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 mengimplementasikan 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. *E-ticketing*

E-ticketing atau *elektronic ticketing* adalah suatu cara untuk mendokumentasikan proses penjualan dari aktifitas perjalanan tanpan harus mengeluarkan dokumen berharga secara fisik ataupun kertas tiket. Semua informasi mengenai *electronic ticketing* disimpan secara *digital* dalam sistem komputer.

E-ticketing ialah peluang untuk meminimalkan biaya dalam mengoptimalkan kenyamanan penumpang. *E-ticketing* mengurangi biaya proses tiket, menghilangkan formulir kertas dan meningkatkan flekibilitas penumpang dan agen perjalanan dalam membuat perubahan-perubahan dalam jadwal keberangkatan.

2.2.1 Manfaat *E-ticketing*

E-ticketing membawa informasi yang sama dengan tiket kertas. Perbedaan utama adalah terletak pada database komputer maskapai penerbangan tersebut. *e-ticket* adalah catatan electronic resevasi perjalanan, mengandung informasi, seperti tanggal, waktu, berat bagasi, besar biaya perjalanan dan kelas kursi. Sewaktu *check-in* di bandara, penumpang perlu menunjukkan *e-ticket* disertai identitas diri (KTP/SIM), setelah itu dari pihak maskapai akan memberikan boarding pass untuk masuk ke ruang tunggu dan pesawat.

E-ticketing memberikan keuntungan baik penumpang maupun maskapai penerbangan, mulai dari aspek keamanan, fleksibilitas, biaya, dan kenyamanan. Selain itu e-ticket memberikan standard jaminan yang sama dengan tiket kertas.

Berbeda dengan tiket kertas yang bisa ketinggalan di kantor, e-ticket tidak mungkin untuk "kehilangan" karena berada dalam jaringan database biro perjalanan. Selain itu, *e-ticket* sulit untuk dicuri. Penumpang biasanya mencetak *e-tiket* mereka, termasuk konfirmasi *e-mail*, jadwal dan dokumen lainnya di rumah masing-masing. Apabila tiket yang telah dicetak hilang, kita dapat melakukan print ulang tiket yang tersimpan pada komputer dan hanya orang yang sesuai dengan identitas yang tertulis di *e-ticket* yang dapat menggunakannya. Dengan tiket kertas, penumpang yang hilang atau lupa mungkin akan dikenakan biaya oleh maskapai penerbangan untuk membuat yang baru. Dalam beberapa kasus, penumpang bahkan diharuskan membeli tiket baru pada full-harga. *E-tiket* menawarkan keuntungan yang berbeda pada situasi seperti ini.

Selain itu e-*ticket* dipesan dan diproses dengan lebih tepat waktu, menghemat jam kerja dan menghilangkan rasa khawatir dan frustrasi calon penumpang terhadap kepastian tiket. Keuntungan juga didapatkan setelah transaksi terjadi. Maskapai penerbangan dapat melacak dan memberitahu perubahan jadwal penerbangan, pembatalan dan perubahan-perubahan lainnya lebih cepat.

Kerugian e-ticket dapat terjadi apabila sistem jaringan komputer crash(rusak). Hal ini menyebabkan data reservasi bisa hilang. Beberapa biro perjalanan biasanya memiliki jaringan *backup* untuk menghadapi kondisi seperti ini. Selain itu, karena

tingkat efisiensi yang tinggi dapat menyebabkan *travel agent* gulung tikar dan menambah pengangguran. Lebih lanjut lagi, *e-ticket* juga menjadi kendala bagi turis asing yang kadang-kadang di beberapa negara memerlukan tiket pulang untuk memastikan mereka tidak melanggar hukum imigrasi mereka.

Walaupun ada kerugian yang juga ditimbulkan, hal ini juga membuat munculnya lapangan pekerjaan yang baru. Dapat dilihat dari banyaknya yang menawarkan pembelian tiket secara on-line untuk orang-orang yang tidak mau repot-repot dan tidak memiliki jaringan internet.

Penggunaan e-ticket memiliki banyak keuntungan dan kerugian yang harus dibenahi. Penerapan di Indonesia sudah sangat laus dan diharapkan dapat berkembang lebih baik lagi kedepannya.

2.3. Biro Perjalanan

2.3.1. Definisi Biro Perjalanan

Biro perjalanan adalah kegiatan usaha yang bersifat komersial yang mengatur, dan menyediakan pelayanan bagi seseorang, sekelompok orang, untuk melakukan perjalanan dengan tujuan utama berwisata. Biro perjalanan umum adalah badan usaha yang menyelenggarakan kegiatan perjalanan usaha di dalam dan ke luar negeri.

Fungsi umum biro perjalanan ialah merupakan badan usaha yang memberikan penerangan atau informasi tentang segala sesuatu yang berhubungan dengan dunia perjalanan pada umumnya dan perjalanan wisata khususnya, sedangkan fungsi khusus biro perjalanan ialah merencanakan dan

meyelenggarakan *tours* dengan tanggung jawab dan resikonya sendiri serta aktif melakukan kerjasama dengan perusahaan lain.

Secara umum perusahaan perjalanan adalah sebuah perusahaan yang mempunyai tujuan mengusahakan dan mengurus perjalanan seseorang dengan segala kebutuhan dari perjalanan itu, baik darat, laut dan udara serta mendapat imbalan dari perusahaan penyedia fasilitas perjalanan atas pelayanan kepada orang yang melakukan perjalanan. Menurut Nyoman S Pendit dalam bukunya *Pengantar dan Pengertian Ilmu Pariwisata*, memberikan batasan antara lain : Perusahaan perjalanan yaitu perusahaan yang mempunyai tujuan untuk menyiapkan suatu perjalanan bagi seseorang yang merencanakan untuk mengadakan suatu perjalanan.

Usaha biro perjalanan di Indonesia meliputi dua bidang, antara lain :

1. Biro perjalan umum yaitu perusahaan perjalanan yang dapat melakukan kegiatan usaha.
2. Agen perjalanan yaitu perusahaan perjalanan yang usahanya hanya menangani perusahan penerbangan, hotel, biro perjalanan umum.

2.3.2. Prosedur Kerja Usaha Jasa Biro Perjalanan

Adapun prosedur kerja dalam usaha jasa biro perjalanan yaitu:

a. Perusahaan perjalanan berperan dalam menjembatani kepentingan orang yang mengadakan perjalanan dengan perusahaan yang menyediakan fasilitas dan sarana perjalanan.

b. Dalam usahanya perusahaan perjalanan bertindak untuk kepentingan orang lain. Kepentingan orang yang mengadakan perjalanan dan kepentingan perusahaan yang menyediakan fasilitas perjalanan.
c. Perusahaan perjalan merupakan *'Department Store of Travel"* dalam arti mampu menyediakan segala urusan yang menyangkut perjalanan.
d. Perusahaan perjalanan adalah sebuah perusahaan yang mempunyai tujuan memperoleh keuntungan dari kegiatannya.

2.4 Handphone

Telepon genggam atau Handphone adalah sebuah perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon fixed line sehingga konvesional namun dapat dibawa keman-mana (portable) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (nirkabel, wireless).

Generasi pertama system selular Analog yaitu AMPS (Advance Mobile Phone Service). Versi dari AMPS dikenal sebagai Narrowband Advance Mobile Phone Service (NAMPS) yang menggabungkan teknologi digital, sehingga system ini dapat digunakan untuk membawa tiga kali lebih besar kapasitas pada setiap panggilan versinya. Pada tahun 1981 muncul NMT (Nordic Mobile Telephone System). Pada tahun 1982 muncullah GSM (Global System For Mobile Communination).

Pada tahun 1990 jaringan Amerika Utara bergabung membentuk standarisasi IS-54B dimana standarisasi ini adalah yang pertama kali menggunakan dual mode

seluler berdasarkan teknik penyebaran spectrum untuk meningkatkan kapasitas yang disebut IS-95. Dengan menggunakan protocol AMPS sebagai defaultnya, akan tetapi mempunyai cara kerja SEC. Normal yang berbeda dengan analaog selular serta lebih canggih dibanding IS-54**.**

2.5. Java

Java menurut Sun Microsystem adalah nama untuk sekumpulan teknologi untuk membuat dan menjalanakan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan (Salahuddin, M., A.S. Rosa, 2010). Java2 adalah generasi kedua dari *Java Platform* (generasi awalnya adalah *Java Development Kit*). Java berdiri di atas mesin interpreter yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu, bahasa Java disebut sebagai bahasa pemograman yang portable karena dapat dijalankan pada berbagai sistem operasi, dengan syarat sistem operasi tersebut terdapat JVM (*Java Virtual Machine*).

` *Platform Java* terdiri dari kumpulan library, JVM, kelas-kelas loader yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah *compiler*, *debuger* dan kakas lain yang dipaket dalam *Java Development Kit* (JDK). Agar sebuah program Java dapat dijalankan, maka file dengan ekstensi .java harus dikompilasi menjadi file *bytecode*. Untuk menjalankan bytecode tersebut dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan

program Java, hanya menjalankan tidak untuk membuat kode baru lagi. JRE berisi JVM dan *library* Java yang digunakan.

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.

2.6. Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti 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

Dibawah ini adalah pengertian dan definisi android menurut beberapa sumber yang dihimpun :

1. www.android.com - Android merupakan kekuatan baru bagi jutaan ponsel, tablet, dan perangkat lain yang membawa kecepatan Google dan web ke tangan anda.

2. Wikipedia - Android adalah sistem operasi untuk telepon seluler yang berbasis Linux
3. Matamaya Studio - Anroid merupakan *operating system* dari Google yang bersifat *open source*, sehingga berbeda dengan *windows* dimana kita harus membeli lisensinya
4. Jubilee Enterprise - Android adalah sebuah sistem operasi yang memberi kemudahan dalam berkirim email melalui fasilitas Gmail Android merupakan sistem operasi *mobile* berbasis kernel Linux yang dikembangkan oleh Android Inc dan kemudian diakuisisi oleh Google.
5. Wei-Meng Lee - Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka.
6. www.developer.android.com - Android adalah *software* untuk perangkat *mobile* yang mencakup aplikasi sistem operasi, *middleware*, dan *key*.
7. Ajith Abraham, Jamie Lloret Mauri & John Buford - Android adalah sistem operasi milik Google. Sistem operasi ini berbeda dengan sistem operasi yang sebelumnya bisa digunakan pada mobile *devices, notebook,* dan komputer.

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)

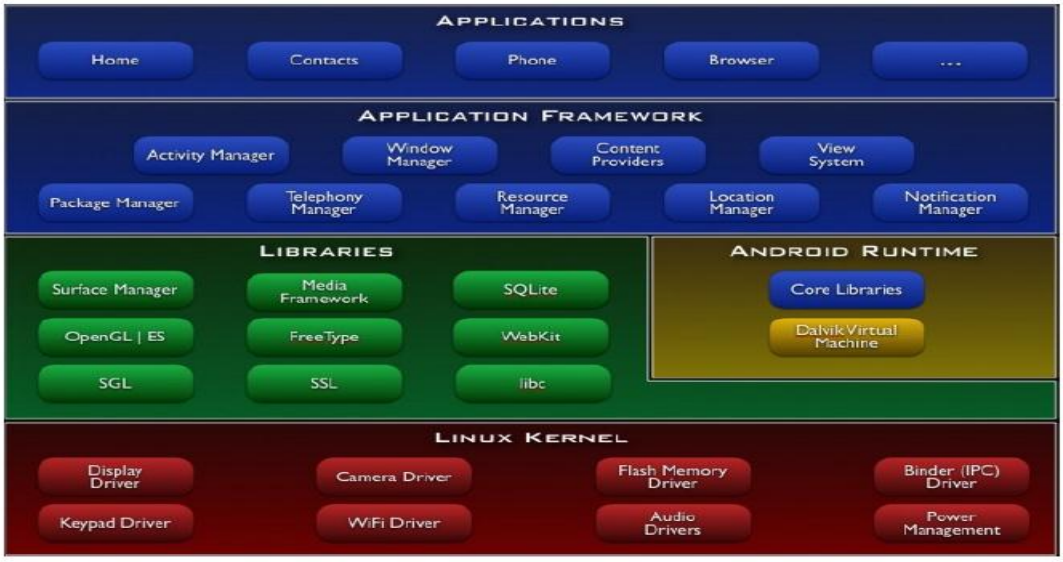
Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, 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. Arsitektur Android

Gambar di bawah ini menampilkan komponen utama dari sistem operasi Android.



Gambar 2.1 *Detail* Anatomi Android

(Sumber: http://ervitakusumaputri228.wordpress.com)

2.6.2.1.Aplication

Android akan disertakan dengan satu set aplikasi inti termasuk *email client*, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2.6.2.2.Application Framework

Dengan menyediakan *platform* pengembangan yang terbuka, Android menawarkan developer kemampuan untuk membangun aplikasi yang benar-benar kaya dan inovatif. Developer bebas untuk mengambil keuntungan dari perangkat keras, mengakses informasi lokasi, menjalankan layanan latar, menset alarm, menambah notifikasi ke status bar, dan banyak lagi yang lain.

Developer mempunyai akses penuh ke *framework* API yang sama yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang untuk memudahkan

penggunaan ulang komponen, beberapa aplikasi dapat mengeluarkan kemampuannya dan beberapa aplikasi yang lain kemudian menggunakan kemampuan tersebut (tergantung pada batasan keamanan yang ditegakkan oleh *framework)*. Mekanisme yang sama ini membolehkan komponen diganti oleh pengguna.

Mendasari semua aplikasi adalah sesusunan layanan dan sistem, termasuk:

- View yang kaya dan extensibel yang bisa digunakan untuk membangun sebuah aplikasi, meliputi, *list, grid, text box, button*, dan bahkan web browser yang bisa ditanam.
- *Content Provider* yang membolehkan aplikasi mengakses data dari aplikasi yang lain (seperti *Contact*), atau untuk berbagi data aplikasi itu sendiri.
- *Resource Manager*, menyediakan akses ke *non-code resource*, seperti berkas lokalisasi string, graphic, dan layout.
- *Notification Manager* yang membolehkan seluruh aplikasi untuk menampilkan peringatan di status *bar*.
- *Activity Manager* mengatur daur hidup aplikasi dan menyediakan navigasi backstack yang umum.

2.6.2.3. *Library*

Android meliputi sesusunan kepustakaan C/C++ yang digunakan oleh beberapa komponen sistem Androi. Kemampuan-kemampuan ini ditampilkan ke developer melalui Android application framework. Beberapa dari kepustakaan inti akan disebutkan di bawah ini:

- *System C library* – sebuah implementasi standar C system library (libc) turunan BSD, disesuaikan dengan perangkat berbasis Linux *embedded*.
- Media library – berdasarkan OpenCORE PacketVideo, library ini mendukung playback dan recording dari banyak format audio dan video

yang terkenal, sebaik file image. *Library* ini meliputi MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.

- *Surface Manager* – mengatur akses ke subsystem tampilan dan secara halus menggabungkan layer graphik dari 2D dan 3D dari beberapa aplikasi.
- LibWebCore – sebuah mesin modem web browser yang menggerakkan Android browser dan embedded *web view.*
- SGL – mesin *graphics* 2D
- *3D libraries* – sebuah implementasi berdasarkan OpenGL ES 1.0 APIs, *library* ini menggunakan perangkat keras akselerasi 3D (jika tersedia) atau yang termasuk, perangkat lunak 3D rasterizer dengan optimasi yang tinggi.
- *FreeType* – perenderingan font bitmap dan **vector**.
- SQLite – sebuah mesin *relasional database* yang hebat dan ringan tersedia untuk semua aplikasi.

2.6.2.4.Android Runtime

Android meliputi beberapa kepustakaan inti yang menyediakan banyak fungsionalitas yang tersedia di kepustakaan inti dari bahasa pemrograman Java. Setiap aplikasi android berjalan pada prosesnya masing-masing, dengan instance mereka masing-masing pada *Dalvik virtual machine*. Dalvik telah ditulis sehingga sebuah perangkat dapat menjalankan banyak VM secara efisien. Dalvik VM mengeksekusi file dalam format *Dalvik Executable* (.dex) yang dioptimasi untuk pemakaian memory minimal. VM berbasis *register,* dan menjalankan *class* yang dikompilasi oleh sebuah kompiler bahasa Java yang telah ditransformasi menjadi format .dex dengan t*ool* yang disediakan yaitu 'dx'. Dalvik VM berada di kernel Linux untuk mendasari fungsionalitas seperti threading dan manajemen memori tingkat rendah.

2.6.2.5.Linux Kernel

Android berada di atas Linux 2.6 untuk pelayanan inti sistem seperti keamanan, manajemen memori, manajemen proses, tumpukan jaringan, dan model driver. Kernel juga bertindak sebagai layer abstraksi antara hardware dan sisa dari tumpukan software. Anatomi dari aplikasi Android Ada 4 yang mendasari dari anatomi aplikasi android, yaitu:

• *Activity*

• *Intent Receiver*

• *Service*

• *Content Provider*

Walaupun tidak semua aplikasi memiliki ke-4 blok di atas, tapi pasti dari kombinasi ke-empat blok tersebut.

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*

Service adalah aplikasi yang berjalan tanpa adanya tatap muka pengguna (user

interface). *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak. Seperti menjalankan media player yang memainkan lagu dari playlist.

2.6.3.3. *Broadcast Receiver*

Broadcast Receiver merupakan komponen yang menerima dan bereaksi untuk menyiarkan notifikasi. Misal notifikasi zona waktu telah berubah, baterai rendah. Sama halnya dengan *service, Broadcast Receivers* tidak menampilkan antarmuka pengguna. Namun, *Broadcast Receivers* dapat menggunakan *Notification Manager* untuk memberitahukan sesuatu kepada pengguna.

2.6.3.4. *Intents*

Digunakan bila ingin membuat sebuah reaksi dari event yang ada.Intent reciver ini biasanya harus di-trigger terhadap sesuatu baru akan dijalankan oleh sistem Android.

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. Content Provider diciptakan untuk berbagi data dengan Activities

lain atau Services. Sebuah Content Provider menggunakan antarmuka standar dalam bentuk URI untuk memenuhi permintaan data dari aplikasi lain.

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.

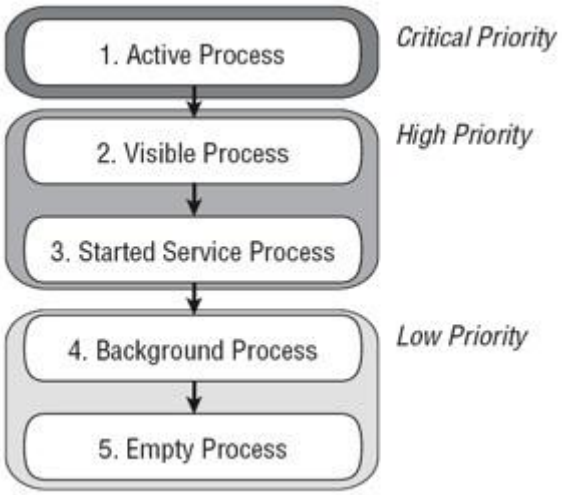
Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi aplikasi memiliki ciri-ciri dari semua kategori. Oleh karenanya

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.



Gambar 2.2 Prioritas Aplikasi Berdasarkan *Activity*

(Sumber: http://ervitakusumaputri228.wordpress.com)

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

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*.

2.7.1. 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.

2.7.2. 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 *Handphone*/*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.

2.8. 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.

2.9. 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.

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. Deployment Diagram

Diagram *deployment* atau *deployment diagram* menunjukan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan 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.