BAB II
LANDASAN TEORI

2.1. Aplikasi

Aplikasi merupakan penerapan, menyimpan sesuatu hal, data permasalahan, pekerjaan kedalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasika hal atau permasalahan tersebut sehinggga berubah menjadi bentuk yang baru tanpa meenghilangkan nilai-nilai dasar hal, data, permasalahan atau pernasalahan atau pekerjaan. Jadi dalam hal ini hanya bentuk dari tampilan data yang berubah, sedangkan isi yang termuat dalam data tersebut tidak mengalami perubahan. Jadi program aplikasi adalah sederetan kode yang digunakan untuk mengatur komputer supaya dapat melakukan pekerjaan sesuai dengan keinginan programmer atau user.

Atau definisi lain aplikasi merupakan kumpulan dari prosedur-prosedur yang digunakan untuk mengolah data menjadi informasi. Misalnya penjumlahan, klasifikasi, rotasi, koreksi geometri, query, *overlay*, *buffer*, *jointable* dan sebagainya.

Menurut Daryanto (2004:347), aplikasi adalah *software* atau perangkat lunak yang dibuat untuk mengerjakan menyelesaikan masalah-masalah khusus. Sedangkan menurut Jogiyanto (2004:4), aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jadi aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program.

Program inilah yang mengendalikan semua aktifitas yang ada pada proses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instructionset*.

2.2. Pariwisata

2.2.1. Definisi Pariwisata

Kata pariwisata baru populer pada tahun 1958. Sebelum itu digunakan kata *turisme,* serapan dari Bahasa Belanda *"tourisme"*. Sejak 1958 resmilah kata pariwisata sebagai padanan *tourisme* (Bld) atau *tourism* (Ing). Perkembangan dan pengayaan makna selanjutnya adalah hadirnya istilah darmawisata, karyawisata, widyawisata, yang semuanya mengandung unsure "wisata". Menurut Kamus Bahasa Indonesia Kontemporer (KBLK) (1992), **Wisata** (*vi Skr*) berarti; bepergian bersama – sama untuk bersenang dan sebagainya; bertamasya; piknik; **wisatawan** (*n*) adalah orang yang berdarmawisata; pelancong; turis. Yoeti mengartikan wisata adalah perjalanan sebagai padanan kata "*travel*" sehingga **wisatawan** adalah "*traveler*", orang yang melakukan perjalanan. (Suwardjoko P. Warpani, Indira P. Warpani, 2007:5).

Adapun berbagai definisi Pariwisata :

a. Undang – Undang No. 10 Tahun 2009 :

   "Pariwisata adalah berbagai macam kegiatan wisata dan didukung berbagai fasilitas serta layanan yang disediakan oleh masyarakat, pengusaha, Pemerintah, dan Pemerintah Daerah".

b. Nyoman S. Pendit (2003:32) :

"Pariwisata adalah salah satu jenis industry baru yang mampu mempercepat pertumbuhan ekonomi dan penyediaan lapangan kerja, peningkatan penghasilan, standar hidup serta menstimulasi sector – sector produktif lainnya. Selanjutnya sebagai sector yang kompleks, ia juga merealisi industri – industri klasik seperti industry kerajinan tangan dan cinderamata. Penginapan dan transportasi secara ekonomis juga dipandang sebagai industri".

2.2.2. Jenis Pariwisata

Batasan pariwisata menurut Suwardjoko P. Warpani dan Indira P. Warpani (2007:13-14) sangat luas dan sesuai dengan maksud berwisata atau kegiatan yang dilakukan oleh wisatawan, maka pariwisata dikategorikan menjadi :

a. Wisata Agro; dapat dikatakan sebagai jenis pariwisata baru yang dikaitkan dengan kegiatan industry pertanian, misalnya wisata durian pada musim buah durian, atau wisata tani, yakni para wisatawan turut terjun aktif menanam padi dan memandikan kerbau di sungai.
b. Wisata Belanja; dilakukan karena kekhasan barang yang ditawarkan atau bagian dari jenis pariwisata lain, misalnya Bandung dengan Pusat Jeans di Jl. Cihampelas, Sidoarjo dengan pusat Tas di Tanggulangin.
c. Wisata Budaya; berkaitan dengan ritual budaya yang sudah menjadi tradisi, misalnya; mudik lebaran setahun sekali. Atau ada peristiwa budaya yang digelar pada saat – saat tertentu, misalnya : Sekaten di Surakarta dan Yogyakarta, Ngaben di Bali, Labuhan di Cilacap. Tidak jarang wisatawan

mempelajari budaya setempat, mengunjungi situs bersejarah, dan sebagainya.

d. Wisata Iklim; bagi negara beriklim empat, pada saat tertentu benar – benar dimanfaatkan untuk melakukan perjalanan mengunjungi tempat – tempat lain hanya untuk 'berburu' panas sinar matahari. Bagi masyarakat tropis seperti Indonesia, kunjungan ke suatu tempat berkaitan dengan maksud mencari perubahan iklim setempat.
e. Wisata Karya; kunjungan kerja, yaitu jenis pariwisata yang para wisatawannya berkunjung dengan maksud dinas atau tugas – tugas lain, misalnya : peninjauan / inspeksi daerah, sigi lapangan. Maksud kedatangan seseorang atau sejumlah orang di suatu Daerah Tujuan Wisata (DTW) memang untuk melaksanakan tugas profesi / pekerjaannya, namun dalam waktu senggang, atau sengaja diacarakan, mereka melakukan rekreasi atau kunjungan wisata ke beberapa objek.
f. Wisata Kesehatan; berhubungan dengan maksud penyembuhan suatu penyakit. Wisatawan mengunjungi suatu tempat karena keberadaan penyembuh.
g. Wisata Konvensi / Seminar; dilakukan dengan sengaja memilih salah satu daerah tujuan wisata sebagai tempat penyelenggaraan seminar dikaitkan dengan upaya pengembangan daerah tujuan wisata yang bersangkutan. Penentuan lokasi tempat penyelanggaraan suatu konvensi, baik nasional maupun internasional, sering dikaitkan dengan kebijakan pemerintah mempromosikan suatu daerah tujuan wisata. Kebijakan pemilihan lokasi

penyelenggaraan konvensi sangat jelas diwarnai oleh kepentingan pariwisata.

h. Wisata Niaga; berkaitan dengan kepentingan perniagaan (usaha perdagangan). Wisatawan datang karena ada urusan perniagaan di tempat tersebut. Seperti halnya wisata dinas, para pengusaha / niagawan datang dengan maksud utama melakukan kegiatan perniagaan namun pada waktu luang pada umumnya berwisata.

i. Wisata Olahraga; yakni mengunjungi peristiwa penting di dunia olahraga, misalnya pertandingan perebutan kejuaraan, Pekan Olahraga Nasional, Asean Games, Olimpiade, atau sekadar pertandingan persahabatan. Para wisatawan adalah para olahragawan, penonton, dan semua yang terlibat dalam peristiwa olahraga.

j. Wisata Pelancongan / rekreasi; dilakukan untuk berlibur, mencari suasana baru, memuaskan rasa ingin tahu, melihar sesuatu yang baru, menikmati keindahan alam, melepaskan ketegangan. Maksudnya adalah memulihkan kesegaran dan kebugaran jasmani dan rohani setelah berwisata.

k. Wisata Petualangan; dilakukan lebih kea rah olahraga yang sifatnya menantang kekuatan fisik dan mental para wisatawan. Termasuk dalam jenis wisata petualangan adalah kegiatan pelatihan (kepemimpinan) di alam terbuka dengan berbagai atraksi yang menantang dan kadang – kadang mengundang risiko.

l. Wisata Ziarah; dalam kaitan dengan agama atau budaya. Mengunjungi tempat ibadah atau tempat ziarah pada waktu tertentu.

m. Darmawisata; perjalanan beramai – ramai untuk bersenang – senang, atau berkaitan dengan pelaksanaan darma di luar ruangan, atau ekskursi; atau melaksanan pengabdian kepada masyarakat di luar waktu kerja sehari – hari.

n. Widiawisata (pendidikan); perjalanan ke luar (daerah, kampong, dsb) dalam rangka kunjungan studi; dilakukan untuk mempelajari seni-budaya rakyat, mengunjungi dan meneliti cagar alam dan budaya, atau untuk kepentingan menuntut ilmu selama waktu tertentu.

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

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.

2.5.1. 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.

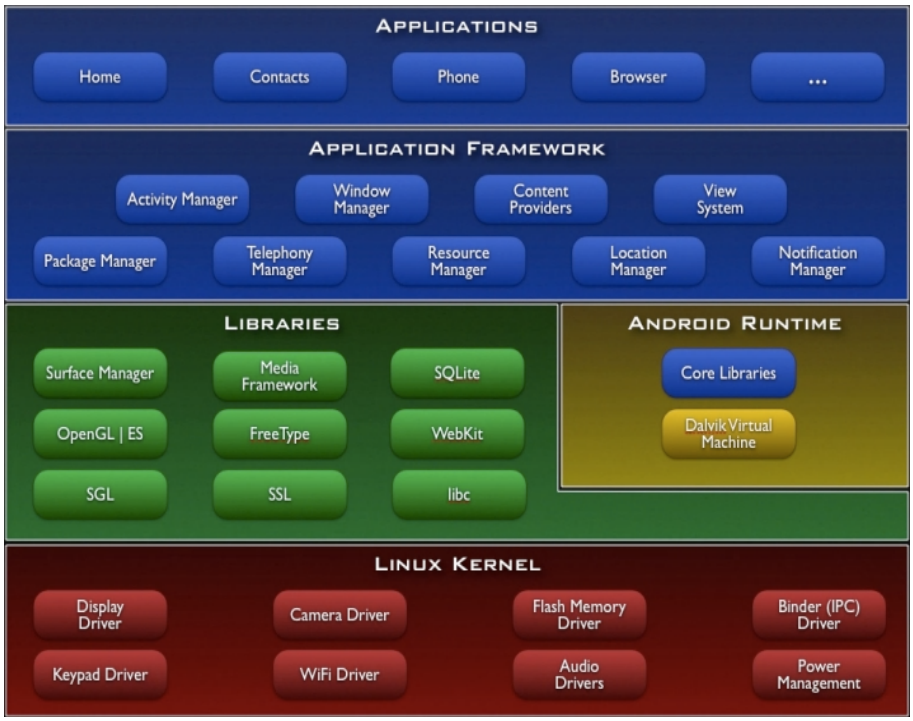
7. Android versi 3.0 (Honeycomb)

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.

2.5.2. Anatomi Aplikasi Android

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



Gambar 2.1 *Detail* Anatomi Android

(Sumber: http:/id.scribd.com/)

2.5.2.1. 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.

2.5.2.2. 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).

2.5.2.3. 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.

2.5.2.4. 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*.

2.5.2.5. Aplication Layer

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*.

2.5.3. 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.

2.5.3.1. *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.

2.5.3.2. *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.

2.5.3.3. *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.

2.5.3.4. *Broadcast Receiver*

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

2.5.3.5. 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.

2.5.4. 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*

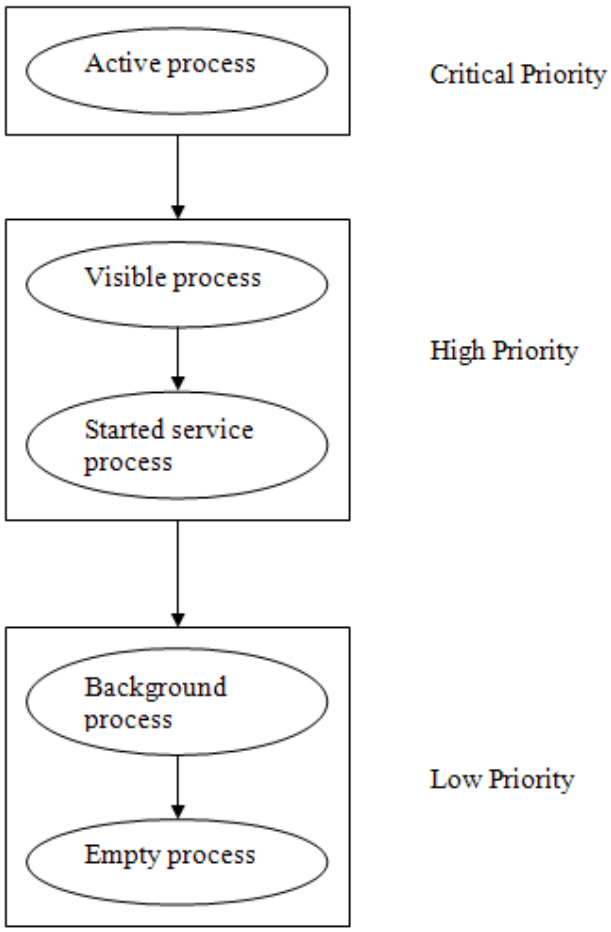
Aplikasi yang masih membutuhkan beberapa masukkan dari pengguna, namun sebagian sangat efektif jika dijalankan di *background* dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu. Contohnya pemutar musik.

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.

2.5.5. 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.



Gambar 2.2 Prioritas Aplikasi Berdasarkan *Activity*

(Sumber: http:/eepis-its.edu/)

2.5.6. 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.

2.6. 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:

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 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*.

2.6.1. 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.

2.7. MySQL

MySQL adalah salah satu jenis database *server* yang sangat terkenal, kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Selain itu , ia bersifat *Open source* (Anda tidak perlu membayar untuk menggunakannya) pada pelbagai *platform* (kecuali untuk jenis enterprise, yang bersifat komersial). Perangkat lunak MySQL sendiri bisa di *download* dari http://www.mysql.com MySQL termasuk jenis RDBMS (*Relational Database Management*). Itulah sebabnya, istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah database mengandung

satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

2.8. 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.

2.8.1. *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.

2.8.2. *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.

2.8.3. *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)*.

2.8.4. 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.*

2.8.5. Deployment Diagram

Diagram *deployment* atau *deployment diagram* menunjukan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digumakan untuk memodelkan hal-hal berikut

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware.*
2. Sistem *client*/*server*
3. Sistem terdistribusi murni
4. Rekayasa ulang aplikasi

(Sumber :Buku Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Pengarang Rosa A.S – M.Shalahuddin Tahun 2011).