BAB II

LANDASAN TEORI

2.1. Definisi Program dan Bahasa Pemrograman

Program adalah kumpulan instruksi yang digunakan untuk mengatur komputer agar melakukan sesuatu tindakan tertentu. Suatu program ditulis dengan mengikuti kaidah bahasa pemrograman tertentu. Bahasa pemrograman merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari:

1. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110
2. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bah.Inggris *Assembly*), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode *mnemonic*), contohnya MOV, SUB, CMP, JMP, JGE, JL, LOOP, dsb.
3. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa

Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <<, >>, &&, ||, dsb.

4. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata-kata bahasa manusia, contohnya begin, end, if, for, while, and, or, dsb.

2.2. Bahasa Pemrograman JAVA

Java merupakan perangkat lunak produksi Sun Microsistem Inc. untuk pemrograman beberapa tujuan (*multi purpose*), dapat berjalan di beberapa sistem operasi (*multiplatform*), mudah dipelajari dan powerful. Aplikasi-aplikasi yang dapat dibuat dengan Java, meliputi pemrograman web (*web programming*), Pemrograman Desktop (*Desktop programming*), Pemrograman mobile/*handphone* (*mobile programming*).

Sun Microsistem sendiri mendeskripsikan Java sebagai bahasa pemrograman yang sederhana, *garbage collected*, *robust*, dan dapat diperluas. Sifat yang dideskripsikan oleh Sun Microsistem dapat dipaparkan sebagai berikut:

1. Sederhana

   Pengembang Java banyak membuang fitur-fitur yang tidak diperlukan seperti yang dimiliki bahasa pemrograman tingkat tinggi lainnya.

2. *Garbage collected* (Pembuang Sampah)

   Program Java menyingkirkan sendiri ‘sampah-sampah’ yang tidak berguna, artinya program tidak perlu menghapus objek-objek yang dialokasikannya di memori.

3. *Robust* (Tangguh)

Karena interpretasi Java memeriksa seluruh akses sistem yang dilakukan program, maka program Java tidak akan membuat sistem menjadi *crash*.

4. Dapat diperluas

   Program Java mendukung metode *native* yakni fungsi-fungsi yang ditulis dalam bahasa latin, biasanya C++.

   Java membagi versi programnya ke dalam tiga kelompok besar, yaitu :

   a. Java 2 Standar Edition (J2SE) untuk konsentrasi pada PC
   b. Java 2 Enterprise Edition (J2EE) untuk konsentrasi pada aplikasi server besar
   c. Java 2 Micro Edition (J2ME) untuk konsentrasi pada mobile

2.3. Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *Open Office.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi

dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.3.1. Klasifikasi aplikasi

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain :

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

2.4. Mobile

Mobile adalah perangkat elektronik portabel yang berfungsi sebagaimana pesawat telepon normal, yang dapat bergerak pada suatu area yang luas. (bandingkan dengan *cordless phone*).

Aplikasi bergerak merupakan istilah yang digunakan untuk suatu aplikasi yang dijalankan pada perangkat *mobile* yang cenderung memiliki ukuran kecil, *portable* dan dapat terkoneksi melalui jaringan nirkabel. Perangkat *mobile*

memiliki banyak jenis dalam hal ukuran, desain dan *layout*, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari sistem desktop.

2.4.1. Aplikasi Mobile

Aplikasi *mobile* adalah aplikasi yang melibatkan piranti bergerak dan melibatkan media komunikasi nirkabel. Oleh karena itu, peminatan ini menekuni sistem tersebar, keamanan jaringan, pemrograman jaringan, aplikasi *mobile* untuk bisnis, analisis dan perancangan jaringan, pemrograman web lanjut, sistem layanan berbasis lokasi, infrastruktur layanan *mobile*, teknologi nirkabel dan aplikasinya, sistem tertanam, serta hal-hal yang berkaitan dengan itu, termasuk di dalamnya aspek perangkat keras komputer.

2.5. Resep

Resep masakan adalah suatu susunan instruksi atau algoritma yang menunjukkan cara membuat suatu masakan.

2.5.1. Komponen Resep

Dalam resep masakan terdapat beberapa komponen yang sebagian harus ditulis dan ada juga yang tidak harus ditulis. Komponen resep yang biasanya ada ialah :

a. Nama masakan
b. Komposisi atau bahan dengan kuantitasnya
c. Alat-alat yang dibutuhkan
d. Cara memasak
e. Lama waktu memasak
f. Jumlah sajian

g. Perkiraan jumlah kalori

h. Ketahanan makanan dan penyimpanan

2.6. Dapur

Dapur adalah ruangan atau area yang dijadikan tempat pengolahan makan hingga makanan tersebut siap untuk dihidangkan. Dapur mempunyai peranan yang sangat penting dalam pengolahan makanan oleh karena itu kebersihan area dapur serta kebersihan perlengkapan dan peralatan memasak sangat diperlukan guna menghasilkan kualitas makanan yang baik.

2.6.1. Jenis-jenis Dapur

Ada dua jenis dapur yang di kenal, yaitu dapur basah dan dapur kering. Pemanfaatan kedua jenis dapur ini juga berbeda dan membutuhkan ruangan yang lebih luas.

1. Dapur Basah

Jenis dapur ini dimiliki semua rumah. Hanya saja, biasanya di rumah dengan ukuran lebih kecil dan lebih memilih letak dapur basah yang proses pencuciannya berbeda tempat unruk menghindari terlalu basah dan kotornya dapur. Ciri-ciri yang dimiliki oleh jenis dapur basah ini antara lain :

a. Jenis dapur basah memiliki ruangan yang lebih luas dan biasanya terletak di bagian belakang.

b. Dekat dengan sumber air atau terdapat tempat khusus unruk proses pencucian.

c. Terdapat peralatan memasak dan perabotan dapur yang lengkap seperti kompor dan peralatan memasak lainnya.

d. Jenis dapur basah ini berfungsi sebagai tempat pengolahan, pencucian serta proses memasak, termasuk ketika ingin menghangatkan makanan atau memmbuat hidangan lain dan minuman.
e. Lebih mudah kotor dan basah, sehingga dalam pemanfaatannya diperlukan perawatan yang lebih teratur, seperti pembersihan dan pengeringan perabotan dapur juga lantai dapur.

2. Dapur Kering

Tidak semua rumah memiliki dapur kering. Selain membutuhkan ruang tambahan untuk membuat jenis dapur kering, juga diperlukan biaya lebih saat membuat desain dan bentuk dari jenis dapur ini. Fungsi utama dapur kering ini hampir sama dengan ruang makan. Bedanya, di dalam jenis dapur kering ini terdapat perabotan ringan lainnya seperti pemanas makan atau *microwave*, kompor induksi maupun peralatan untuk membuat minuman. Dan dapur kering bukanlah sebagai tempat memproses pengolahan makanan atau memasak.

2.6.2. Fungsi Dapur

Menurut Sihite (2001:1) menyebutkan beberapa fungsi utama dapur yaitu :

1. Pusat kegiatan proses bahan baku makanan.
2. Pusat kegiatan pengolahan makanan.
3. Pusat masak memasak makanan.
4. Tempat menghasilkan resep yang baku suatu makanan.
5. Alat mengukur reputasi dan image suatu perusahaan.

2.7. 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*.

2.7.1. 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 indikator 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. Perubahanperubahan 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*)

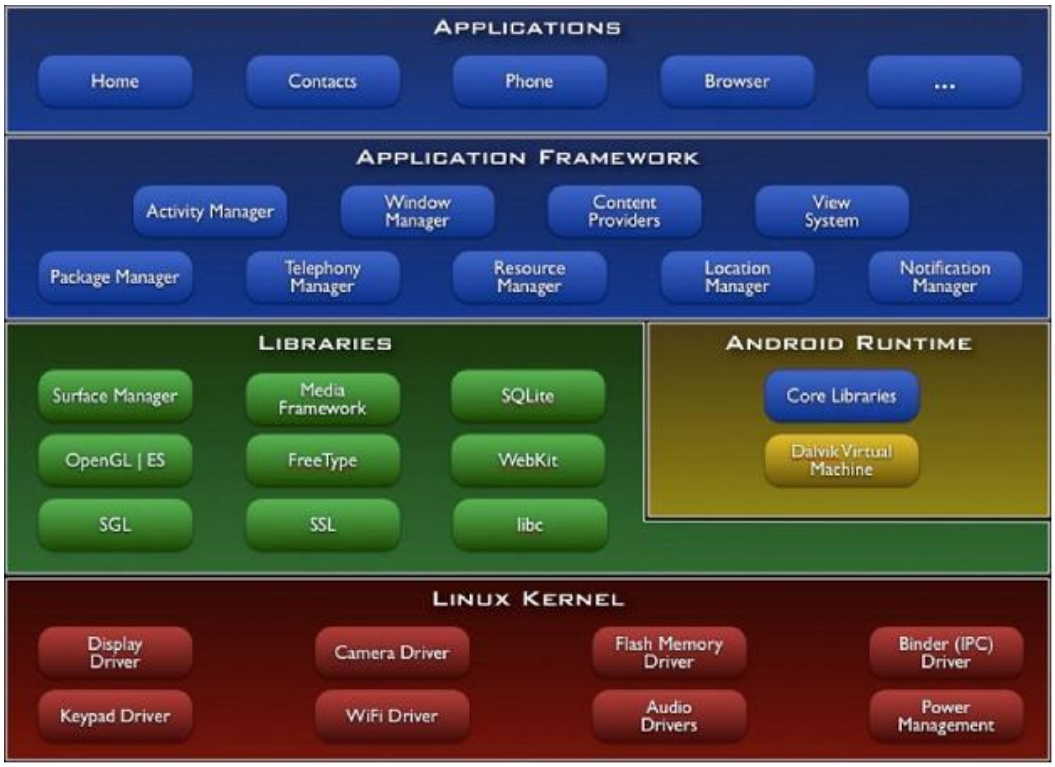
Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. *User Interfac*e pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardwar*e) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.

8. Android versi 4.0 (*ICS :Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

2.7.2. 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.



Gambar 2.1. Detail Anatomi Android
(Sumber : http://studyfuture.blogspot.com/2011/03/arsitektur-android.htm**)**

2.7.3. 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.

2.7.4. 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).*

2.7.5. 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.

2.7.6. Aplication Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik 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.

2.7.7. *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 sistem 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.

2.8. 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 sistem untuk memanggil dan menjalankan ketika dibutuhkan.

2.8.1. *Activity*

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 antar muka 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.

2.8.2. *Service*

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.

2.8.3. Intens

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.

2.8.4. *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.

2.8.5. *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.

2.9. 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.

2.9.1. 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 adanyapenanganan terhadap setiap kondisi agar aplikasi menjadi stabil.

2.9.2. 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 sistem 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 sistem 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.

2.10. *SQLite*

SQLite merupakan sebuah sistem manajemen basis data relasional yang bersifat ACID-compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. *SQLite* merupakan proyek yang bersifat public domain yang dikerjakan oleh *D. Richard Hipp*.

Tidak seperti pada paradigma client-server umumnya, Inti *SQLite* bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency times*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata

(definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

2.10.1. Fitur-Fitur Pada *SQLite*

Pustaka *SQLite* mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada *SQL-92*, termasuk transaksi yang bersifat atomic, konsistensi basisdata, isolasi, dan durabilitas (dalam bahasa inggris lebih sering disebut ACID), trigger, dan kueri-kueri yang kompleks. Tidak ada pengecekan tipe sehingga data bisa dientrikan dalam bentuk string untuk sebuah kolom bertipe integer. Beberapa kalangan melihat hal ini sebagai sebuah inovasi yang menambah nilai guna dari sebuah basisdata, utamanya ketika digunakan dalam bahasa pemrograman berbasis script (PHP, Perl), sementara kalangan lain melihat hal tersebut sebagai sebuah kekurangan.

Beberapa proses ataupun thread dapat berjalan secara bersamaan dan mengakses basisdata yang sama tanpa mengalami masalah. Hal ini disebabkan karena akes baca data dilakukan secara paralel. Sementara itu akses tulis data hanya bisa dilakukan jika tidak ada proses tulis lain yang sedang dilakukan; jika tidak, proses tulis tersebut akan gagal dan mengembalikan kode kesalahan (atau bisa juga secara otomatis akan mencobanya kembali sampai sejumlah nilai waktu yang ditentukan habis). Hanya saja ketika sebuah tabel temporer dibuat, mekanisme penguncian pada proses multithread akan menyebabkan masalah. Update yang terkini (versi 3.3.4) dikatakan telah memperbaiki masalah ini.

Sebuah program yang mandiri dinamakan sqlite disediakan dan bisa digunakan untuk mengeksekusi kueri dan memanajemen file-file basisdata *SQLite*. Program tersebut juga merupakan contoh implementasi penulisan aplikasi yang menggunakan pustaka *SQLite*.

2.10.2. Integrasi Dengan Bahasa Lain

1. *SQLite* termasuk dalam *framework REALbasic*, yang memungkinkan aplikasi yang dikembangkan dengan menggunakan *REALbasic* dapat memanfaatkan basisdata *SQLite.*
2. Pustaka *SQLite* bisa digunakan secara langsung pada bahasa C/C++, namun untuk Tcl dan beberapa bahasa pemrograman berbasis script juga tersedia.
3. Modul DBI/DBD untuk Perl juga tersedia pada CPAN, DBD:SQLite, namun modul ini bukanlah antarmuka dengan SQLite melainkan memasukkan *SQLite* secara keseluruhan dalam modul tersebut.
4. Modul *Python* juga tersedia (*PySQLite*) yang diimplementasikan pada DB API Python versi 2.0 (PEP 249)
5. PHP dimulai dengan PHP5 telah memasukkan *SQLite*, versi PHP4 sebelumnya bisa juga digunakan untuk mengakses *SQLite*, namun modul *SQLite* tidak dimasukan secara standar bawaan.
6. Dimulai Lazarus versi 0.9.8 dan Free Pascal 2.0.0, *SQLite* didukung untuk digunakan oleh para programmer. Tutorial tersebut bisa didapatkan disini.

7. Meskipun borland tidak mempaketkan SQLite secara standar bawaan, Delphi sudah mendukung SQLite juga menggunakan pustaka yang dibuat oleh pihak ketiga (*Aducom dan Zeos*). Banyak program administrasi basisdata SQLite baik komersil ataupun bebas yang menawarkan kemudahan-kemudahan dalam memanajemen basisdata SQLite terdapat di internet dibuat dengan menggunakan bahasa ini.
8. SQLite juga termasuk dalam paket yang dibundle secara standar bawaan pada Mac OS X, dan digunakan sebagai salah satu pilihan mekanisme penyimpanan data pada API Apple.

2.11. Kelebihan Dan Kekurangan Pada *SQLite*

2.11.1. Kelebihan *SQLite*

Sqlite merupakan salah satu embbeded database yang wajib dipertimbangkan untuk digunakan pada aplikasi bukan enterprise karena memiliki beberapa kelebihan yaitu :

1. Secara umum cukup stabil
2. Perfomanya lebih efesien dan cepat dibandingkan database yang lain
3. Sedikit menggunakan memory, hanya membutuhkan sigle library untuk mengakses database
4. Berjalan dibanyak platform yang berbeda, dan dapat dipindahkan dengan mudah tanpa setting administrasi yang rumit.
5. Memenuhi standart ACID (*Atomicity, Consistensy, Isolation, dan Durability*)

6. Mendukung ANSI 92 SQl Standar.
7. Mempunyai API untuk bahasa C++, PHP, Perl, Phyton, dan Tcl.
8. License public domain, free untuk redistributed.

2.11.2. Kekurangan *SQLite*

Meskipun merupakan database server yang cukup sederhana dengan beberapa fasilitas, bukan berarti SQLite tidak memiliki kekurangan. Beberapa fasilitas yang tidak didukung oleh SQLite sengaja ditiadakan untuk mempertahankan SQLite sebagai embeddable database server. Berikut beberapa fasilitas yang umum ada pada database server dan tidak dimiliki atau tidak didukung oleh SQLite.

1. *CHECK constraint*

   Fasilitas ini umumnya digunakan sebagai pemeriksa, namun ditiadakan karena masih dapat digantikan oleh NOT NULL serta UNIQUE.

2. *Variabel subquery*

   Meskipun memiliki arti yang penting, penggunaan subquery atau perintah query di dalam query ini tidak didukung.

3. *Foreign Key Constraint*

   Sama halnya dengan CHECK constraint, fasilitas ini sengaja dihilangkan oleh SQLite.

4. *Alter Table*

Oleh karena tidak memiliki fasilitas ini, maka untuk mengubah struktur tabel harus dilakukan dengan menghapus tabel kemudian membuat ulang.

5. *Grant dan Revoke*

   Fasilitas ini cukup penting untuk keamanan, namun karena SQLite melakukan pembacaan serta penulisan pada file disk, maka hak izin akses bergantung pada sistem operasi yang digunakan.

6. *Right dan Full Outer Join*

   Meskipun penggunaan right serta full outer join tidak diperkenankan, akan tetapi SQLite masih menyediakan left outer join.

7. Penulisan pada *View*

   SQLite hanya memperbolehkan penggunaan view untuk dapat dibaca saja, dengan demikian anda tidak dapat melakukan modifikasi pada view. Sebagai gantinya, SQLite menyediakan trigger yang dapat digunakan untuk memodifikasi view.

Harus diakui memang, dengan kekurangan-kekurangan yang ada, SQLite kurang begitu cocok untuk dijadikan database server pada aplikasi client/server berbasis dekstop. Mengapa? Kebanyakan aplikasi dekstop yang digunakan untuk mengelola aplikasi bisnis memerlukan fitur-fitur yang kompleks. Meski demikian, untuk keperluan aplikasi web, bisa dikatakan SQLite mampu mengganti semuanya.

2.12. Tipe Data pada *SQLite*

SQLite menggunakan sistem tipe lebih umum dinamis. Dalam SQLite, tipe data dari nilai dikaitkan dengan nilai itu sendiri, tidak dengan kontainer. Sistem tipe dinamis SQLite adalah kompatibel dengan sistem yang lebih umum jenis statis mesin database lain dalam arti bahwa pernyataan SQL yang bekerja pada database statis diketik harus bekerja dengan cara yang sama di SQLite. Namun, dinamis mengetik di SQLite memungkinkan untuk melakukan hal-hal yang tidak mungkin dalam database kaku diketik tradisional.

2.13. Penyimpanan Kelas Dan *Datatypes*

Setiap nilai yang disimpan dalam database SQLite (atau dimanipulasi oleh mesin database) memiliki salah satu kelas penyimpanan berikut:

a. NULL . Nilai adalah nilai NULL.
b. INTEGER . Nilai adalah integer ditandatangani, disimpan dalam 1, 2, 3, 4, 6, atau 8 byte, tergantung pada besarnya nilai.
c. REAL . Nilai adalah nilai floating point, disimpan sebagai nomor 8-byte IEEE floating point.
d. Nilai adalah string teks, disimpan menggunakan pengkodean database (UTF-8, UTF-16BE atau UTF-16LE).
e. BLOB . Nilai adalah gumpalan data, disimpan persis seperti itu masukan.

Perhatikan bahwa kelas penyimpanan adalah sedikit lebih umum daripada sebuah datatype. Kelas penyimpanan INTEGER, misalnya, termasuk 6 tipe data bilangan bulat yang berbeda dari panjang yang berbeda. Tapi begitu nilai-nilai INTEGER dibaca off disk dan ke dalam memori untuk pemrosesan, mereka akan dikonversi ke tipe data yang paling umum (8-byte signed integer). Dan untuk sebagian besar, "kelas penyimpanan" tidak dapat dibedakan dari "datatype" dan kedua istilah ini dapat digunakan secara bergantian.

1. Boolean Datatype

   SQLite tidak memiliki kelas penyimpanan terpisah Boolean. Sebaliknya, nilai Boolean disimpan sebagai bilangan bulat 0 (false) dan 1 (benar).

2. Date and Time Datatype

   SQLite tidak memiliki kelas penyimpanan disisihkan untuk menyimpan tanggal dan / atau waktu. Sebaliknya, built-in Date and Time Fungsi dari SQLite mampu menyimpan tanggal dan waktu sebagai TEXT, REAL, atau INTEGER:

   a. TEXT sebagai ISO8601 strings ("YYYY-MM-DD HH: MM: SS.SSS").
   b. REAL angka hari Julian, jumlah hari sejak tengah hari di Greenwich pada tanggal 24 November, 4714 SM menurut kalender Gregorian proleptic.
   c. INTEGER as Unix Time, jumlah detik sejak 1970/01/01 00:00:00 UTC.

Aplikasi dapat memilih untuk menyimpan tanggal dan waktu dalam salah satu format dan mengkonversi antara format bebas menggunakan built-in fungsi tanggal dan waktu.

2.14. *Structured Query Language*

Sebelum kita mulai bekerja dengan SQLite, kita mendefinisikan beberapa istilah penting. Sebuah query database adalah mencari informasi dari database. Sebuah permintaan yang ditulis dalam bahasa SQL. Structured Query Language (SQL) adalah bahasa komputer yang digunakan untuk membuat, mengambil, update dan menghapus data dari database. Ini dikembangkan oleh perusahaan IBM. SQL memiliki tiga subset :

1. DDL (*Data Definition Language*) adalah statement yang digunakan untuk mendefinisikan skema atau struktur dari suatu database. Terdiri dari :

    a. *CREATE* - membuat suatu objek dalam database.

    b. *ALTER* - mengubah struktur objek dari suatu database.

    c. *DROP* - menghapus objek dari suatu database.

    d. *TRUNCATE* - menghapus semua record dari sebuah tabel, termasuk semua alokasi ruang untuk record tersebut ikut dihapus.

    e. *RENAME* - mengganti nama suatu objek database.

2. DML (*Data Manipulation Language*) adalah statement yang digunakan untuk mengelola data di dalam skema atau struktur objek.

    a. *SELECT* - mendapatkan data dari suatu database.

    b. *INSERT* - memasukkan data ke dalam sebuah tabel.

    c. *UPDATE* - mengubah data yang sudah ada dalam sebuah tabel.

    d. *DELETE* - menghapus record dari sebuah tabel, alokasi ruang tidak dihapus.

3. DCL (*Data Control Language*).

    a. GRANT - memberikan hak akses pengguna ke database.

    b. REVOKE - mencabut atau menarik kembali hak akses yang telah diberikan dengan statement GRANT

2.15. SQL pada SQLite

SQLite telah memahami sebagian besar syntax SQL. Dalam tutorial kali ini pembahasan materi terbatas pada beberapa syntax SQL saja. Yaitu : *CREATE, INSERT, UPDATE, DELETE,* dan *SELECT.*

2.15.1. Eclipse

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

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

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

3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, *test* perangkat lunak, pengembangan web, dan lain sebagainya. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.

2.15.2. Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US$ 40 juta[1] untuk pengembangannya. Semenjak itu konsursium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

2.15.3. 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:

1. Core platform
2. OSGi
3. SWT (Standard Widget Toolkit)
4. Jface
5. Eclipse Workbench

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

mengembangkan plug-in baru. Eclipse beserta plug in-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya.

2.15.4. Perkembangan Eclipse

Sejak tahun 2006, Eclipse Foundation mengkoordinasikan peluncuran Eclipse secara rutin dan simultan yang dikenal dengan nama Simultaneous Release. Setiap versi peluncuran terdiri dari Eclipse Platform dan juga sejumlah proyek yang terlibat dalam proyek Eclipse. Tujuan dari sistem ini adalah untuk menyediakan distribusi Eclipse dengan fitur-fitur dan versi yang terstandarisasi. Hal ini juga dimaksudkan untuk mempermudah *deployment* dan *maintenance* untuk sistem enterprise. Adapun versi eclipse yang telah diluncurkan yaitu :

1. Eclipse 3.0

2. Eclipse 3.1

3. Callisto
4. Europa
5. Ganymede
6. Galileo
7. Helios

2.16. Software Development Kit (SDK)

Software Development Kit (SDK) adalah suatu kit atau *library* dari bahasa pemrograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya SDK terdiri dari kumpulan *tools* yang dibutuhkan. Misalnya bahasa pemrograman java, mempunyai SDK yang berisi suatu *library* yang dapat digunakan untuk membuat suatu aplikasi berbasis java.

2.17. Java Development Kit (JDK)

Java Development Kit (JDK) adalah sekumpulan perangkat lunak yang dapat kamu gunakan untuk mengembangkan perangkat lunak yang berbasis Java, Sedangkan JRE adalah sebuah implementasi dari Java Virtual Machine yang benar-benar digunakan untuk menjalankan program java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber kompiler java, bundling, debuggers, development libraries dan lain sebagainya. Perbedaan JDK dengan SDK (Software Development Kit) yaitu JDK adalah sebuah SDK tetapi sebuah SDK tidak harus menjadi sebuah JDK.

2.18. ADT

Android Development Tools (ADT) adalah plugin untuk Eclipse Intergrated Development Environment (IDE) yang dirancang untuk memberikan lingkungan yang terpadu di mana untuk membangun aplikasi Android.

ADT memperluas kemampuan Eclipse untuk membiarkan para developer lebih cepat dalam membuat proyek baru Android, membuat aplikasi UI, menambahkan komponen berdasarkan Android Framework API, debug aplikasi dalam pengunaan Android SDK, dan membuat file APK untuk mendistribusikan aplikasi.

Mengembangkan aplikasi di Eclipse dengan ADT sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat aplikasi android, karena banyak kemudahan-kemudahan sebagai tools yang terintegrasi seperti, custom XML editor, dan debug panel *ouput*. Selain itu ADT memberikan dorongan luar biasa dalam mengembangkan aplikasi Android.