BAB II

LANDASAN TEORI

2.1. Aplikasi

Menurut Dhanta (2009:32), *aplikasi adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu*. Jadi perangkat lunak aplikasi yaitu perangkat lunak yang digunakan untuk membantu pemakai komputer untuk melaksanakan pekerjaannya. Jika ingin mengembangkan program aplikasi sendiri, maka untuk menulis program aplikasi tersebut, dibutuhkan suatu bahasa pemrograman, yaitu language software, yang dapat berbentuk assembler, compiler ataupun interpreter. Jadi language software merupakan bahasanya dan program yang ditulis merupakan program aplikasinya. Language software berfungsi agar dapat menulis program dengan bahasa yang lebih mudah, dan akan menterjemahkannya ke dalam bahasa mesin supaya bisa dimengerti oleh komputer. Bila hendak mengembangkan suatu program aplikasi untuk memecahkan permasalahan yang besar dan rumit, maka supaya program aplikasi tersebut dapat berhasil dengan baik, maka dibutuhkan prosedur dan perencanaan yang baik dalam mengembangkannya.

Sekarang, banyak sekali program-program aplikasi yang tersedia dalam bentuk paket-paket program. Ini adalah program-program aplikasi yang sudah ditulis oleh orang lain atau perusahaan-perusahaan perangkat lunak. Beberapa perusahaan perangkat lunak telah memproduksi paket-paket perangkat lunak yang mempunyai

reputasi internasional. Program-program paket tersebut dapat diandalkan, dapat memenuhi kebutuhan pemakai, dirancang dengan baik, relatif bebas dari kesalahan-kesalahan, user friendly (mudah digunakan), mempunyai dokumentasi manual yang memadai, mampu dikembangkan untuk kebutuhan mendatang, dan didukung perkembangannya. Akan tetapi, *bila permasalahannya bersifat khusus dan unik, sehingga tidak ada paket-paket program yang sesuai untuk digunakan, maka dengan terpaksa harus mengembangkan program aplikasi itu sendiri*. (Jogiyanto Hartono, 2004, pp397-398).

2.2. Angkutan Kota

Angkutan kota adalah pemindahan manusia atau barang dari satu tempat ke tempat lainnya dengan menggunakan sebuah kendaraan yang digerakkan oleh manusia atau mesin. Transportasi digunakan untuk memudahkan manusia dalam melakukan aktivitas sehari-hari. Di negara maju, mereka biasanya menggunakan kereta bawah tanah (subway) dan taksi. Penduduk disana jarang yang mempunyai kendaraan pribadi karena mereka sebagian besar menggunakan angkutan umum sebagai transportasi mereka.

2.2.1. Karakteristik Transportasi Angkutan

Transportasi sendiri dibagi 3 yaitu, transportasi darat, laut, dan udara. Transportasi udara merupakan transportasi yang membutuhkan banyak uang untuk memakainya. Selain karena memiliki teknologi yang lebih canggih, transportasi udara merupakan alat transportasi tercepat dibandingkan dengan alat transportasi lainnya, dan Transportasi darat adalah segala bentuk transportasi menggunakan jalan untuk

mengangkut penumpang atau barang. Bentuk awal dari transportasi darat adalah menggunakan kuda, keledai atau bahkan manusia untuk membawa barang melewati jalan setapak. Seiring dengan berkembangkan perdagangan, jalan diratakan atau dilebarkan untuk mengakomodir aktivitas. Roda kemudian ditemukan.

2.3. Basis Data

Menurut Chendramata (2009:106), *Basis Data adalah sebuah perangkat lunak yang dirancang dan diperuntukkan sebagai media untuk menyimpan data-data transaksi yang dihasilkan pada sebuah proses bisnis*. Basis data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya.

Basis data sendiri dapat didefinisiikan dalam sejumlah sudut pandang seperti : Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimjpanan elektronis.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya

adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronis seperti disk (disket atau hard disk). Hal ini merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola/ditangani oleh manusia, sementara basis data dikelola/ditangani melalui perantara alat/mesin pintar elektronis (yang dikenal sebagai komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metoda/cara yang dapat digunakan dalam upaya penyimpanan.

2.3.1. Operasi Dasar Basis Data

Didalam sebuah disk, basis data dapat diciptakan dan dapat pula ditiadakan. Didalam sebuah disk, kita dapat pula menempatkan beberapa (lebih dari satu) basis data. Sementara dalam sebuah basis data, kita dapat menempatkan satu atau lebih file/tabel. Pada file/tabel inilah sesungguhnya data disimpan/ditempatkan. Setiap basis data umumnya dibuat untuk mewakili sebuah semesta data yang spesifik. Misalnya, ada basis data kepegawaian, basis data akademik, basis data inventori (Pergudangan), dan sebagainya. Sementara dalam basis data akademik, misalnya, kita dapat menempatkan file mahasiswa, file mata kuliah, file dosen, file jadwal, file kehadiran, file nilai, dan seterusnya. Karena itu, operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi :

1. Pembuatan basis data baru (create database), yang identik dengan pembuatan lemari arsip yang baru
2. Penghapusan basis data (drop database), yang identik dengan perusakan lemari arsip (sekaligus beserta isinya, jika ada).
3. Pembuatan file/tabel dari suatu basis data (create table), yang identik dengan penambahan map arsip baru ke sebuah lemari sarsip yang telah ada.
4. Penghapusan file/tabel dari suatu basis data (drop table), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
5. Penambahan/pengisian data baru ke sebuah file/tabel disebuah basis data (insert), yang identik dengna penambahan ke lemari arsip ke sebuah map arsip
6. Pengambilan data dari sebuah file/tabel (retrieve/search) yang identik dengan pencarian lembaran arsip dari sebuah map arsip
7. Pengubahan data dari sebuah file/tabel (update), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
8. Penghapusan data dari sebuah file/tabel (delete), yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

Operasi yang berkenaan dengan pembuatan objek (basis data dan tabel) merupakan operasi awal yang hanya dilakukan sekali dan berlaku seterusnya. Sedang operasi-operasi yang berkaitan dengan isi tabel (data) merupakan operasi rutin yang akan berlangsung berulang-ulang dan arena itu operasi-operasi inilah yang lebih tepat mewakili aktivitas pengelolaan (management) dan pengolahan (processing) data dalam basis data.

2.3.2. Objektif Basis Data

Telah disebutkan di awal bahwa tujuan awal dan utama dalam pengelolaan data dalam sebuah basis data adalah agar kita dapat memperoleh menemukan kembali data (yang dicari) dengan mudah dan cepat. Disamping itu, pemanfaatan basis data untuk pengelolaan data, juga memiliki tujuan-tujuan lain. Secara lebih lengkap, pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) seperti berikut ini :

1. Kecepatan dan kemudahan

Pemanfaatan basis data memungkinkan untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah, daripada jika menyimpan data secara manual (non-elektronis) atau secara elektronis (tetapi tidak dalam bentuk penerapan basis data, misalnya dalam bentuk spread sheet atau dokumen teks biasa).

2. Efisiensi ruang penyimpanan (Space)

Karena keterkaitan yang erat antara kelompok data dalam sebuah basis data, maka redudansi (pengulangan) data pasti akan selalu ada. Banyaknya redudansi ini tentu akan memperbesar ruangan penyimpanan (baik di memori utama maupun memori sekunder) yang harus disediakan. Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena kita dapat melakukan penekanan jumlah redudansi data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antara kelompok data yang saling berhubungan.

3. Keakuratan (Accurancy)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (constrain) tipe data, domain data, keunikan data, dan sebagainya, yang secara ketat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

4. Ketersediaan (Availability)

Pertumbuhan data (baik dari sisi jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu selalu kita gunakan/butuhkan. Karena itu kita dapat memilah adanya data utama/master/referensi, data transaksi, data histori hingga data kadaluarsa. Data yang sudah jarang atau bahkan tidak pernah lagi kita gunakan, dapat kita atur untuk dilepaskan dari sistem basis data yang sedang aktif (menjadi off-line) baik dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan off-line (seperti removable disk, atau tape). Di sisi lain, karena kepentingan pemakaian data, sebuah geografis. Data nasabah sebuah bank, misalnya, dipisah-pisahkan dan disimpan di lokasi yang sesuai dengna keberadaan nasabah. Dengan pemanfaatan teknologi jaringan komputer, data yang berada di suatu lokasi/cabang, dapat juga di akses (menjadi tersedia/available) bagi lokasi/cabang lain.

5. Kelengkapan (Completeness)

Lengkap/tidaknya data yang kita kelola dalam sebuah basis data bersifat relatif (baik terhadap kebutuhan pemakai maupun terhadap waktu). Bila seorang pemakai sudah menganggap bahwa data yang dipelihara sudah lengkap, maka pemakai yang

lain belum tentu berpendapat sama. Atau, yang sekarang diangap sudah lengkap belum tentu di masa yang akan data juga demikian. Dalam sebuah basis data, disamping data kita juga harus menyimpan struktur (baik yang mendefinisikan objek-objek dalam basis data maupun definisi detail dari tiap objek seperti struktur file/tabel atau indeks). Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita tidak hanya dapat menambahkan record-record data, tetapi juga dapat melakukan perubahan struktur dalam basis data baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan field-field baru pada suatu tabel.

6. Keamanan (Security)

Memang ada sejumlah sistem (aplikasi) pengelola basis data yang tidak menerapkan aspek keamanan dalam penggunaan basis data. Tetapi untuk sistem yang besar dan serius, aspek keamanan juga dapat diterapkan dengan ketat. Dengan begitu kita dapat menentukan siapa-siapa (pemakai) yang boleh menggunakan basis data beserta objek-objek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukannya.

7. Kebersamaan Pemakai (Sharebility)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja, atau di satu lokasi saja atau boleh satu sistem/aplikasi saja, data pegawai dalam basis data kepegawaian, misalnya, dapat digunakan oleh banyak pemakai, dari sejumlah departemen dalam perusahaan atau oleh banyak ssitem (sistem penggajian, sistem akuntansi, sistem inventori, dan sebagainya). Basis data yang dikelola oleh sistem

(aplikasi) yang mendukung lingkungan multiuser, akan dapat memenuhi kebutuhan ini, tetapi tetap dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat yang bersamaan) atau kondisi deadlock (karena banyak pemakai yang saling menunggu untuk menggunakan data).

2.4. Global Positioning System (GPS)

GPS adalah sistem radio navigasi dan penentuan posisi dengan menggunakan satelit navigasi yang dimiliki dan dikelola oleh Departemen Pertahanan Amerika Serikat. Nama formalnya adalah NAVSTAR GPS (Navigation Satellite Timing and Ranging Global Positioning System).

2.4.1. Sistem Navigasi GPS dan Sinyal GPS

Sistem ini digunakan untuk memberikan informasi mengenai posisi, waktu dan kecepatan kepada siapa saja secara global tanpa ada batasan waktu dan cuaca. Satelit GPS pertama diluncurkan pada tahun 1978 dan secara resmi sistem GPS dinyatakan operasional pada tahun 1994. Sinyal GPS yang dipancarkan oleh satelit-satelit GPS menggunakan band frekuensi L pada spektrum gelombang elektromagnetik. Setiap satelit GPS memancarkan dua (2) gelombang pembawa yaitu L1 dan L2 yang berisi data kode dan pesan navigasi. Pada dasarnya sinyal GPS terdiri dari tiga komponen, yaitu: penginformasi jarak (kode), penginformasi posisi satelit (navigation message) dan gelombang pembawanya (carrier wave).

2.4.2. Penginformasi Jarak

Penginformasi jarak yang dikirimkan oleh satelit GPS terdiri dari dua buah kode PRN (Pseudo Random Noise) yaitu kode-C/A (Coarse Acquisition/Clear Access) yang dimodulasikan pada gelombang pembawa L1 dan kode-P(Y) (Private) yang dimodulasikan baik pada gelombang pembawa L1 maupun L2. Kedua kode tersebut disusun oleh rangkaian kombinasi bilanganbilangan biner (0 dan 1). Setiap satelit GPS mempunyai struktur kode yang unik dan berbeda antara satu satelit dengan satelit lainnya yang memungkinkan receiver GPS untuk membedakan sinyal-sinyal yang datang dari satelit-satelit GPS yang berbeda. Sinyal-sinyal tersebut dapat dibedakan oleh receiver dengan menggunakan teknik yang dinamakan CDMA (Code Division Multiple Accsess).

2.4.3. Penginformasi Posisi

Pesan navigasi yang dibawa oleh sinyal GPS terdiri dari informasi ephemeris (orbit) satelit yang biasa disebut broadcast ephemeris yang terdiri dari parameter waktu, parameter orbit satelit dan parameter perturbasi dari orbit satelit. Parameter – parameter tersebut digunakan untuk menentukan koordinat dari satelit. Disamping broadcast ephemeris , pesan navigasi juga berisi almanac satelit yang memberikan informasi tentang orbit nominal satelit yang berguna bagi receiver dalam proses akuisasi awal data satelit maupun bagi para pengguna dalam perencanaan waktu pengamatan yang optimal. Informasi lain yang dibawa oleh pesan navigasi adalah koefisien koreksi jam satelit, parameter koreksi ionosfer, status konstelasi satelit dan informasi kesehatan satelit.

2.4.4. Penentu Posisi Absolut dengan GPS

Penentuan posisi dengan GPS adalah penentuan posisi tiga dimensi yang dinyatakan dalam sistem koordinat kartesian (X,Y,Z) dalam datum WGS (World Geodetic System) 1984. Untuk keperluan tertentu, koordinat kartesian tersebut dapat dikonversi ke dalam koordinat geodetik (f,.,h). Titik yang akan ditentukan posisinya dapat diam (static positioning) maupun bergerak (kinematic positioning). Penentuan posisi absolut merupakan metode penentuan posisi yang paling mendasar dan paling banyak digunakan untuk aplikasi-aplikasi yang tidak memerlukan tingkat ketelitian posisi yang tinggi dan tersedia secara instant real-time) seperti pada aplikasi navigasi wahana bergerak (darat, laut dan udara).

2.4.5. Prinsip Penentu Posisi Absolut dengan GPS

Prinsip dasar penentuan posisi absolut dengan GPS adalah dengan reseksi jarak ke beberapa satelit GPS sekaligus yang koordinatnya telah diketahui. Pada penentuan posisi absolut dengan data pseudorange, jarak pengamat (receiver) ke satelit GPS ditentukan dengan mengukur besarnya waktu tempuh sinyal GPS dari satelit ke receiver pengamat.Waktu tempuh ditentukan dengan menggunakan teknik korelasi kode (code correlation technique) dimana sinyal GPS yang datang dikorelasikan dengan sinyal replika yang diformulasikan dalam receiver. Jarak dari receiver ke pengamat kemudian dapat ditentukan dengan mengalikan waktu tempuh dengan kecepatan cahaya. Karena ada perbedaan waktu pada jam satelit dan jam receiver maka data jarak yang diperoleh bukan merupakan jarak yang sebenarnya melainkan jarak pseudorange.

2.4.6. Format Data Keluaran GPS.

Receiver GPS memiliki format keluaran sebanyak lima (5) jenis yaitu NMEA 0180, NMEA 0182, NMEA 0183, AVIATION, dan PLOTTING. Format data tersebut ditetapkan oleh NMEA (National Maritime Electronic Association) dan dapat dikoneksikan ke komputer melalui port komunikasi serial dengan menggunakan kabel RS-232. Data keluaran dalam format NMEA 0183 berbentuk kalimat (string) yang merupakan rangkaian karakter ASCII 8 bit. Setiap kalimat diawali dengan satu karakter '$' , dua karakter Talker ID, tiga karakter Sentence ID, dan diikuti oleh data fileds yang masing-masing dipisahkan oleh koma serta diakhiri oleh optional cheksum dan karakter cariage return/line feed (CR/LF). Jumlah maksimum karakter dihitung dari awal kalimat ($) sampai dengan akhir kalimat (CR/LF) adalah 82 karakter.

2.4.7. Cara kerja GPS

Navigasi satelit didasarkan pada jaringan global satelit yang mengirimkan sinyal radio di orbit bumi menengah. Pengguna Navigasi satelit paling akrab dengan 24 Global Positioning System (GPS) satelit. Amerika Serikat, yang mengembangkan dan mengoperasikan GPS, dan Rusia, yang mengembangkan sistem serupa yang dikenal sebagai GLONASS, telah menawarkan sistem bebas menggunakan masing-masing kepada masyarakat internasional. Internasional Organisasi Penerbangan Sipil (ICAO), serta kelompok-kelompok pengguna internasional, telah menerima GPS dan GLONASS sebagai inti untuk kemampuan navigasi satelit sipil internasional dikenal sebagai Global Navigation Satellite System (GNSS).

2.5. Java

Java adalah bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya seperti Patrick Naugton, Chris Warth, Ed Frank, dan Mike Sheridan di suatu perusahaan perangkat lunak yang bernama Sun Microsystem, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama "oak", namun pada tahun 1995 diganti namanya menjadi "Java" Alasan utama pembentukan bahasa java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan diberbagai macam perangkat elektronik, seperti microwave oven dan remote control, sehingga Java harus bersifat portable atau yang sering disebut dengan platform independent (tidak tergantung pada platform). Itulah yang menyebabkan dalam dunia pemrograman Java, dikenal adanya istilah 'write once, run everywhere', yang berarti kode program hanya ditulis sekali, namun dapat dijalankan dibawah platform manapun, tanpa harus melakukan perubahan kode program.

2.5.1. Arsitektur Java

Secara arsitektur, Java tidak berubah sedikitpun semenjak awal mula bahasa tersebut dirilis. Kompiler Java (yang disebut dengan Javac atau Java Compiler) akan mentransformasikan kode-kode dalam bahsa Java ke dalam suatu bytecode. Apa itu bytecode? Bytecode adalah sekumpulan perintah hasil kompilasi yang kemudian dapat dieksekusi melalui sebuah mesin komputer abstrak, yang disebut dengan JVM (Java Virtual Machine). JVM juga sering dinamakan sebagai interpreter, kiarena sifatnya yang selalu menerjemahkan kode-kode yang tersimpan dalam bytecode dengan cara baris demi baris.

2.5.2. Java Versi Lama (Java 1)

Pada awal perilisannya, versi Java masih disebut dengan JDK (Java Development Kit). Dalam JDK, semua kebutuhan untuk pengembangan program dan eksekusi program masih tergabung jadi satu. Penamaan ini berlaku sampai Java 1.1. namun sekarang, setelah Java 1.2, Sun Microsystem menamainya dengan JSDK (Java Software Development Kit) dalam hal ini kebutuhan untuk pengembangan program dipisahkan dengan kebutuhan kesekusi. Bagian software yang digunakan untuk kebutuhan eksekusi program disebut dengan JRE (Java-Runtime Environment). Selanjutnya, Java 1.2 disederhanakan penamaanya menjadi "Java 2"

2.5.3. Java 2

Sun Microsystems telah mendefinisikan tiga buah edisi dari Java 2, yaitu sebagai berikut:

1. Java 2 Standard Edition (J2SE), yang digunakan untuk mengembangkan aplikasi-aplikasi desktop dan applet (aplikasi Java yang dapat dijalankan di dalam browser web).
2. Java 2 Enterprise Edition (J2EE), merupakan superset dari J2SE yang memperbolehkan untuk mengembangkan palikasi-aplikasi berskala besar (enterprise), yaitu dengan melakukan pembuatan aplikasi-aplikasi di sisi server dengan menggunakan EJBs (Enterprise JavaBeans), aplikasi web dengan menggunakan Servlet dan JSP (JavaServer Pages) dan teknologi lainnya serperti CORBA (Common Object Request Broker Architecture) dan XML (Extensible Markup Language).

3. Java 2 Micro Edition (J2ME), merupakan subset dari J2SE yang digunakan untuk menangani pemrograman di dalam perangkat-perangkat kecil, yang tidak memungkinkan untuk mendukung implementasi J2SE secara penuh.

2.5.4. J2ME

J2ME merupakan sebuah kombinasi yang terbentuk antara kumpulan interface Java yang sering disebut dengan Java API (Application programming interface) dengan JVM (Java Virtual Machine) yang didesain khusus untuk alat, yaitu JVM dengan ruangan terbatas. Kombinasi tersebut kemudian digunakan untuk melakukan pembuatan aplikasi-aplikasi yang dapat berjalan di atas alat (dalam hal ini mobile device). J2ME sendiri pada dasarnya tterdiri dari tiga buah bagian, yaitu konfigurasi, profil, dan paket-paket opsional.

2.6. Android

Android sebagai suatu sistem, adalah sistem operasi yang berbasis java yang dijalankan pada kenrnel linux 2.6. Sistem ini sangat ringan dan berfitur lengkap (DiMarzio, 2008, p6). Pembahasan mengenai android dan tools yang digunakan dalam pembuatan aplikasi, akan dijelaskan pada sub bab berikut.

2.6.1. Sejarah Android

Android adalah sebuah sistem operasi untuk perangkat lunak mobile berbasis inux yang mencakup sistem 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 dibawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

Didunia ini terdapat dua jenis distributor sistem operasi android. 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).

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis smartphone yang menggunakan android sebagai sistem operasinya. Telpon 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 AEM Holdings, Atheros Communication, diproduksi oleh Asustek Komputer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, android perangkat mobile yang merupakan modifikasi kernel Linux 2.6. sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Pada masa saat ini kebanyakan vendor-vendor smartphone sudah memproduksi smartphone berbasis android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, LG, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi vendeor smartphone didunia yang memproduksi android. Hal ini karena android adalah sistem oerasi yang open source sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

Tidak hanya menjadi sistem operasi di smartphone, saat ini android menjadi pesaing utama dari Apple pada sistem operasi Table PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan diatas adalah karena android itu sendiri adalah platform sangat lengkap baik itu sistem operasinya, aplikasi dan Tool Developmen, Market aplikasi android serta dukungan yang sangat tinggi dari komunitas Open source didunia, sehingga android terus berkembang pesat dari segi teknologi maupun dari segi jumlah device yang ada didunia.

2.6.2. 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 status notifications 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 :

Views, Content Provider, Resource Manager, Notification Manager, 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 : Libraries media untuk pemutar media audio dan video, Libraries untuk manajemen tampilan, Libraries Graphics mencakup SGL dan OpenGL untuk grafis

2D dan 3D, Libraries SQLite untuk dukungan database, Libraries SSL dan WebKit terintegrasi dengan web browser dan security, 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 dua 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.

2.6.3. Fundamental Aplikasi

Aplikasi android ditulis dalam bahasa pemrograman java, kode java dikompilasi bersama data file resource yang dibutuhkan oleh aplikasi dimana prosesnya dipaket oleh tools yang dinamakan “apt tools” ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang sebenernya disebut dengan aplikasi yang dapat diinstal di perangkat mobile nantinya.

Ada enam jenis komponen pada aplikasi android yaitu:

1. 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 dapat dilakukan dengan satu even misalnya click tombol, memilih opesi atau menggunakan triggers tertentu. Secara hirarki sebuah window activity dinyatakan dengan method Activity.setContentView(). ContentView adalah objek yang berada pada root hirarki.

2. Service

Service tidak memiliki visual user interface (UI), tetapi service berjalan secara background, sebagai contoh dalam memainkan music, service mungkin memainkan

music atau mengambil data dari jaringan, tetapi setiap service haruslah berada dalam kelas induknya.

3. Broadcast Receiver

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh broadcast seperti notifikasi zona waktu berubah, baterai low, gambar telah selesai diambil oleh kamera, atau pengubah referensi bahasa yang digunakan.

4. Content Provider

Content Provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain.

2.7. Tools-Tools yang digunakan

Tools-Tools yang digunakan adalah sebagai berikut:

2.7.1. Konsep Perancangan Berorientasi Obyek

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 obke. Tujuannya adalah mempermudah programmer didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata.

Suatu perusahaan software yaitu Rational Software, telah membentuk konsarium dengan berbagai organisasi untuk meresmikan pemakaian Unified

Modelling Language (UML) sebagai bahasa standar dalam Object Oriented Analysist Design (OOAD).

2.7.2. Unified Modelling Language (UML)

Menurut Yasmi Afrizal dan Wahyuni (2012 : 181) *UML (Unified Modeling Language) merupakan pengganti dari metode analisis berorientasi objek dan desain berorientasi objek (OOA & OOD) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an.* UML (Unified Modelling Language) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, Object Modeling Technique (OMT) dan Object Oriented Software Engineering (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode Design Object Oriented. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iterative, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantic dari hubungan obyek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur pemodelan entity-relationship. Tahapan utama dalam metodologi ini adalah analisis, design sistem, design obyek dan implementasi.

Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih member penekanan pada use case. OOSE memiliki tiga tahapan yaitu membuat model requirement dan analisis, design dan implementasi, dan model pengujian (test model). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Design UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam dari pada metode lainnya.

2.7.3. Use Case Diagram

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Use Case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut scenario. Setiap scenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan use case adalah serangkaian scenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

Dalam pembicaraan tentang use case, pengguna biasanya disebut dengan aktor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model use case adalah bagian dari model requirement

(Jacob et all, 1992). Termasuk disini adalah problem domain object model dan penjelasan tentang user interface. Use case memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif user.

2.8. Eclipse

Menurut Ir. Yuniar Supardi (2011 : 11) *Eclipse merupakan perangkat lunak untuk bahasa pemrograman Java. Perangkat lunak ini berguna untuk membuat dan meng-compile source code bahasa pemrograman Java.* Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platformindependent).

Berikut ini adalah sifat dari Eclipse:

1. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

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.

2.8.1. Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Berikut ini adalah komponen yang membentuk RCP: Core platform, OSGi, SWT(Standard Widget Toolkit), JFace, Eclipse Workbench.

Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman Java. Konsep Eclipse adalah IDE yang terbuka (open), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik[2]. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya.

2.8.2. Android SDK

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, android member anda kesempatan unutk membuat aplikasi yang kita butuhkan yang ubkan merupakan aplikasi bawaan Hadphone/Smartphone. Beberapa fitur-fitur android yang paling penting adalah :

1. Framework : aplikasi yang mendukung pengganti komponen dan reusable.
2. Dalvik Virtual Machine dioptimalkan untuk perangkat mobile
3. Integrated Browser verdasarkan engine open source WebKit.
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional Ekselerasi hardware)
5. SQLite untuk penyimpanan data.
6. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, GIF), GSM Telephony (tergantung Hardware)
7. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
8. Kamera, GPS, Kompas, dan Accelerometer (tergantung hardware)

Lingkungan Development yang lengkap dan termasuk pernagkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse. Untuk source SDK Android ini dapat dilihat dan didownload langsung di situs resmi pengembang SDK Android di http://www.developer.android.com/