BAB II

LANDASAN TEORI

2.1. Pengertian Aplikasi

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 agar data diolah lebih berdaya guna secara optimal. Jogiyanto (2004:4)

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.

Aplikasi adalah *software* atau perangkat lunak yang dibuat untuk mengerjakan menyelesaikan masalah-masalah khusus. Daryanto (2004:347)

Dari uraian diatas dapat disimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang berisi perintah untuk menyelesaikan masalah dan pengolahan data.

2.2. Pengertian Pemesanan

Pesan adalah kata baku dari pemesanan yang memiliki arti “hendak membeli supaya dikirim”. Pesanan adalah barang yang dipesan. Jadi pemesanan adalah proses, perbuatan atau cara memesan.(sumber : Kamus Bahasa Indonesia)

Pemesanan merupakan proses memilih dan menentukan barang atau jasa yang akan dibeli sebelum dilakukan persetujuan penjualan. Pemesanan dilakukan

oleh konsumen. Adanya pemesanan membantu penjual menentukan jumlah produksi. Kemajuan teknologi sekarang ini memungkinkan pemesanan tidak hanya dapat dilakukan jika konsumen dan penjual bertatap langsung. Tetapi dapat dilakukan dari jarak yang jauh sekalipun, misalnya dengan menggunakan *e-commerce*, yaitu sistem jual – beli melalui internet.

2.3. Pengertian Tiket

Tiket merupakan suatu alat/media yang digunakan oleh perusahaan tertentu sebagai pengganti uang langsung. Tiket biasanya berupa kertas yang didalamnya terdapat item – item tertentu yang menunjukan suatu nilai

Tiket pertandingan sepak bola dapat diperoleh dengan cara melakukan pemesanan sebelum pertandingan berlangsung atau dengan cara membeli langsung pada saat waktu pertandingan dilaksanankan.

2.3. Pengertian *Mobile*

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.

Pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman konensional pada *PC*. Aspek karakteristik dari perangkat bergerak sering memperngaruhi arsitektur dan implementasi dari aplikasi tersebut. Dalam pemrograman aplikasi bergerak berbagai aspek teknis perangkat lebih menonjol karena memiliki banyak keterbatasan dibandingkan komputer konvensional atau *PC*. Budi Darytamo dkk (2007:1)

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*. Ukuran teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti *harddisk* dan *RAM*, dapat ditemukan dalam ukuran yang pas dengan sebuah kemasan yang kecil.

4. Mengkonsumsi daya yang rendah

   Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin *desktop*. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai.

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.

2.4. Teknologi Java

2.4.1. Sejarah Java

Bahasa pemrograman Java pertama lahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oeh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot Duke yang dibuat oleh Joe Palrang.

Pertemuan proyek berlangsung di sebuah gedung perkantoran Sand Hill Road di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program Java Oak pertama, yang ditujukan sebagai

pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada *PDA* sekarang ini. Teknologi baru ini dinamai "*7" (*Star Seven*).

Setelah era *Star Seven* selesai, sebuah anak perusahaan tv kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto.Perusahaan baru ini bertambah maju, jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian internet sebagai medium yang menjembatani kerja dan ide diantara mereka. Pada awal tahun 1990-an, internet masih merupakan rintisan, yang dipakai hanya dikalangan akademisi dan militer. Mereka menjadikan perambah (*browser*) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai *Web Runner*, terinsipirasi dari film 1980-an, *Blade Runner*. Pada perkembangan rilis pertama, *Web Runner* berganti nama menjadi Hot Java.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar San Jose Mercury News pada tanggal 23 Mei 1995. Sayang terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace. Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape. Nama Oak, di ambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak di pakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi

"Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau Java. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Java (bahasa Inggris untuk Java adalah Java).

2.4.2. Bahasa Pemrograman Java

Java adalah suatu platform teknologi yang dikembangkan oleh Sun Microsystem sekitar tahun 90an. Java bukan sekedar bahasa pemrograman, tetapi merupakan suatu sistem *platform* yang disediakan oleh Java untuk membangun suatu sistem baik dari yang berskala kecil (*game, workstation* program, *enterprise* sampai ke *mobile device*) dapat dibangun dengan *platform* yang disediakan oleh Sun tersebut. Kadir (2004:76)

Java sebagai suatu platform dapat dibagi menjadi :

a) Bahasa pemrograman Java.

   Bahasa pemrograman Java adalah suatu bahasa yang murni *Object Oriented Programming*. Semua kriteria *OOP* terhadap di dalam Java antara lain :

   - Abstraksi data dan enkapsulasi (*public, private, protected*).
   - *Inheritance*.
   - *Polimorfism*.

b) JVM (Java *Virtual Machine*).

   Dengan JVM ini maka semboyan Java yaitu "*write once run everywhere*" dapat direalisasikan dimana dengan JVM ini suatu program tidak lagi tergantung terhadap *platform OS* yang digunakan dan berinteraksi dengan *OS*. Kompilasi terhadap suatu file Java (*Source*) akan menghasilkan suatu file *byte code* (extention class) dimana *byte code* java adalah sama untuk semua *platform*, sehingga ketika menjalankan sebuah program Java. JVM

yang akan menangani segala sesuatu yang berhubungan dengan OS dan menjalankan *byte code* yang telah dihasilkan.

c) Java basic API (J2SDK)

Java Basic API adalah sekumpulan *class* yang disediakan oleh java untuk mempermudah melakukan proses pengembangan terhadap aplikasi java. JDK (Java Development Kit) dapat diunduh dari website Sun.

Sun membagi arsitektur Java membagi tiga bagian, yaitu:

1. J2EE (Java 2 *Enterprise Edition*)
2. J2SE (Java 2 *Second Edition*)
3. J2ME (Java 2 *Micro Edition*)

2.4.3. *J2EE (Java 2 Enterprise Edition)*

J2EE adalah kelompok dari beberapa API (*Application Programming Interface*) dari Java dan teknologi selain Java. *J2EE* dirancang untuk membuat aplikasi yang rumit. *J2EE* sering dianggap sebagai *middle-ware* atau teknologi yang berjalan di *server*, namun sebenarnya *J2EE* tidak hanya terbatas untuk itu. Faktanya *J2EE* juga mencakup teknologi yang dapat digunakan disemua lapisan dari sebuah sistem informasi. Implementasi *J2EE* menyediakan kelas dasar dan *API* dari Java yang mendukung pengembangan dan rutin standard untuk aplikasi *client* maupun *server*, termasuk aplikasi yang berjalan di *web browser*. untuk aplikasi berbasis *web*, aplikasi sistem tersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi.

2.4.4. *J2SE (Java 2 Standard Edition)*

J2SE atau Java 2 *Standard Edition* merupakan bahasa pemrograman Java untuk aplikasi *desktop* yang merupakan *object-oriented programming*. Pada *J2SE*, terdiri dari dua buah produk yang dikeluarkan untuk membantu dalam membuat aplikasi tanpa tergantung dari *platform* yang digunakan, yaitu:

1. Java *SE Runtime Environment* (*JRE*)

   Java *Runtime Environment (JRE)* menyediakan perpustakaan, Java *Virtual Machine (JVM)*, dan komponen lain untuk menjalankan *applet* dan aplikasi yang ditulis dengan bahasa pemrograman Java. Selain itu, terdapat dua buah kunci teknologi yang merupakan bagian dari *JRE*, yaitu: *Plug-in*, yang memungkinkan menjalankan *applet* di *browser* populer dan Java *Web Start*, yang menyebarkan aplikasi mandiri melalui jaringan. *JRE* tidak mengandung utilitas seperti *compiler* atau *debugger* untuk mengembangkan *applet* dan aplikasi.

2. Java *Development Kit (JDK)*

   Java *Development Kit (JDK)* merupakan perangkat lunak yang digunakan untuk menajemen dan membangun berbagai aplikasi Java. *JDK* merupakan superset dari *JRE*, berisikan segala sesuatu yang ada di *JRE* ditambahkan *compiler* dan *debugger* yang diperlukan untuk mengembangkan *applet* dan aplikasi. Bahasa pemrograman Java menyediakan *library-library* standar yang telah *dicompile* dan dapat langsung digunakan dalam implementasi pembuatan sebuah aplikasi. Pada

library, terdapat berbagai macam *class* yang dapat digunakan dan telah dikelompokkan ke dalam *package*.

2.4.5. *J2ME (Java 2 Micro Editon)*

J2ME adalah lingkungan pengembangan yang dirancang untuk meletakan 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. *J2ME* adalah bagian dari *J2SE*, karena itu tidak semua l*ibrabry* yang ada pada *J2SE* dapat digunakan pada *J2ME*.

2.4.6. Kelebihan Pemrograman Java

Java memiliki beberapa kelebihan dibandingkan dengan bahasa pemrograman lainnya. Adapun kelebihan tersebut di antaranya:

1. Sederhana dan Ampuh

   Java dirancang untuk mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++ akan mudah sekali untuk berpindah ke Java. Pemakai dapat belajar membuat program dengan Java secara cepat jika telah memahami konsep dasar pemrograman berorientasi objek. Java member programmer kemampuan untuk menuangkan semua ide, karena bahasa pemrograman ini bukan merupakan *scripting language* (bahasa

naskah) yang menghilangkan kemampuan programmer untuk berinovasi, tetapi dengan cara berorientasi objek yang mudah dan jelas.

2. Aman

   Java dirancang sebagai bahasa pemrograman yang handal dan aman. Aplikasiaplikasi yang dibangun dengan bahasa Java sangat handal dengan manajemen memori yang bagus. Aplikasi Java juga dikenal sangat *secure*, yaitu kasus - kasus seperti *buffer everflow* yang umumnya menjadi lubang keamanan aplikasi-aplikasi berbasis C/C++ tidak terjadi di Java, karena pengaturan keamanannya yang bagus.

3. Berorientasi-Objek

   Java merupakan bahasa pemrograman berorientasi objek dan bukan turunan langsung dari bahasa pemrograman manapun, juga sama sekali tidak kompetibel dengan semuanya. Java memiliki keseimbangan, menyediakan mekanisme peng-*class*-an sederhana, dengan model antar muka dinamik yang intuitif hanya jika diperlukan.

4. Kokoh

   Kesalahan sering terjadi pada saat kompilasi karena Java merupakan bahasa pemrograman yang sensitif dalam hal deklarasi dan tipe data. Oleh karena hal tersebut, Java memiliki batasan agar dapat menemukan kesalahan lebih cepat saat mengembangkan program yaitu dengan langsung memeriksa program saat ditulis, dan sekali lagi ketika program di jalankan. Hal ini akan lebih menghemat waktu jika dibandingkan dengan keharusan menjalankan program terlebih dahulu dan memeriksa

semua bagian program untuk melihat ketidak cocokan dinamis selama program berjalan.

5. Interaktif

   Java memiliki kemampuan yang memungkinkan program melakukan beberapa hal pada saat bersamaan, tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya. Jalinan program-program Java yang mudah digunakan memungkinkan *programmer* untuk memikirkan pembuatan perilaku khusus, tanpa harus mengintegrasikan perilaku tersebut dengan model pemrograman global yang mengatur perulangan kejadian.

6. Netral Terhadap Berbagai Arsitektur

   Java memiliki kemampuan yang memungkinkan program melakukan beberapa hal pada saat bersamaan, tanpa harus kesulitan menangani proses yang akan terjadi selanjutnya. Jalinan program-program Java yang mudah digunakan memungkinkan *programmer* untuk memikirkan pembuatan perilaku khusus, tanpa harus mengintegrasikan perilaku tersebut dengan model pemrograman global yang mengatur perulangan kejadian.

7. Terinterpretasi dan Berkinerja-Tinggi

   Java dirancang untuk tetap berkinerja baik pada *CPU* yang tidak terlalu kuat. Walaupun Java merupakan bahasa terinterpretasi, kode-kode Java telah dirancang dengan hati-hati sehingga mudah diterjemahkan ke dalam bahasa asli suatu mesin untuk menghasilkan kinerja yang tinggi. Java dilengkapi keajaiban lintas-*platform* yang luar biasa dengan kompilasi ke dalam representasi langsung yang disebut kode-byte Java *(Java byte-*

code), yang dapat diterjemahkan oleh sistem manapun yang memilki program Java didalamnya.

8. Multiplatform

Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform*/ sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin/*bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis Java dikerjakan diatas operating sistem Linux tetapi dijalankan dengan baik di atas Microsoft Windows. *Platform* yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk menginterpretasikan *bytecode* tersebut.

9. Perpustakaan kelas yang lengkap

Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman Java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

10. Bergaya C++

    Dalam bahasa pemrograman manapun selalu ada fitur komentar. Java sendiri memiliki sintaks seperti bahasa pemrograman C++, dengan tambahan yang sangat istimewa yaitu komentar yang nantinya digunakan untuk menciptakan dokumentasi secara otomatis oleh suatu aplikasi yang disebut javadoc, sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.

11. Pengumpulan sampah otomatis

    Memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas). Serupa dengan C, java mengalokasikan operator baru untuk mengalokasikan memori pada heap untuk objek baru.

2.4.7. Kekurangan Pemrograman Java

Adapun kekurangan pemrograman Java, yaitu:

1. Tulis sekali, perbaiki di mana saja

   Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk *J2SE*, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.

2. Mudah didekompilasi

   Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/*direverse-engineer*.

3. Penggunaan memori yang banyak

   Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.5. Pengertian Android

The basic architecture of Android is shown in Figure 1. At its core, Android OS builds are based on the Linux 2.6 kernel. When running on a hard drive, the Linux system device defaults to the first physical hard drive, or /dev/hd0. In addition, Linux only understands character and block devices, such as keyboards and disk drives, respectively. With Linux on flash, however, a Flash Transition layer provides the system evice functionality. A Memory Technology Device (MTD) is needed to providean interface between the Linux OS and the physical flash device because flash memory devices are not seen as character or block devices. Dedekind (2009)

Android adalah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* yang 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.

The security mechanisms of the Android OS could impede a forensic examinational though some of the basic tools and techniques could allow investigators to recover data from the device. The first, most obvious step is to perform a traditional forensics analysis of the microSD card from the phone. This is the least effective method as it can only is access the data that apps directly store on the SD card. SD cards use the FAT32 file system and are easily imaged and examined using traditional forensics tools (including write-blocking hardware) .TalkForensics (2009)

Keunggulan utama Android adalah gratis dan *open source*, yang membuat *smartphone* Android dijual lebih murah dibandingkan dengan Blackberry atau iPhone meski fitur (*hardware*) yang ditawarkan Android lebih baik. Beberapa fitur utama dari Android antara lain *WiFi hotspot*, *Multi-touch*, *Multitasking*, *GPS*, *support* java, mendukung banyak jaringan (GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, and WiMAX) dan juga kemampuan dasar *handphone* pada umumnya.

2.5.1. Sejarah Android

Android pertama kali dikembangkan pada tahun 2003 oleh sebuah perusahaan **Android Inc.** Android, Inc didirikan di Palo Alto, California, Amerika Serikat pada bulan Oktober 2003 oleh Andy Rubin (pendiri Danger), Rich Miner (co-pendiri Wildfire Communications, Inc), Nick Sears (Vice Prsident T-Mobile), dan Chris White (pemimpin desain dan pengembangan di Webtv).

Sejak saat itu Google mulai melirik untuk mengembangkan bisnis di pasar Smart Phone. Dan pada akhirnya tahun 2005 Google membeli perusahaan **Android Inc.**

Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance* (OHA), konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Samsung, dan Nvidia.

Android dibangun menggunakan kernel **Linux**. Dengan *library* dan **API** yang ditulis dengan bahasa pemrograman C. Serta perangkat lunak aplikasi yang berjalan pada kerangka aplikasi yang mencakup Java *Library* yang berbasis pada Apache Harmony.

Saat ini terdapat dua jenis distributor sistem operasi Android di dunia. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD). Sistem Operasi Android membuka pintu untuk para developer mengembangkan software ini dengan Android SDK (Software Development Kit), yang menyediakan tool dan API yang dibutuhkan untuk memulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemograman Java.

Hal ini membuka kesempatan bagi para penggemar open source untuk ikut terjun mengembangkan sistem operasi Android. Kemudian mulai bermunculan berbagai komunitas yang membangun dan berbagi sistem Android berbasis firmware dengan sejumlah penyesuaian dan fitur-fitur tambahan, seperti FLAC

lossless audio dan kemampuan untuk menyimpan download aplikasi pada microSD card. Mereka sering memperbaharui paket-paket firmware dan menggabungkan elemen-elemen fungsi Android yang belum resmi diluncurkan ke dalam suatu firmware.

Pada bulan September tahun 2007 Google mulai mengajukan hak paten aplikasi telepon seluler. Disusul dengan dikenalkanya perangkat seluler Android pertama pada tahun 2008, HTC Dream. Perangkat ini menggunakan sistem operasi versi pertama Android 1.0.

Saat ini sistem operasi Android menjadi pilihan yang menguntungkan bagi banyak vendor smart phone karena memiliki biaya lisensi yang lebih murah dan sifatnya yang semi open source. Selain itu Android tentunya juga akan support dengan berbagai layanan dari Google.

Saat ini OS Android telah sampai versi 4.1 dengan code name **Jelly Bean**. Berikut adalah sejarah perkembangan versi OS Android :

a) **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.

b) **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).

c) **Android 1.1**

Pada 9 Februari, 2009, Android 1.1 di luncurkan, awalnya hanya untuk T-Mobile G1. Versi ini guna memecahkan masalah (bug), merubah API dan menambahkan sejumlah fitur dibanding versi 1.0. 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.

d) **Android 1.5 (CupCake)**

Pada 30 April, 2009, Android 1.5 di rilis, dinamakan CupCake, berbasis Linux Kernel 2.6.27. Update ini membawa sejumlah fitur baru dan perubahan pada antarmuka (UI).

e) **Android 1.6 (Donut)**

Pada 15 September, 2009 SDK Android 1.6 diberi nama Donut, dirilis, bebasis Linux Kernel 2.6.29. Di dalamnya terdapat sejumlah fitur baru. Di antaranya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

f) **Android 2.0 / 2.1 (Eclair)**

Pada 26, Oktober, 2009, SDK Android 2.0 dengan nama kode Eclair dirilis, berbasis Linux Kernel 2.6.29. Dan Motorola Droid perangkat yang menggunakan Eclair saat itu. 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.

g) **Android 2.2.x (Froyo : Frozen Yoghurt)**

Pada tanggal 20 bulan Mei, 20120, SDK Android 2.2 (Froyo) di rilis, berbasis Linux Kernel 2.6.32. 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.Dan Google Nexus One merupakan ponsel pintar pertama yang menggunakan Android Froyo.

h) **Android 2.3.x (Gingerbread)**

Pada tanggal 6, Desember, 2010, SDK Android 2.3 (Gingerbread) diluncurkan, berbasis Linux Kernel 2.6.35. 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. Dan Google Nexus S merupakan perangkat mobile pertama yang hadir dengan Android 2.3 Ginggerbread ini.

i) **Android Honeycomb 3.x (Honeycomb)**

Pada tanggal 22, Februari, 2011, SDK Android 3.0 (Honeycomb) yang merupakan versi Android pertama yang diperuntukan untuk perangkat tablet di luncurkan, berbasis Linux Kernel 2.6.36. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras untuk grafis.Dan perangkat tablet komputer pertama yang gunakan versi ini, ialah Motorola Xoom, yang dirilis pada 24, Februari, 2011.

j) **Android 4.0.x (Ice Cream Sandwich)**

SDK untuk Android 4.0.1 atau Ice Cream Sandwich dirilis untuk publik pada 19, Oktober,2011 dan berbasis Linux Kernel 3.0.1. Gabe Cohen dari Google menegaskan bahwa Android 4.0 secara teoritis cocok dengan perangkat Android 2.3 keatas dalam produksi masa itu. Android ICS 4.0 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. Kode sumber untuk Android 4.0 tersedia secara publik pada 14 November, 2011.

k) **Android 4.1.x (Jelly Bean)**

Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat.

2.5.2. The *Dalvik Virtual Machine* (DVM)

Salah satu elemen kunci dari android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di Dalvik Virtual Machine (DVM) bukan di Java Virtual Machine (JVM), sebernarnya banyak persamaan dengan Java Virtual Machine (JVM) seperti Java ME (Java Mobile Edition), tetapi android menggunakan Virtual Machine sendiri yang diskostumisasi dan dirancang untuk memastikan bahwa beberapa fitur-fitur berjalan lebih efisien pada perangkat mobile.

Dalvik Virtual Machine (DVM) adalah "register bases" sementara java Virtual Machine (JVM) bersifat "stack based", DVM didesain dan ditulis oleh Dan Bornsten dan beberapa enginers Google lainnya. Jadi bisa dikatakan "Dalvik equal(Java) == False" Dalvik Virtual Machine menggunakan kernel linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, threading, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C/C++ sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dan proses sistem dari

kernel linux yang digunakan dalam Android tersebut.

Semua hardware yang berbasis android dijalankan dengan menggunakan Virtual Machine untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi pernagkat keras tertentu. Dalvik Virtual Machine mengeksekusi Executable file, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. The Executable file diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan tools yang disediakan dalam SDK Android.

2.5.3. Arsitektur Android

Secara garis besar arsitektur android dapat dijelas dan digambarkan sebagai berikut :

1. *Application and Widgets*

   Application and widgets adalah layer dimana berhubungan dengan aplikasi dan biasanya download aplikasi kemudian lakukan instalasi dan jalankan aplikasi tersebut, delayer inilah terdapat seperti aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

2. *Application Frameworks*

   Android adalah “Open Development Platform” yaitu android menawarkan kepada pengembang atau member kemampuan kepada pengembangan untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan serive background, mengatur alarm, dan menambahkan tambahan seperti

statusnotifications dan masih banyak lagi. Pengembang memiliki akses penuh menuju API Framework seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya dengan mudah dapat menggunakan komponen yang sudah digunakan (reuse). Sehingga bisa disimpulkan Application Framework adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem android, karena pada layer inilah aplikasi dapat dirancang, seperti content-providers yang berupa sms dan lain sebagainya.

Komponen-komponen yang termasuk didalam application Framework adalah sebagai berikut :

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

3. *Libraries*

   Libraries adalah layer dimana fitur-fitur android berada biasanya para pembuat aplikasi kebanyakan mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL, serta :

   a) *Libraries* media untuk pemutar media audio dan video.
   b) *Libraries* untuk manajemen tampilan.
   c) *Libraries* Graphics mencakup SGL dan OpenGL untuk grafis 2D

dan 3D.

d) *Libraries* SQLite untuk dukungan *database*.

e) *Libraries* SSL dan WebKit terintegrasi dengan web browser dan security

f) Libraries Live Webcore mencakup modern web browser dengan engine embedded web view

4. Android Run Time

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. Dalvik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Didalam Android Run Time dibagi menjadi tiga bagian yaitu:

1. Core Libraries

   Aplikasi android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesin bukan Java Virtual Machine, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa Java/C yang dihandle oleh Core Libraries

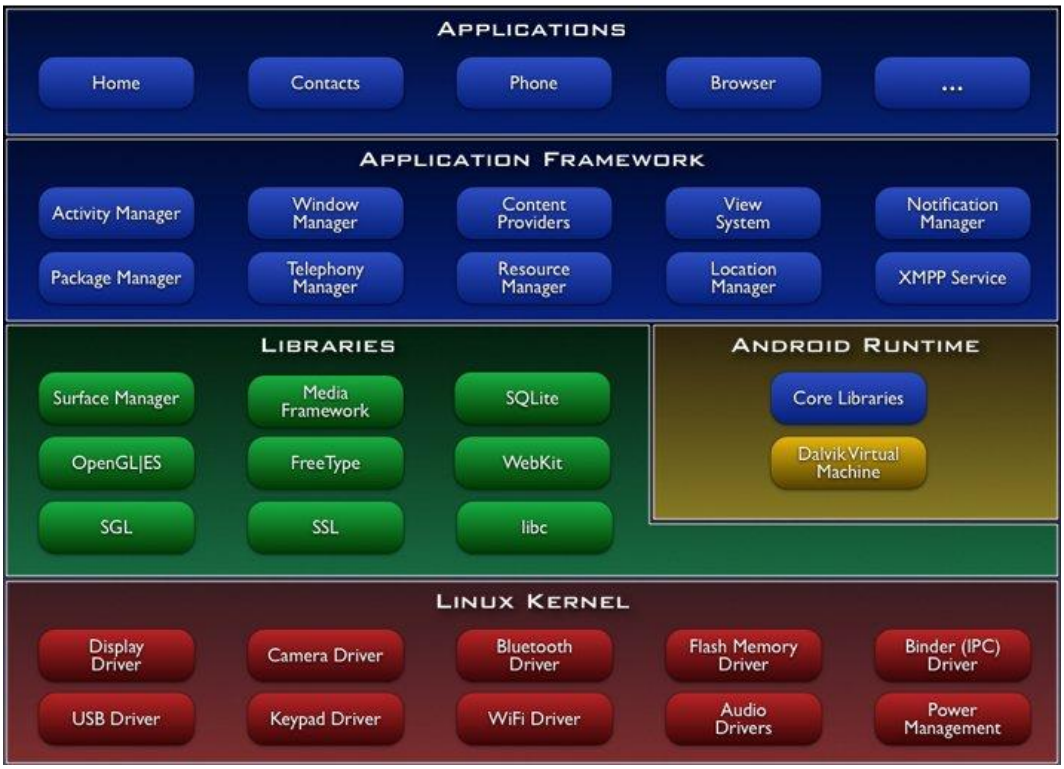
2. Dalvik Virtual Machine

   Virtual mesin yang berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien dimana merupakan pengembangan yang mampu membuat linux kernel untuk threading dan manajemen tingkat rendah.

3. Linux Kernel

   Linux kernel adalah layer dimana inti dari operating sistem dari

android itu sendiri, berisi file-file sistem yang mengatur sistem processing, memory, resources, drivers, dan sistem-sistem operanting andoroid lainnya. Linux kernel yang digunakan android adalah linux kernel release 2.6.



Gambar 2.1. Arsitektur Android

2.5.4. 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.

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.

2. Sevices

   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.

3. Intents

   Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain

sebagainya. Intents tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. Intents dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin menampilkan URL, sistem akan menentukan komponen apa yang dibutuhkan oleh Intents tersebut.

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.

5. Content Providers

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

2.5.5. 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 memberi kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone/Smartphone. Beberapa fitur - fitur Android yang paling penting adalah :

1. Framework Aplikasi yang mendukung penggantian komponen dan reusable
2. Mesin virtual Dalvik dioptimalkan untuk perangkat mobile
3. Integrated browser berdasarkan engine open source WebKit
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasrkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware)
5. SQLite untuk penyimpanan data (database)
6. Media Support yang mendukung audio, video, dan gambar.
7. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
8. Kamera, GPS, kompas, dan accelerometer (tergantung hardware)
9. Lingkungan Development yang lengkap dan kaya termasuk perangkat

emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

2.6. Eclipse Juno

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. Eclipse meluncurkan produk terbarunya yaitu eclipse Juno(versi 4.2) tentunya ada beberapa perubahan dari versi sebelumnya yaitu eclipse Indigo(versi 3.7).

2.7. XML

XML (Extensible Markup Language) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (HyperText Markup Language) yang merupakan bahasa standar untuk melacak Internet.

2.7.1. Tipe XML

XML memiliki tiga tipe file :

1. XML, merupakan standar format dari struktur berkas (file).
2. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. XSD, merupakan standar yang mendefinisikan struktur database dalam XML.

2.7.2. Keunggulan XML

Keunggulan XML bisa diringkas sebagai berikut :

1) Pintar (Intelligence). XML dapat menangani berbagai tingkat (level) kompleksitas.
2) Dapat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri.
3) Mudah pemeliharaannya.
4) XML lebih sederhana.
5) Mudah dipindah-pindahkan (Portability). XML mempunyai kemudahan perpindahan (portabilitas) yang lebih bagus.

2.8. MySQL

MySQL merupakan software yang tergolong database server dan bersifat Open Source. Open Source menyatakan bahwa software ini di lengkapi oleh source code (kode yang di pakai untuk membuat MySQL), selain tentu saja bentuk executable-nya atau kode dapat di jalankan secara langsung di dalam sistem operasi, dan bias diperoleh dengan cara mengunduh di internet secara gratis. Hal lainnya adalah MySQL juga bersifat multiplatform. MySQL dapat dijalankan pada berbagai sistem operasi. Abdul Kadir (2009:15)

2.8.1. Keunggulan MySQL

Sebagai database server yang memiliki konsep database modern, MySQL memiliki banyak keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL.

1) Portability

   MySQL dapat berjalan stabil pada berbagai system operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, dan lain-lain.

2) Open Source

   MySQL didistribusikan secara Open Source, sehingga dapat digunakan secara bebas.

3) Multi-user

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini memungkinkan sebuah database server MySQL dapat diakses client secara bersamaan.

4) Performance Tuning

   MySQL memiliki kecepatan yang tinggi dalam menangani query, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5) Column Types

MySQL memiliki tipe kolom yang sangat kompleks, seperti signed/unsigned integer, float, double, char, varchar, text, blob, date, time, timestamp, year, set.

6) Command dan Function

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.

7) Security

MySQL memiliki lapisan lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan system perizinan yang mendetail serta password terenkripsi.

8) Scalability dan Limits

MySQL mampu menangani database dengan skala besar, dengan jumlah records lebih dari 50 juta dan 60 juta table serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.

9) Connectivity

MySQL dapat melakukan koneksi dengan client menggunakan protocol TCP/IP, Unix Socet (UNIX), atau Named Pipes (NT).

10) Localisation

Deteksi pesan kesalahan pada client dengan menggunakan lebih dari 20 bahasa.

11) Interface

Interface terhadap berbagai aplikasi dan bahasa pemrograman dengan

menggunakan fungsi API (Application Programming Interface)

12) Client dan Tools

Dilengkapi dengan berbagai Tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

13) Struktur Tabel

Struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibanding database lainnya semacam PostgreSQL ataupun Oracle.

2.9. PHP (Personal Home Page)

PHP adalah program aplikasi yang bersifat server side, artinya hanya dapat berjalan pada sisi server saja dan tidak dapat berfungsi tanpa adanya sebuah server di dalamnya. PHP juga bukan bahasa pemograman yang lengkap. (Bunafit Nugroho, 2005:369)

Maksud dari definisi tersebut yakni program ini tidak menyertakan compiler tersendiri yang membuat program hasilnya menjadi program .exe yang dapat di jalankan tersendiri. PHP (Hypertext Preprocessor) merupakan sebuah bahasa pemograman scripting berlisensi Open Source. Script ini dapat bercampur dengan script Tag HTML sehingga karena kemampuannya tersebut, ia disebut bahasa yang embeded pada Tag HTML.

2.9. JSON PHP

JSON (JavaScript Object Notation) adalah sebuah format data-interchange yang ringan. Sangat mudah bagi manusia untuk membaca dan menuliskannya. Semudah mesin memparsing dan men-generate format datanya. JSON merupakan bagian dari bahasa pemrograman JavaScript (Standard ECMA-262 3rd Edition – December 1999).

JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan sebagainya. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa data-interchange yang ideal. JSON dibangun dalam dua struktur:

1. Beberapa pasangan dari nama/nilai. Dalam beberapa bahasa perograman biasa disebut dengan istilah object, record, struct, tabel hash, key list atau associative array.
2. Nilai-nilai yang terusun secara ordered list. Biasa disebut dengan array, vector, list atau daftar dalam bahasa pemrograman.

JSON adalah struktur data yang universal, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON. Format data JSON mempunyai aturan sebagai berikut :

1. Object adalah satu set nama/nilai yang tidak terurut (An object is an unordered set of name/value pairs). Penulisan object dimulai dengan tanda { (left brace) dan diakhiri dengan tanda } (right brace). Setiap nama diikuti oleh tanda : (colon) dan pasangan nama/nilai dipisahkan dengan tanda , (comma).
2. Array adalah sekumpulan nilai yang teratur (An array is an ordered collection of values). Penulisan sebuah array dimulai dengan tanda [(left

bracket) dan diakhiri dengan tanda] (right bracket). Nilai dipisahkan menggunakan tanda , (comma).

3. Nilai bisa berupa string dalam tanda kutip, atau number (angka), TRUE atau FALSE atau NULL, sebuah object atau sebuah array. Struktur ini dapat ditulis menggunakan metode bersarang.
4. String adalah rangkaian atau urutan karakter unicode yang berada daslam tanda kutip, bisa juga hanya berisi karakter kosong, menggunakan tanda \ (backslash) untuk escape. Karakter di representasikan sebagai string tunggal. Tipe data string pada JSON sangat mirip dengan definisi pada bahasa C atau Java.
5. Number sangat mirip dengan definisi pada bahasa C atau Java, hanya saja tipe bilangan oktal dan heksadesimal tidak digunakan.
6. Ruang kosong (Whitespace) dapat diselipkan diantara pasangan atau token apa saja. Kecuali **detail encoding**, yang secara jelas akan berfungsi untuk menjelaskan bahasa yang digunakan.

JSON sangat berguna saat kita sering melakukan pertukaran data pada sebuah aplikasi. Contoh nyata adalah penggunaan tabulasi menggunakan JqGrid, yang merupakan salah satu plugin jQuery untuk menampilkan data dalam bentuk spreadsheet. Plugin ini juga mendukung format JSON, selain format XML.

2.10. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (Object-Oriented Programming) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-

objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah mempermudah programmer didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebutuntuk kemudian dimodelkan dalam sistem nyata.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

2.11. UML (Unified Modelling Language)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat

(tool) desain berorientasi objek dari IBM.UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2.11.1. Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

1. Use Case Diagram untuk memodelkan proses bisnis.
2. Conceptual Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. Sequence Diagram untuk memodelkan pengiriman pesan (message) antar objects
4. Collaboration Diagram untuk memodelkan interaksi antarobjects.
5. State Diagramuntuk memodelkan perilaku object di dalam sistem.
6. Activity Diagra muntuk memodelkan perilaku Use Cases dan object di dalam system.
7. Class Diagram untuk memodelkan struktur kelas.
8. Object Diagram untuk memodelkan strukturobject.

9. Component Diagram untuk memodelkan komponen object.
10. Deployment Diagram untuk memodelkan distribusi aplikasi.

2.11.2. Kelebihan UML

Kelebihan UML dibandingkan dengan bahasa permodelan yang lain antara lain:

1) Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2) Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep inti.
3) Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4) Menyediakan basis formal untuk bahasa pemodelan.
5) Memadukan praktek-praktek terbaik di industri perangkat lunak menjadi terminologi dan notasi yang diterima luas.
6) Menyediakan kemampuan merepresentasikan semua konsep yang relevan untuk sistem perangkat lunak.
7) Menyediakan fleksibilitas yang diperlukan bagi konsep-konsep perangkat lunak yang baru.

2.11.3. Kekurangan UML

Sedangkan kekurangan UML antara lain:

1) UML bukanlah bahasa pemrograman visual, melainkan bahasa pemodelan visual.
2) UML bukan spesifikasi dari tool, tapi spesifikasi bahasa pemodelan.
3) UML bukanlah proses, tapi yang memungkinkan proses-proses.

2.12. Internet

Internet adalah sekumpulan jaringan komputer yang saling terhubung secara fisik dan memiliki kemampuan untuk membaca dan menguraikan protokol komunikasi tertentu yang disebut Internet Protocol (IP) dan Transmission Control Protocol (TCP). Protokol adalah spesifikasi sederhana mengenai bagaimana komputer saling bertukar informasi.

Internet merupakan seluruh jaringan yang saling berhubungan secara fisik dan memiliki kemampuan untuk membaca dan menguraikan protokol komunikasi yang disebut TCP/IP.