BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi

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.

Menurut Ibisa**,** Aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya. Beberapa aplikasi yang digabung bersama menjadi suatu paket disebut sebagai suatu paket atau *application suite*. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi.

Software aplikasi adalah software program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu. Software aplikasi terdiri dari :

1. Bahasa pemrograman (*programming language*)
2. Program aplikasi (*Application Program*)

3. Program paket / paket aplikasi (*packet program*)
4. Program utilitas (*utility program*)
5. *Games* dan *entertainment* dan lain-lain

Untuk mendukung operasi software aplikasi di atas tugas pengguna komputer dibagi menjadi :

a. Analis system
b. Programmer
c. Operator
d. Administrator *database*
e. Administrator jaringan

2.2 Pengertian Huruf

Dalam kamus besar bahasa Indonesia, huruf didefinisikan sebagai tanda aksara dalam tata tulis yg merupakan anggota abjad yang melambangkan bunyi bahasa. Berdasarkan pengertian tersebut, dapat kita katakan kalau huruf adalah lambang dari bunyi. Misalnya bunyi **be** lambangnya atau hurufnya adalah **b**, bunyi **el** lambangnya adalah **l**, dan seterusnya.

2.2.1 Jenis-jenis Huruf

Huruf dapat diklasifikasikanberdasarkan bentuk dan bunyinya menjadi beberapa macam.

1. **Huruf berdasarkan bentuk**

   Berdasarkan bentuknya, huruf dibedakan menjadi 4 yaitu sebagai berikut.

a. **Huruf fonemis**, yaitu huruf yang melambangkan satu bunyi seperti huruf latin.

b. **Huruf silabis**, yaitu huruf yang melambangkan satu suku kata seperti huruf jepang atau aksara jawa.

c. **Huruf logograf atau idiograf**, yaitu huruf yang melambangkan bunyi satu kata seperti huruf cina.

d. **Huruf piktograf**, yaitu bunyi huruf yang dilambangkan dalam bentuk gambar atau lukisan peristiwa seperti relif manusia prasejarah.

2. **Huruf berdasarkan bunyi**

a. **Huruf vokal**, yaitu bunyi yang tidak disertai hambatan pada alat bicara, hambatan hanya terdapat pada pita suara, tidak pada artikulator. Jadi udara yang keluar dari paru-paru melewati pita suara dan tidak ada artikulator atau alat ucap yang menghambat seperti bibir, gigi, ataupun lidah. Yang termasuk bunyi vokal adalah **a, i, u, e, o, é.**

b. **Huruf konsonan**, yaitu bunyi yang dibentuk dengan menghambat arus udara yang keluar dari paru-paru. Hambatan dapat terjadi pada sebagian alat bicara, seperti hambatan pada dua bibir pada bunyi **B**, hambatan pada ujung lidah dengan menyentuh belakang gigi depan atas pada bunyi **T**, dan sebagainya.

c. **Huruf semivokal**, yaitu bunyi huruf antara vokal dan konsonan. Dalam kamus besar bahasa indonesia, semivokal didifinisikan sebagai bunyi

bahasa yangg mempunyai ciri vokal ataupun konsonan, bunyi ini mempunyai sedikit geseran dan tidak muncul sbg inti suku kata. Yang termasuk bunyi semivokal adalah bunyi **y** dan **w**.

2.3 Pengertian Angka

Angka disebut juga digit. Angka tidak sama dengan bilangan, tetapi bilangan terdiri dari angka-angka. Misal: “456” adalah lambang bilangan untuk empat ratus lima puluh enam yang terdiri dari dua angka. Arti suatu angka dalam suatu lambang bilangan ditentukan oleh nilai tempatnya dalam lambang bilangan itu. Berikut contoh nilai tempat yang membedakan nilai angka dalam bilangan basis 10.

2.4 Pengertian Mobile Aplikasi

Aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada system komputer.

Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi.

System aplikasi *Mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain

tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, telepon seluler dan PDA.

Karakteristik perangkat *mobile* :

1. Ukuran yang kecil

Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

2. Memory yang terbatas

Perangkat *mobile* juga memiliki memori yang kecil, yaitu *primary (RAM)* dan *secondary (disk).* Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan dari jumlah memori, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.

3. Daya proses yang terbatas

Sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop*.

4. Mengkonsumsi daya yang rendah

Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin *desktop*.

5. Kuat dan dapat diandalkan

Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

6. Konektivitas yang terbatas

Perangkat *mobile* memiliki *bandwith* rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi *wireless*.

7. Masa hidup yang pendek

Perangkat *mobile* memiliki *bandwidth* rendah, beberapa dari mereka bahkan tidak tersambung.

2.5 Android

Android adalah sebuah system operasi untuk perangkat *mobile* bebasis linux yang mencakup system operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. 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.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Di dunia ini terdapat 2 jenis distributor system operasi Android. Perta,a yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribbusinya tanpa dukungan langsung dari Google atau dikenal sebagai *Open Handset Distribution* (OHD).

2.5.1 Android *Platform* Masa Depan

Android dipuji sebagai "*Platform mobile* pertama yang Lengkap, Terbuka, dan Bebas".

a. Lengkap (*Complete Platform*) : Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sendang mengembangkan *platform* Android. Android merupakan system operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.
b. Terbuka (*Open Source Platform*) : *Platform* Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.
c. *Free* (*Free Platform*) : Android adalah *platform/* aplikasi yang bebas unutk develop. Tidak ada lisensi atau biaya royalty untuk dikembangkan pada *platform* Android.

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembangan sesuai yang diharapkannya. System operasi yang mendasari Android dilisensikan di bawah GNU, *General Public* Lisensi Versi 2 (GPLv2), yang sering dikenal dengan istilah "*copyleft*" lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh di bawah *terms*.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis Android. Sebagian besar pengembang menggunakan *Eclipse* yang

tersedia secara bebas untuk merancang dan mengembangkan aplikasi Android. *Eclipse* adalah IDE yang paling popular untuk pengembangan Android, karena memiliki Android *plug-in* yang tersedia untuk memfasilitasi pengembangan Android. Selain itu *Eclipse* juga mendapat dukungan langsung dari Google untuk menjadi IDE pengembang aplikasi Android, ini terbukti dengan adanya penambahan *plugins* untuk *Eclipse* utnuk membuat *project* Android di mana *source software* langsung dari situs resminya Google.

Aplikasi Android dapat dikembangkan pada sistem operasi berikut:

a. Windows XP Vista/seven.
b. Mac OS X (Mac OS X 10.4.8 atau lebih baru).
c. Linux.

Berikut sejarah perkembangan versi OS Android yang pernah dirilis :

1. **Android 1.0 beta**

   Android Beta, pertama kali di rilis tanggal 5, November, 2007, dan versi SDK (*Software developer Kit*) beta di rilis tanggal 12, November di tahun yang sama.

2. **Android 1.0**

   Android 1.0, merupakan versi *software* Android komersial pertama, dirilis 23 September, 2008. Dan perangkat Android pertama yang menggunakan dan memperkenalkan Android 1.0 adalah ponsel HTC Dream (G1).

3. **Android versi 1.5 Cup Cake**

   Android *Cup Cake* di rilis pada pertengahan Mei 2009, masih oleh Google Inc. Android ini dilengkapi *software development kit* dengan berbagai pembaharuan termasuk penambahan beberapa fitur antara lain yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube, upload gambar ke Picasa langsung dari telepon, serta mendapat dukungan *Bluetooth A2DP*.

4. **Android versi 1.6 Donut**

   Android *Donut* di rilis pada September 2009 menampilkan proses pencarian yang lebih baik dibandingkan versi-versi sebelumnya. Selain itu Android *Donut* memiliki fitur-fitur tambahan seperti galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; *Text-to-speech engine*; kemampuan dial kontak; teknologi *text to change speech.* Android Donut juga dilengkapi baterai indikator, dan kontrol *applet* VPN.

5. **Android versi 2.0/2.1 Eclair**

   Android *Eclair* dirilis pada 3 Desember 2009. Perubahan yang ada antara lain 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*. Android *Eclair* merupakan Adroid pertama yang mulai dipakai oleh banyak *smartphone*, fitur utama *Eclair* yaitu perubahan total struktur dan tampilan *user interface*.

6. Android versi 2.2 Froyo (Frozen Yogurt)

Android *Froyo* dirilis pada 20 mei 2012. Adroid versi ini memiliki kecepatan kinerja dan aplikasi 2 sampai 5 kali dari versi-versi sebelumnya. Selain itu ada penambahan fitur-fitur baru seperti dukungan *Adobe Flash* 10.1, 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.

7. Android versi 2.3 Gingerbread

Android *Gingerbread* di rilis pada 6 Desember 2010. 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.

8. Android versi 3.0/3.1 Honeycomb

Android *Honeycomb* di rilis pada awal 2012. Merupakan versi Android yang dirancang khusus untuk device dengan layar besar seperti Tablet PC. Fitur baru yang ada pada Android *Honeycomb* antara lain yaitu dukungan terhadap prosessor *multicore* dan grafis dengan *hardware acceleration*. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. Tablet pertama yang memakai *Honeycomb* adalah tablet Motorola

Xoom yang dirilis bulan Februari 2011. Selain itu sebuah perangkat keras produksi Asus bernama *Eee Pad* Transformer juga menggunakan OS Android *honeycomb* dan diharapkan akan masuk ke pasaran Indonesia pada Mei 2011.

9. Android versi 4.0 ICS (Ice Cream Sandwich)

Android *Ice Cream Sandwich* diumumkan secara resmi pada 10 Mei 2011 di ajang Google I/O *Developer Conference* (San Francisco), pihak Google mengklaim Android *Ice Cream Sandwich* akan dapat digunakan baik di *smartphone* ataupun tablet. Android *Ice Cream Sandwich* membawa fitur *Honeycomb* untuk *smartphone* serta ada penambahan fitur baru seperti 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. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

10. Android versi 4.1 Jelly Bean

Android *Jelly Bean* juga diluncurkan pada acara Google I/O 10 Mei 2011 yang lalu. Android versi ini membawa sejumlah keunggulan dan fitur baru, diantaranya peningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui *Voice Search* yang lebih cepat. Versi ini juga dilengkapi Google Now yang dapat memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas,

ataupun hasil pertandingan olahraga. Sistem operasi Android *Jelly Bean* 4.1 pertama kali digunakan dalam produk tablet Asus, yakni Google Nexus 7.

11. Android versi 4.4 Kitkat

Beberapa waktu yang lalu Google secara resmi memperkenalkan sistem operasi terbarunya, yaitu update Sistem Operasi Android versi 4.4 yang diberi nama *KitKat*. Kabar tersebut cukup mengagetkan banyak pihak terlebih pengambilan nama untuk versi terbaru Android tersebut ternyata diluar perkiraan. Sebelumnya banyak kabar yang berhembus bahwa update sistem operasi Android terbaru yang akan diusung oleh Google akan dinamai Android *Key Lime Pie*, namun ternyata Google menepis semua rumor tersebut dengan memperkenalkan Android *KitKat*.

2.5.2 *The Dalvic Virtual Machine* (DVM)

Salah satu elemen kunci dari Android adalah *Dalvic Virtual Machine* (DVM). Android berjalan di dalam *Dalvic Virtual Machine* (DVM) bukan di *Java Virtual Machine* (JVM), sebenarnya banyak persamaannya dengan *Java Virtual Machine* (JVM) seperti Java ME (*Java Mobile Edition*), tetapi Android menggunakan *Virtual Machine* sendiri yang dirancang untuk memastikan bahwab beberapa *feature-feature* berjalan lebih efisien pada perangkat *mobile.*

2.5.3 Android SDK (*Software Development Kit*)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperllukan untuk mulai mengembangkan aplikasi pada *platform* Android

menggunakan bahasa pemograman Java. Sebagai *platform* aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *handphone/smartphone*. Beberapa fitur Android yang paling penting adalah:

1. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable.*
2. Mesin *Virtual Davic* dioptimalkan untuk perangkat *mobile.*
3. *Integrated Browser* berdasarkan *engine open source WebKit.*
4. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional Akselerasi *hardware*).
5. SQLite untuk penyimpanan data.
6. Media Support yang mendukung audio, video, dan gambar.
7. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
8. Kamera, GPS, kompas, dan acc elerometer (tergantung hardware)
9. Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

2.5.4 ADT (*Android Development Tools*)

Android Development Tools (ADT) adalah plugin yang didesain untuk IDE *Ecplise* yang memberikan kita dalam mengembangkan aplikasi Android dengan menggunakan ADT untuk *Ecplise* akan memudahkan kita dalam membuat

aplikasi *project* Android. Dengan ADT juga kita dapat melakukan pembuatan *package* android (.apk) yang digunakan untuk distribusi aplikasi Android yang kita rancang.

Mengembangkan aplikasi Android dengan menggunakan ADT di *Eclipse* sangat di anjurkan dan sangat mudah untuk memulai mengembangkan aplikasi Android. Berikut adalah versi ADT untuk *Eclipse* yang sudah dirilis:

1. ADT 12.0.0 (July 2011)
2. ADT 11.0.0 (June 2011)
3. ADT 10.0.1 (March 2011)
4. ADT 10.0.0 (February 2011)
5. ADT 9.0.0 (January 2011)
6. ADT 8.0.1 (December 2010)
7. ADT 8.0.0 (December 2010)
8. ADT 0.9.9 (September 2010)
9. ADT 0.9.8 (September 2010)
10. ADT 0.9.7 (May 2010)
11. ADT 0.9.6 (March 2010)
12. ADT 0.9.5 (December 2010)
13. ADT 0.9.4 (October 2009)

2.5.5 Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

1. *Application* dan *widget*

Application dan *widget* ini adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut.

2. *Applications Frameworks*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur *alarm*, dan menambahkan status *notification.*

3. *Libraries*

Libraries ini adalah *layer* di mana fitu-fitur Android berada, biasanya para pembuat aplikasi mengakses *Libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, *layer* ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL

4. *Android Run Time*

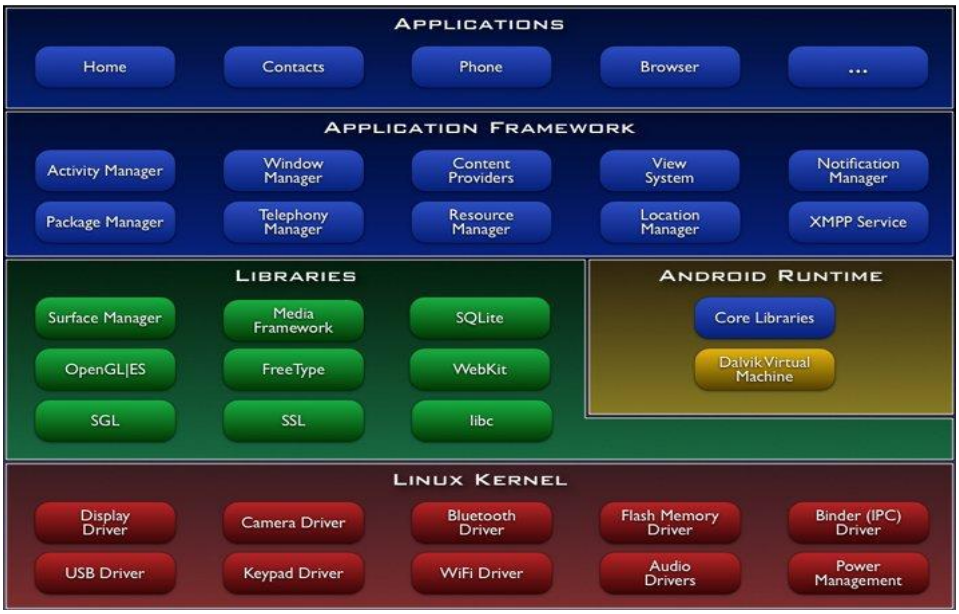
Layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakaan Implementasi Linux. *Dalvic Virtual Machine* (DVM)

merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android *Run Time* dibagi menjadi dua bagian yaitu:

a. *Core Libraries:* Aplikasi Android dibangun dalam bahasa Java, sementara Dalvic sebagai virtual mesinnya bukan *Virtual Machine Java,* sehingga diperlukan sebuah *libraries* yang berfungsi untuk memterjemahkan bahasa Java/C yang ditangani oleh *Core Libraries*.
b. *Dalvic Virtual Machine:* Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, di mana merupakan pengembang yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

5. Linux *Kernel*

Linux *Kernel* adalah *layer* di mana inti dari operating system dari Android itu berada. Berisi *file-file system* yang mengatur *system processing, memory, resource, drivers*, dan system-sistem operasi Android lainnya. Linux *Kernel* yang digunakan Android adalah Linux *Kernel release* 2.6.



Gambar 2.1 Arsitektur Android

2.5.6 Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemograman Java. Kode Java dikompilasikan bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, di mana prosesnya dipackage oleh *tools* yang dinamakan "*apt tools*" ke dalam paket Android sehingga menghasilkan *file* dengan ekstensi apk. *File* apk itulah yang kita sebut dengan aplikasi, dan nantinya dapat di install di perangkat *mobile.*

Ada enam komponen pada aplikasi Android yaitu:

a. *Activities*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi Android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi

tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain kita dapat melakukannya dengan satu *even*, misalnya *click* tombol, memilih opsi atau menggunakan triggers tertentu. Secarfa hirarki sebuah *windows activity* dinyatakan dengan *method* Activity.setContentView(). ContentView adalah objek yang berada pada *root* hirarki.

b. *Service*

Suatu *service* tidak memiliki *Graphic User Interface* (GUI), melainkan berjalan di *background* untuk waktu yang tidak terbatas. Misalnya memainkan music, *service* mungkin memainkan music atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. *Service* dijalankan pada Thread utama dari proses aplikasi.

c. *Broadcast Reciever*

Broadcast Reciever berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *Broadcast* seperti notifikasi zona waktu berubah, baterai low, gambar telah selesai diambil oleh *camera*, atau pengubahan referensi bahasa yang digunakan. *Broadcast Reciever* tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau mungkin menggunakan *Notification Manager* untuk memberi tahu kepada pengguna.

d. *Content Provider*

Content Provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file system seperti database SQLite. *Content Provider* menyediakan cara mengakses data yang dibutuhkan oleh suatu activity. Misalnya ketika kita membutuhkan peta (Map), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka di sinilah fungsi *Content Provider.*

2.6 Eclipse

Eclipse adalah sebuah IDE (*Intergrated 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. *Multi-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.

2.7 Java

Bahasa Java merupakan bahasa akhir abad 20, diciptakan *Sun Microsystem*. Pada 1991, sekelompok insinyur Su dipimpin Patrick Naughton dan James Gosling ingin merancang bahasa komputer untuk perangkat consumer seperti *cable TV Box*. Karena perangkat itu tidak mempunyai banyak memori, bahasa harus berukuran kecil dan menghasilkan kode yang liat. Juga karena manufaktur berbeda memilih pemroses berbeda, maka bahasa harus bebas dari arsitektur manapun. Proyek diberi nama kode "*GREEN*".

Bahasa Java adalah bahasa *modern* yang telah diterima masyarakat komputasi dunia. Pemilihan bahasa Java karena aplikasi tidak mensyaratkan *platform* tertentu sehingga aplikasi akan dapat dijalankan di *platform* manapun yang sangat beragam termasuk *platform* Microsoft.

Bahasa Java dirancang dengan mengumpulkan kemampuan-kemampuan penting bahasa-bahasa pemograman sebelumnya. Bahasa Java memberi harapan untuk menjadi perekat universal yang dapat mengkoneksi pemakai dengan informasi dengan informasi yang datang dari *web server*, basis data, penyedia informasi dan sumber-sumber lain, bahkan untuk pengembang program/aplikasi waktu nyata. Bahasa Java memiliki fitur keamanan *built in* yang membuat tugas

pemograman lanjut seperti pemograman jaringan, pemograman tersebar, konektifitas basis data. Juga bahasa Java memiliki fitur *multithreading* sehingga mempermudah dalam pemograman aplikasi *multithreading*, pemograman menjadi lebih nyaman.

Kebutuhan untuk kecil, liat dan kode netral terhadap platform mengantar tim mempelajari implementasi Pascal yang pernah dicoba. Niklaus Wirth, pencipta bahasa Pascal telah merancang bahasa portable yang menghasilkan kode antara (*intermediate code*) untuk mesin hipotesis. Mesin ini sering disebut mesin maya (*virtual machine*). Kode antara ini kemudian dapat digunakan di sembarang mesin yang memiliki interpreter. Proyek *Green* menggunakan mesin maya untuk mengatasi isu utama netral terhadap arsitektur mesin.

Karena orang-orang di proyek *Green* berbasis C++, maka kebanyakan sintaks diambil dari C++, serta mengadopsi orientasi objek bukan procedural. Mulanya bahasa yang diciptakan diberi nama "Oak", kemudian diganti "Java" karena telah ada bahasa pemograman yang bernama "Oak".

Produk pertama proyek "Green" adalah "*7", sebuah kendali jauh yang sangat cerdas. Bahasa Java merupakan karya *Sun Microsystem Inc*. rilis resmi level beta dilakukan pada November 1995. Dua bulan berikutnya *Netscape* menjadi perusahaan pertama yang memperoleh lisensi bahasa Java dari *Sun*. Pada 1995, Netscape memutuskan membuat *browser* yang dilengkapi dengan Java. Setelah itu diikuti oleh IBM, Symantec, Insprise, bahkan Microsoft.

Pada tahun 1996 Sun mengeluarkan JSDK (*Java Software Development Kit*) kemudian secara berturut-turut:

1. Versi 1.02 yang mendukung konektivitas basisdata dan objek-objek tersebar.
2. Versi 1.1 pada 1997 menambahkan *event model* yang handal, *internationalization*, dan model komponen *JavaBeans*.
3. Versi 1.2 dirilis pada 1998 mempunyai banyak peningkatan, diantaranya *user interface, toolkit "Swing"* yang memungkinkan pemograman membuat aplikasi berbasis GUI yang sepenuhnya *portable*.
4. Versi 1.3 dirilis pada 2000, dengan banyak peningkatan.
5. Versi 1.4 menambahkan fasilitas asersi untuk dukungan *design-by-contact.*

Bahasa Java tidak lagi hanya untuk pemanis di web sebagai applet yang membuat Duke berdansa. Java adalah kakas, tetap hanya perangkat, bagaimanapun tetap hanya orang hebat yang dapat memberi arti penting kakas itu seperti dikatakan James Gosling, tokoh terpenting di Java. "*All along, the language was a tool, not the end*".

2.7.1 Edisi Java

Kebanyakan bahasa pemograman *modern* berdiri di atas pustaka-pustaka kelas yang telah ada untuk mendukung fungsionalitas bahasanya. Pada bahasa Java, kelompok-kelompok kelas yang berkaitan erat dimasukkan dalam satu paket, bervariasi sesuai edisi Java. Masing-masing paket untuk maksud tertentu: *aplet*, aplikasi standar, skala *enterprise*, dan produk konsumer.

Pada pengembangan *enterprise applications*, kita menggunakan sejumlah besar paket. Pada *consumer electronic product*, hanya sejumlah kecil bagian bahasa yang perlu digunakan. Masing-masing edisi berisi *Java 2 software*

development Kit (JSDK) untuk mengembangkan aplikasi dan *Java 2 Runtime Environment* (JRE) untuk menjalankan aplikasi.

2.7.2 Standard Edition (J2SE)

The Java 2 Platform, Standard Edition (J2SE) menyediakan lingkungan pengembangan yang kaya fitur, stabil, aman, *cross-platform*. Edisi ini mendukung fitur konektivitas, basis data, rancangan antarmuka pemakai, input/output, dan pemograman jaringan dan termasuk sebagai paket-paket dasar bahasa Java.

2.7.3 Enterprise Edition

The Java 2 Enterprise Edition (J2EE) menyediakan lingkungan pengembangan dan menjalankan *multitier enterprise applications*. J2EE berisi paket-paket di J2SE ditambah paket-paket utnuk mendukung pengembangan *Enterprise JavaBeans, Java Servlets, aJava Server Pages, XML,* dan kendali transaksi yang fleksibel.

2.7.4 Micro Edition (J2ME)

Java2 Micro Edition atau yang biasa disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer *desktop* yang biasanya lebih kecil dibandingkan perangkat komputer *desktop.* J2ME biasa digunakan pada telepon selular, pager, personal digital assistants (PDA) dan sejenisnya.

2.8 Kelebihan Java

Java memiliki kelebihan disbanding bahasa pemograman yang lainnya. Beberapa diantaranya adalah :

1. Bahasa Sederhana

   Java dirancang agar mudah dipelajari dan digunakan secara efektif. Java tidak menyediakan fitur-fitur rumit bahasa pemograman level tingkat tinggi, serta banyak pekerjaan pemograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan Java secara otomatis seperti dealokasi memori.

2. Bahasa Beorientasi Objek

   Java merupakan penyeimbang antara mazhab berorientasi objek "murni" yang memandang "semua harus objek" dan mahzab pragmatis yang menerapkan model pragmatis "*stay out my way*". Model objek di Java adalah sederhana dan mudah diperluas.

3. Bahasa yang Aman

   Java merupakan bahasa yang dirancang telah diperhitungkan keamanan system. Saat ini Java telah menerapkan keamanan yang ketat namun fleksibel. Keamanan berdasar suatu file kebijakan sehingga dapat diatur dan dikendalikan untuk memperoleh kemampuan maksimal bagi program mandiri dan applet yang dipercaya (*trusted code*).

4. Bahasa *Multithreading*

   Thread adalah untuk menyatakan program computer melakukan lebih dari satu tugas di satu waktu yang sama. Java menyediakan kakas untuk menulis program *multithreaded.*

5. Bahasa yang Didukung *Garbage Collector*

   Program Java melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek-objek yang tidak diguakan lagi.

6. Bahasa Independen terhadap *Platform*

   Platform independence adalah kemampuan program bekerja di system operasi atau system computer berbeda. Bahasa Java merupakan bahasa yang secara sempurna tidak bergantung *platform.*

2.9 UML

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan, standar. (Chonoles, 2003:bab 1) menyatakan sebagai bahasa, berarti UML memiliki syntaks dan semantik. Ketika kita membuat model menggunakan konsep UML, ada aturan-aturan yang harus diikuti. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangjat lunak dengan proses bisnisnya.
3. Menjabarkan system secara rinci untuk analisa dan mencari apa yang diperlukan system.
4. Mendokumentasi system yang ada, proses-proses dan organisasinya.

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis timing diagram) dan lainnya ada yang bersifat umum (misalnya diagram kelas). UML memungkinkan para anggota team untuk bekerja sama

dengan bahasa model yang sama dalam mengaplikasikan beragam system. Intinya, UML merupakan alat komunikasi yang konsisten dalam mensuport para pengembang system saat ini. Sebagai perancang system, mau tidak mau pastimenjumpai UML, baik kita sendiri yang membuat atau sekedar membaca diagram UML buatan orang lain (Pilone, 2005:bab 1).

2.9.1 Use Case Diagram

Menurut (pilone, 2005 : bab 7.1) *use case* menggambarkan fungsi tertentu dalam suatu system berupa komponen, kejadian atau kelas. Sedangkan (Whitten, 2004: 258) mengartikan *use case* sebagai urutan, langkah-langkah yang secara tindakan saling terkait (*scenario*), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk ellips/oval.

2.9.2 Class Diagram

Class Diagram adalah inti dari proses pemodelan objek. Baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini. *Forward engineering* adalah proses perubahan model menjadi kode program, sedangkan *reverse engineering* sebaliknya mengubah kode program menjadi model.

2.9.3 Package Diagram

Package diagram bersifat statis, diagram ini memperlihat kan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

2.9.4 Sequence Diagram

Bersifat dinamis. *Sequence diagram* adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

2.9.5 Communication Diagram

Diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.

2.9.6 Activity Diagram

Activity diagram adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu system. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu system dan memberi tekanan pada aliran kendali antar objek.

2.9.7 Component Diagram

Component diagram ini memperlihatkan organisasi serta ketergantungan system/ perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

2.9.8 Deployment Diagram

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. Diagram deployment berhubungan erat dengan diagram komponen, dimana diagram ini memuat satu atau lebuh komponen-komponen.

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 SQLite Database Server

Pada prinsipnya SQLite merupakan library C yang diimplementasikan secara *embeddable* (tertanam) sebagai SQL *database engine*. Hal ini akan memungkinkan program yang berhubungan dengan SQLite dapat mengakses *database* tanpa menjalankan proses RDBMS secara terpisah, sebagaimana pada umumnya ketika anda menggunakan MySQL Server, PostgreSQL, Oracle, dan sebagainya.

Kemasan praktis inilah yang memberikan banyak sekali keuntungan, di mana anda tidak perlu melakukan manajemen *database* server terpisah. Selain itu,

ukuran kecil yang sengaja didesain pada SQLite memungkinkan kita untuk membuat aplikasi yang ringan dengan kinerja tinggi.

2.10.3 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*.

2.10.4 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 table 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.