BAB II

LANDASAN TEORI

2.1. Aplikasi

Perangkat lunak memiliki pengertian menunjuk pada program dan alat bantu lain yang bersifat menambah kemampuan komputer sebagai alat untuk melaksanakan tugas atau operasi tertentu. Program aplikasi dapat dibuat secara khusus untuk memenuhi kebutuhan khusus pula (*tailor-made*) atau berupa paket yang mempunyai aplikasi umum.

2.1.1. Pengertian Aplikasi

Aplikasi adalah sebuah perangkat lunak yang berisi perintah untuk menyelesaikan masalah dan pengolahan data. Aplikasi disebut juga dengan perangkat lunak, merupakan kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya. perangkat lunak ini merupakan catatan bagi mesin komputer untuk menyimpan perintah, maupun dokumen serta arsip lainnya. Merupakan data elektronik yang disimpan sedemikian rupa oleh komputer itu sendiri, data yang disimpan ini dapat berupa program atau instruksi yang akan dijalankan oleh perintah, maupun catatan-catatan yang diperlukan oleh komputer untuk menjalankan perintah yang dijalankannya. Untuk mencapai keinginannya tersebut dirancanglah suatu susunan logika, logika yang disusun ini diolah melalui perangkat lunak, yang disebut juga dengan program beserta data-data yang diolahnya. Pengeloahan pada *software* ini melibatkan beberapa hal, diantaranya adalah sistem operasi, program,

dan data. *Software* ini mengatur sedemikian rupa sehingga logika yang ada dapat dimengerti oleh mesin komputer.

2.1.2. Klasifikasi Aplikasi

Klasifikasi aplikasi dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi software spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu. Macam-macam data yang digunakan untuk membuat aplikasi adalah:
   a. Data Sumber (source data), ialah fakta yang disimpan di dalam basis data, misalnya: nama, tempat lahir, tanggal lahir, dan lain-lain
   b. Meta Data, digunakan untuk menjelaskan struktur dari basis data, type dan format penyimpanan data item dan berbagai pembatas (constraint) pada data.
   c. Data Dictionary atau Data Repository, digunakan untuk menyimpan informasi katalog skema dan pembatas serta data lain seperti: pembakuan, deskripsi program aplikasi dan informasi pemakai.
   d. Overhead Data, berisi linked list, indeks dan struktur data lain yang digunakan untuk menyajikan relationship record.

Klasifikasi aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

1. Perangkat lunak perusahaan (*enterprise*)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja

4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

2.2 Aplikasi *Mobile*

Aplikasi *mobile* adalah sebuah aplikasi yang memungkinkan Anda melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau *handphone*. Dengan menggunakan aplikasi *mobile*, Anda dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya. Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemari oleh hampir 70% pengguna telepon seluler, karena dengan memanfaatkan adanya fitur game, music player, sampai video player membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun.

2.3. *Handphone*

Telepon seluler (ponsel) atau telepon genggam (telgam) atau handphone (HP) atau disebut pula adalah perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengantelepon konvensional saluran tetap, namun dapat dibawa ke mana-mana (portabel, *mobile*) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (nirkabel; *wireless*). Saat ini Indonesia

mempunyai dua jaringan telepon nirkabel yaitu sistem GSM (*Global System for Mobile Telecommunications*) dan sistem CDMA (*Code Division Multiple Access*).

Handphone atau biasa disebut Telepon Genggam atau yang sering dikenal dengan nama Ponsel merupakan perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon konvensional saluran tetap, namun dapat dibawa ke mana-mana (portabel, *mobile*) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (nirkabel; *wireless*).

2.4 Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java2 adalah generasi kedua dari Java *platform* (generasi awalnya adalah *Java Development Kit*). Java bediri atas sebuah mesin *interpreter* yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam file *.class* dari suatu program sebagai representasi langsung program yang berisai bahasa mesin. Oleh karena itu bahasa java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalakan pada sistem operasi tersebut terdapat JVM.

Platform Java terdiri dari kumpulan *library*, JVM, kelas-kelas *loader* yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah kompiler, *debugger* dan kakas lain yang dipaket dalam *Java Development Kit* (JDK). Java2 adalah generasi yang sekarang sedang berkembang dari *platform* Java. Agar sebuah program Java

dapat dijlankan, maka file dengan ekstensi *.java* harus dikompilasi menjadi file *bytcode.* Untuk menjalankan *bytecode* tersebut dibutuhkan JRE (*Java Runtime Environment*) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi. Platform java mempunyai tiga buah edisi yang bebeda yaitu J2EE (*Java2 Enterprise Edition*), J2ME (*Java2 Micro Edition*), dan J2SE (*Java2 Second Edition*). J2EE adalah kelompok dari beberapa API dari Java dan teknologi selain Java. J2EE dibuat untuk membuat aplikasi yang kompleks. J2EE sering dinggap sebagai *middleware* atau teknologi yang berjalan di *server*, namun sebenarnya J2EE tidak hanya sebatas untuk itu. Faktanya J2EE juga mencakup teknologi yang dapat digunakan di semua lapisan dari sebuah sistem informasi. Implementasi J2EE menyediakan kelas dasar dan API dari Java yang mendukung pengembangan dan rutin standar untuk aplikasi klien maupun *server*, termasuk aplikasi yang berjalan di web *browser*. J2SE adalah lingkungan dasar dari Java.

2.5 *Java2 Micro Edition* (J2ME)

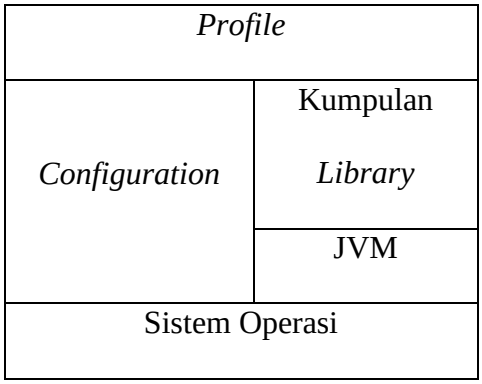
2.5.1 Pengertian

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 sebuah perangkat yang lainnya. J2ME membawa Java ke dunia informasi,

komunikasi, dan perangkat komputasi selain perangkat computer desktop yang biasanya lebih kecil dibandingkan perangkat computer desktop. J2ME biasa digunakan pada telepon seluler, *pager*, PDA dan sejenisnya.

J2ME adalah bagian dari J2SE, karena itu tidak semua library yang ada pada J2SE dapat digunakan pada J2ME. Tetapi J2ME mempunyai beberapa *library* khusus yang tidak dimiliki J2SE. Inti dari J2ME terletak pada *configuration* dan *profile-profile*. Suatu *configuration* menggambarkan lingkungan *runtime* dasar dari suatu sistem J2ME. Ia menggambarkan *core library*, *virtual machine*, fitur keamanan dan jaringan.

2.5.2 Arsitektur J2ME



Gambar 2.1 Arsitektur J2ME

Arsitektur J2ME dapat dilihat pada gambar 2 diatas. J2ME terbagi dalam 2 bagian yaitu:

a. *Configuration*

Configuration merupakan Java *library* minimum dan kapabilitas yang dipunya oleh para pengembang J2ME, yang maksudnya sebuah *mobile device* dengan

kemampuan Java akan dioptimalkan untuk menjadi sesuai. *Configuration* hanya mengatur hal-hal tentang kesamaan sehingga dapat dijadikan ukuran kesesuaian antar-*device.* Dalam J2ME telah didefinisikan dua buah konfigurasi, antara lain:

1. *Connected Limited Device Configuration* (CLDC)

   CLDC atau *Connected Limited Device Configuration* adalah perangkat dasar dari J2ME, spesifikasi dasar berupa *library* dan API yang diimplementasikan pada J2ME, seperti yang digunakan pada telepon seluler, *pager*, dan PDA. Perangkat tersebut dibatasi dengan keterbatasan memori, sumber daya, dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari *package,* kelas, dan sebagian fungsi *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kylobyte Virtual Machine*).

2. *Connected Device Configuration* (CDC)

   CDC atau *Connected Device Configuration* adalah super set dari CLDC. CDC menyediakan lingkungan Java runtime yang lebih luas dibandingkan CLDC dan lebih dekat kepada lingkungan J2SE. CDC terdiri dari *virtual machine* dan kumpulan *library* dasar untuk dipergunakan pada *profile* industri. Implementasi CDC pada J2ME adalah *source code* yang menyediakan sambungan dengan macam-macam platform.

b. *Profile*

Profile membahas sesuatu yang spesifik untuk sebuah perangkat. *Profile* juga menggambarkan set-set tambahan dari API dan fitur untuk pasar tertentu, kategori perangkat atau industri. Sementara *configuration* menggambarkan *library* dasar, *profile-profile* menggambarkan *library* yang penting untuk membuat aplikasi-aplikasi efektif. *Library* ini memasukkan user interface, jaringan dan penyimpanan API. Dalam J2ME terdapat dua buah *profile* yaitu:

1. MIDP atau *Mobile Information Device Profile*. Yaitu *profile* yang menyediakan library Java untuk implementasi dasar antarmuka (GUI), jaringan (*Networking*), basisdata (*database*), penyimpanan tetap (*persistant strage*), daur hidup aplikasi (*Application life cicle*), dan timer (*Personal Information Management*). MIDP dirancang khusus untuk *wireless phone* dan *pager* serta dibangun berdasarkan CLDC. Saat ini terdapat MIDP 1.0 dan MIDP 2.0 . Fitru tambahan MIDP 2.0 dibanding dengan MIDP 1.0 adalah API untuk multimedia. Pada MIDP 2.0 terdapat dukungan memainkan *tone, tone sequence,* dan *file* WAV walaupun tanpa adanya *Mobile Media API (MMAPI).*
2. *Foundation Profile* (FP). Profil dasar untuk non – GUI network *devices* pada CLDC dengan ROM 1024 k dan RAM 512 k.

2.5.3 Verifikasi Class

Spesifikasi CLDC memerlukan semua class untuk melewati proses verifikasi dua tingkat. Verifikasi pertama dilaksanakan diluar perangkat sebelum instalasi pada

perangkat. Verifikasi kedua terjadi pada perangkat selama runtime dan dilaksanakan oleh KVM.

2.5.4. KVM (Kilobyte *Virtual Machine*)

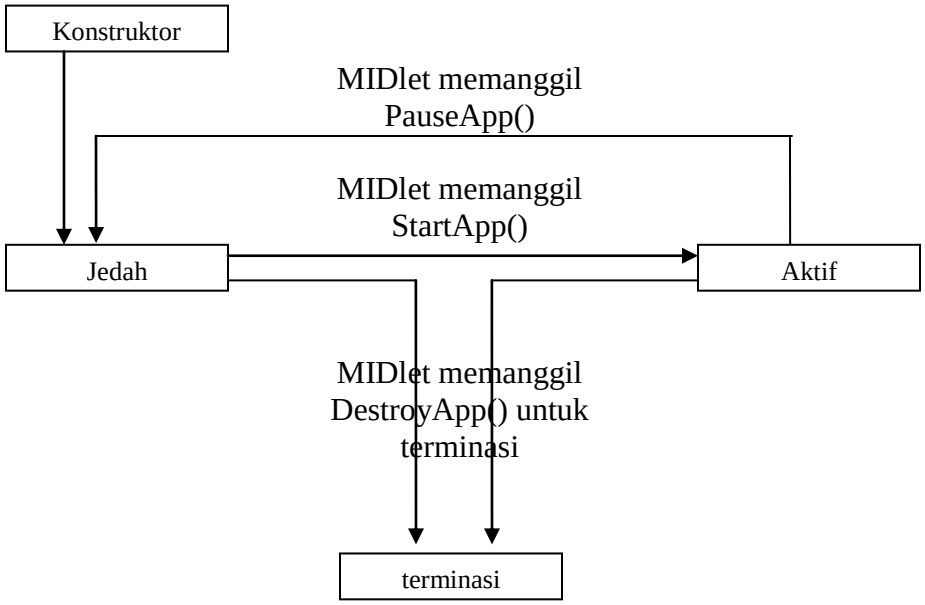
KVM adalah paket JVM yang didesain untuk perangkat yang kecil. KVM mendukung sebagian dari fitur-fitur JVM, seperti misalnya KVM tidak mendukung operasi *floating-point* dan finalisasi objek. KVM diimplemen-tasikan dengan menggunakan C sehingga sangat mudah beradaptasi pada tipe *platform* yang berbeda.

2.5.5. CVM (C-*Virtual Machine*)

CVM adalah paket JVM optimal yang digunakan pada CDC. CVM mempunyai seluruh fitur dari *virtual machine* yang didesain untuk perangkat yang memerlukan fitur-fitur *Java2 virtual machine.*

2.5.6. MIDlet

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari kelas javax.microedition.midlet MIDlet yang didefinisikan pada MIDP. MIDlet berupa sebuah kelas abstrak yang merupakan subkelas dari bentuk dasar aplikasi sehingga antarmuka antara aplikasi J2ME dan aplikasi manajemen pada perangkat terbentuk. Midlet terdiri dari beberapa metode yang harus ada, yaitu *constructor(), protected void startApp() throws MIDletStateChangeException, protected void pauseApp(), protected void destroyApp(Boolean unconditional) throws MIDletStateChangeException.*



Gambar 2.2 Daur Hidup MIDlet

Pada gambar diatas terlihat bahwa ketika MIDlet dijalankan maka akan diinisialisasi dengan kondisi *pause* dan dijalankan *pauseApp(),* Kondisi berikutnya adalah fungsi MIDlet yang dijalanka, yaitu pada *startApp().* Metode yangada tersebut diimplemntasikan sebagai *protected,* hal ini dimaksudkan agar MIDlet lain tidak dapat memanggil metode tersebut. Pada saat pemakai keluar dari MIDlet, maka metode *destroyApp()* akan dijalankan saebelum MIDlet benarbenar tidak berjalan lagi. Metode *notifydestroyed()* akan dipanggil sebelum MIDlet benar-benar tidak berjalan lagi, *destroyApp()* akan memanggil *notifyDestroyed (),* dan *notifyDestroyed ()* akan member tahu *platform* untuk menterminasi MIDlet dan membersihkan semua sumber daya yang mengacu pada MIDlet.

Aplikasi-aplikasi MIDlet dibungkus dan dikirim kedalam perangkat sebagai MIDlet suites. Sebuah MIDlet suite terdiri dari Java Archive (JAR) dan sebuah tambahan Java Application Descriptor (JAD).

2.5.7. JAD (*Java Application Descriptor*)

Digunakan untuk mendiskripsikan isi aplikasi untuk keperluan pemetaan. File JAD berisi deskripsi file JAR (*Java Archive*) dan pemetaan atribut MIDlet, sedangkan file JAR berisi kumpulan kelas dan *resource.*

2.6. *Netbeans*

NetBeans IDE adalah sebuah lingkungan pengembangan terintegrasi, open source. NetBeans IDE mendukung pengembangan. Semua jenis program Java (termasuk JavaFX Java SE (Java ME, web, EJB, kotak masuk atau aplikasi mobile). Dibandingkan dengan fitur-fitur lain, waktu sistem Sistem Indeks REKOMENDASI Ant, Maven refactorings dukungan, menjelaskan versi (mendukung CVS, Subversion, ClearCase dan terus). Fasilitas ini bisa membuat kita melihat program kita pada device yang sesungguhnya. yang di bangun menggunakan plat form Neatbeans.

Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut ‘modul’. Sebuah modul adalah suatu arsip Java (*Java archive*) yang memuat kelas-kelas Java untuk berinetraksi dengan NetBeans Open API dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara

independen, aplikasi berbasis platform NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan *powerful*.

2.7. *UML*

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. (Chonoles, 2003: bab1) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan semantic. Ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. *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 perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sitem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Diagram-diagram *UML* :

Beberapa literature menyebutkan bahwa UML menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat

dikelompokan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.
2. Diagram *Use-Case*. Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan actor-aktor (suatu jenis husus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
3. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
4. Diagram Aktivitas (*Aktivity Diagram*). Bersifat dinamis. Diagram aktivitas adalah tipe husus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.

2.8. Sekolah Menengah Atas

Sekolah menengah atas (disingkat SMA; bahasa Inggris: *Senior High School*), adalah jenjang pendidikan menengah pada pendidikan formal di Indonesia setelah

lulus Sekolah Menengah Pertama (atau sederajat). Sekolah menengah atas ditempuh dalam waktu 3 tahun, mulai dari kelas 10 sampai kelas 12.

Pada tahun kedua (yakni kelas 11), siswa SMA dapat memilih salah satu dari 3 jurusan yang ada, yaitu Sains, Sosial, dan Bahasa. Pada akhir tahun ketiga (yakni kelas 12), siswa diwajibkan mengikuti Ujian Nasional (dahulu Ebtanas) yang memengaruhi kelulusan siswa. Lulusan SMA dapat melanjutkan pendidikan ke perguruan tinggi atau langsung bekerja.

Pelajar SMA umumnya berusia 16-18 tahun. SMA tidak termasuk program wajib belajar pemerintah - yakni SD (atau sederajat) 6 tahun dan SMP (atau sederajat) 3 tahun - maskipun sejak tahun 2005 telah mulai diberlakukan program wajib belajar 12 tahun yang mengikut sertakan SMA di beberapa daerah, contohnya di Kota Yogyakarta dan Kabupaten Bantul.

SMA diselenggarakan oleh pemerintah maupun swasta. Sejak diberlakukannya otonomi daerah pada tahun 2001, pengelolaan SMA negeri di Indonesia yang sebelumnya berada di bawah Departemen Pendidikan Nasional, kini menjadi tanggung jawab pemerintah daerah kabupaten/kota. Sedangkan Departemen Pendidikan Nasional hanya berperan sebagai regulator dalam bidang standar nasional pendidikan. Secara struktural, SMA negeri merupakan unit pelaksana teknis dinas pendidikan kabupaten/kota

2.9. Jepang

Jepang adalah sebuah negara kepulauan di Asia Timur. Letaknya di ujung barat Samudra Pasifik, di sebelah timur Laut Jepang, dan bertetangga dengan Republik Rakyat Cina, Korea, dan Rusia. Pulau-pulau paling utara berada di Laut Okhotsk, dan wilayah paling selatan berupa kelompok pulau-pulau kecil di Laut Cina Timur, tepatnya di sebelah selatan Okinawa yang bertetangga dengan Taiwan.

Menurut mitologi tradisional, Jepang didirikan oleh Kaisar Jimmu pada abad ke-7 SM. Kaisar Jimmu memulai mata rantai monarki Jepang yang tidak terputus hingga kini. Meskipun begitu, sepanjang sejarahnya, untuk kebanyakan masa kekuatan sebenarnya berada di tangan anggota-anggota istana, shogun, pihak militer, dan memasuki zaman modern, di tangan perdana menteri. Menurut Konstitusi Jepang tahun 1947, Jepang adalah negara monarki konstitusional di bawah pimpinan Kaisar Jepang dan Parlemen Jepang.