

1) ***Active,*** setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukkan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.

2) ***Paused,*** dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused.* Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat *active* namun tidak dapat menerima masukkan dari pengguna. Dalam kasus ekstrim, android akan menghentikan *activity* dalam keadaan *paused* ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.

3) ***Stopped,*** ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped. Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu *activity* dalam kondisi

stopped maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi *inactive.*

4) ***Inactive,*** kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah ditiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali.

Kondisi transisi ini sepenuhnya ditangani oleh manajer memori android. Android akan memulai menutup aplikasi yang mengandung *activity inactive*, kemudian *stopped activity*, dan dalam kasus luar biasa *paused activity* juga akan di tutup.

2.6.3.2 Services

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

2.6.3.3 Intents

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain sebagainya. *Intents* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. *Intents* dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin menampilkan *URL*, sistem akan menentukan komponen apa yang dibutuhkan oleh *Intents* tersebut.

2.63.4 Broadcast Receiver

Broadcast Receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *Broadcast* berasal dari sistem misalnya, Batre sudah hampir habis, informasi zona waktu telah berubah, atau pengguna telah merubah bahasa *default* pada perangkat. Sama halnya dengan *service*, *Broadcast Receivers* tidak menampilkan antarmuka pengguna. Namun, *Broadcast Receivers* dapat menggunakan *Notification Manager* untuk memberitahukan sesuatu kepada pengguna.

2.6.3.5 Content Providers

Content Providers digunakan untuk mengelola dan berbagi database. Data dapat disimpan dalam file sistem, dalam database *SQLite*, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *Content Provider* memungkinkan antar aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya.

2.6.4 Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada android :

1. Foreground Activity

Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini pasti mempertimbangkan siklus hidup *activity*, sehingga perpindahan antar *activity* dapat berlangsung dengan lancar.

2. Background Service

Aplikasi yang memiliki interaksi terbatas dengan user, selain dari pengaturan konfigurasi, semua dari prosesnya tidak tidak tampak pada layar. Contohnya aplikasi penyaringan panggilan atau sms auto respon.

3. Intermittent Activity

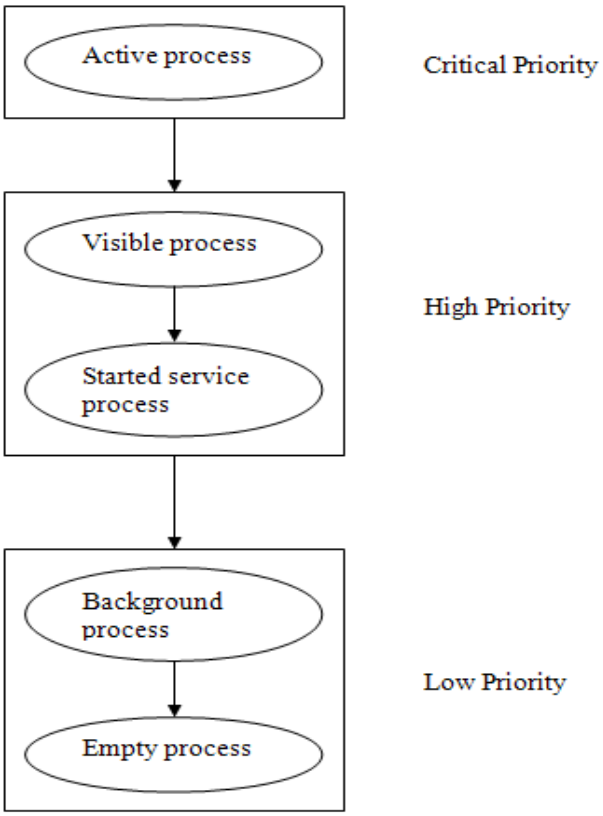
Aplikasi yang masih membutuhkan beberapa masukkan dari pengguna, namun sebagian sangat efektif jika dijalankan di *background* dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu. Contohnya pemutar musik.

Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi aplikasi memiliki ciri-ciri dari semua kategori. Oleh karenanya perlu pertimbangan bagaimana aplikasi tersebut digunakan dan menentukan kategori aplikasi yang sesuai.

2.6.5 Siklus Hidup Android

Siklus hidup aplikasi android dikelola oleh sistem, berdasarkan kebutuhan pengguna, sumberdaya yang tersedia, dan sebagainya. Misalnya Pengguna ingin menjalankan browser web, pada akhirnya sistem yang akan menentukan menjalankan aplikasi. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika pengguna ketika itu sedang menjalankan sebuah *Activity*, maka sistem akan memberikan perioritas utama untuk aplikasi yang tersebut. Sebaliknya, jika suatu *Activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *Activity* yang prioritas rendah akan ditutup.

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin virtual pengolah sendiri, dengan ini melindungi penggunaan memori pada aplikasi. Selain itu juga android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karenanya android sangat sensitive dengan siklus hidup aplikasi dan komponen-komponennya. Perlu adanya penanganan terhadap setiap kondisi agar aplikasi menjadi stabil. Gambar 2.2 menunjukkan prioritas dari aplikasi.



(Mulyadi,2010,http://ayobelajarilmukomputer.wordpress.com/2011/04/16/)

Gambar 2.2 Prioritas aplikasi berdasarkan *activity*

2.6.6 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.

2.7 Eclipse

2.7.1 Pengertian 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 sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

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

c. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bias 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*.

2.7.2 Arsitektur Eclipse

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

a. *Core platform*

b. OSGi

c. SWT (*Standard Widget Toolkit*)

d. JFace

e. *Eclipse Workbench*

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan

PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

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.

2.7.3 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.

2.8 SMARTPHONE

Menurut Renan Prasta Jenie, "*A smartphone is a device that lets you make telephone calls, but also adds features that you might find on a personal digital assistant or a computer. A smartphone also offers the ability to send and receive e-mail and edit Office documents, for example. While there is no standard definition of the term "smartphone" across the industry, there are a few features that you'll find on every smartphone. Here's a list: Operating System: A smartphone will be based on an operating system that allows it to run productivity applications, such as the BlackBerry OS, the Palm OS, or Windows Mobile Software: A smartphone will offer*

more than just an address book. It may allow you to create and edit Microsoft Office documents or manage your finances. Web Access: A smartphones offer some sort of access to the Internet. You can use your smartphone to browse your favorite Web sites. QWERTY Keyboard: A smartphone includes a QWERTY keyboard, so the keys are laid out in the same manner they would be on your computer keyboard. Messaging: All cell phones can send and receive text messages, but what sets a smartphone apart is its ability to send and receive e-mail. Some smartphones can support multiple e-mail accounts. Others include access to the popular instant messaging services, like AOL's AIM and Yahoo! Messenger." (*Bina Nusantara University*, 2008, hal 2)

2.9 Unified 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.*