BAB II

LANDASAN TEORI

2.1 Aplikasi

Merupakan perangkat lunak yang dijalankan oleh para pengguna untuk mendapat suatu tujuan tertentu.

2.1.1 Pengerian Aplikasi

Pengertian Aplikasi menurut Barry Pratama dalam artikel di *http://ilmukomputer.org/category/aplikasi-server/*, yaitu:

1. *Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas.*
2. *Aplikasi adalah sistem lengkap yang mengerjakan tugas spesifik*.
3. *Aplikasi basis data terdiri atas sekumpulan menu, formulir, laporan dan program yang memenuhi kebutuhan suatu fungsional unit bisnis/organisasi/instansi*.

Aplikasi merupakan program siap pakai. Program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi ialah program pemproses kata dan Web Browser. Aplikasi akan menggunakan sistem operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. 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.

Jadi dapat disimpulkan bahwa aplikasi merupakan subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna agar lebih berguna secara optimal.

2.1.2 Karakteristik Aplikasi

Salah satu karakteristik yang menonjol dari mobile user ini selalu dilengkapi oleh perangkat portabel yang memungkinkan bisa terkoneksi dengan internet atau bisa terhubung dengan pengguna lainnya. Perangkat-perangkat tersebut pada umumnya berbentuk kecil bisa berupa notebook/netbook, smartphone, PDA, tablet PC, atau mobile devices yang lain. Mereka menggunakan perangkat tersebut untuk melakukan aktivitas bisnis, belajar-mengajar, interaksi sosial ataupun sekedar untuk hiburan.Selain oleh perangkat portabel, untuk melakukan berbagai macam aktifitasnya tentu pengguna mobile ini menggunakan aplikasi yang terinstall di perangkat tersebut.

2.1.3 Komponen Aplikasi

Fungsionalitas aplikasi dapat dikustomisasi secara mudah dan diperkaya dengan arsitektur komponennya yang fleksibel. Aplikasi mengatur satu set komponen aplikasi, masing-masing mengimplementasi fitur tertentu. Dengan mengkonfigurasi properti komponen aplikasi, kita bisa mengkustomisasi kelasi dan nilai properti setiap komponen aplikasi yang dipakai dalam sebuah aplikasi.

2.2 Aplikasi *Mobile*

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.

2.2.1 Pengertian Aplikasi *Mobile*

Aplikasi *Mobile* merupakan sebuah aplikasi yang memungkinkan untuk melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau *Handphone*. Dengan menggunakan aplikasi *mobile,* User dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing* dan lain sebagainya.

2.2.2 Karakteristik Perangkat *Mobile*

Karakteristik yang menonjol dari perangkat mobile, ukuran yang kecil, memory yang terbatas, daya proses yang terbatas, mengkonsumsi daya yang rendah, kuat dan dapat diandalkan, konektivitas yang terbatas, dan masa hidup yang pendek.

2.3 *HandPhone*

Dari wikipedia bahasa Indonesia dijelaskan bahwa telepon selular atau *handphone* merupakan alat 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). Selain berfungsi untuk melakukan dan

menerima panggilan telepon, ponsel umumnya juga mempunyai fungsi pengiriman dan penerimaan pesan singkat *Short Message Service* (SMS). Mengikuti perkembangan teknologi, kini ponsel dilengkapi dengan berbagai pilihan fitur, seperti bisa menangkap siaran televisi, perangkat lunak pemutar audio dan video, kamera digital, game, dan layanan internet. Ada pula penyedia jasa telepon genggam di beberapa negara yang menyediakan layanan generasi ketiga atau 3G dengan menambahkan jasa videophone, sebagai alat pembayaran, maupun untuk televisi online pada telepon genggam mereka. Sekarang, telepon genggam menjadi gadget yang multifungsi. Selain fitur-fitur tersebut, ponsel sekarang sudah ditanamkan fitur komputer. Jadi pada ponsel tersebut, orang bisa mengubah fungsi ponsel tersebut menjadi mini komputer. Di dunia pendidikan, fitur ini sangat membantu bagi para pelajar untuk melakukan pembelajaran di satu alat.

2.4 Java

Merupakan sebuah teknologi yang berkembang sangat pesat akhir-akhir ini. Bahkan belakangan ini dikabarkan berusaha mengalahkan Microsoft yang terkenal sebagai kampiun dari produsen operating system dimuka bumi ini

2.4.1 Pengertian Java

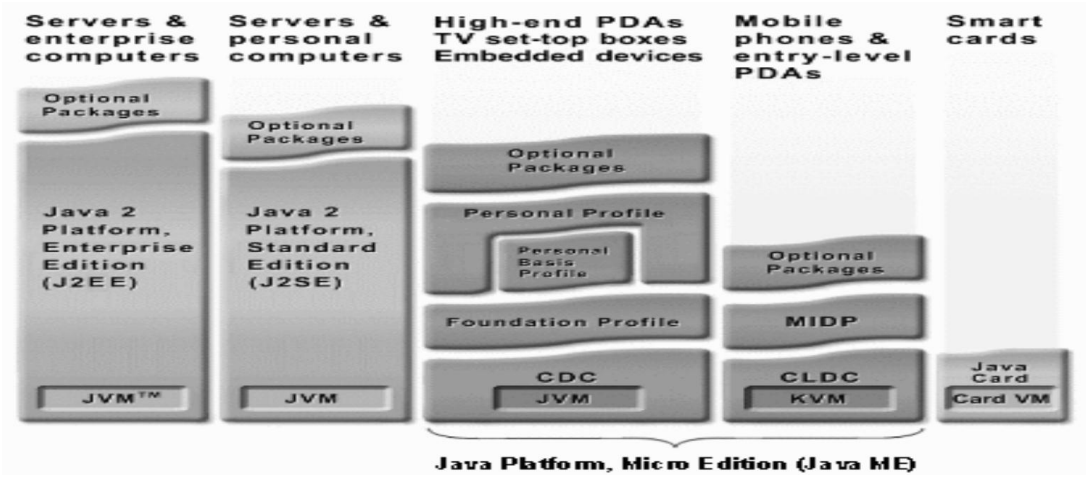
Java menurut definisi Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java2 adalah generasi kedua dari Java *platform* (generasi awalnya adalah *Java Development Kit*). Java berdiri diatas sebuah mesin *interpreter*

yang di beri nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam file **.class** dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu, bahasa Java disebut sebagai bahasa pemrogramaan yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan sistem operasi tersebut terdapat JVM.

Platform Java terdiri dari kumpulan *library*, JVM, kelas-kelas *loader* yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah *compiler*, *debuger*, dan perkakas lain yang dipaketkan dalam *Java Development Kit* (JDK). Java2 adalah generasi sekarang yang sedang berkembang dari *platform* Java. Agar sebuah program Java dapat dijalankan, maka file dengan eksistensi .**java**harus dikompilasi menjadi file *bytecode*. Untuk menjalankan *bytecode* tersebut dibutuhkan *Java Runtime Environment* (JRE) yang memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan *library* Java yang digunakan.

Platform Java memiliki tiga buah edisi yang berbeda, 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 dianggap sebagai *middleware* atau teknologi yang berjalan di *server*, namun sebenarnya J2EE tidak hanya terbatas untuk itu. 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 standar untuk

aplikasi *client* maupun *server*, termasuk aplikasi yang berjalan di *web browser*. J2SE adalah lingkungan dasar dari Java. Ruang lingkup keterhubungan J2EE, J2SE, dan J2ME.



2.1 Gambar Keterhubungan J2SE, J2EE dan J2ME

Sumber : Jurnal Aplikasi Mobile Learning Organizer Mahasiswa

2.4.2 J2ME

Menurut Candra (2010:32), J2ME adalah platform-independen dan membuat penciptaan mobile, aplikasi berbasis internet.

Menurut sarwosri (2010:184), Java ME adalah lingkungan pengembangan yang didesain untuk menggunakan aplikasi java pada peralatan elektronik kecil, seperti telepon seluler, PDA, dan sejenisnya. Java ME dibuat untuk mengatasi keterbatasan yang berhubungan dengan pembuatan aplikasi pada peralatan elektronik kecil

J2ME merupakan satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterei, layar yang kecil dan bandwith jaringan yang rendah.

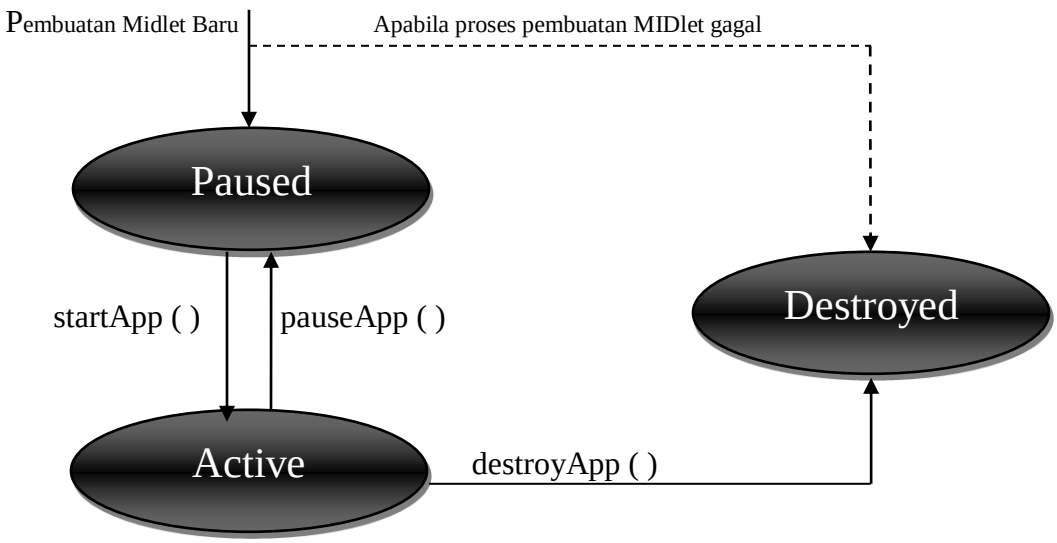
Karateristik dari perangkat mobile, dan bagaimana hal tersebut mempengaruhi pengembangan program untuk perangkat ini. Kita akan diperkenalkan kepada Java 2 Mobile Edition (J2ME), termasuk pentingnya configuration dan profilenya.

2.4.3 MIDlet

MIDlet merupakan bagian dari package javax.microedition.midlet. Sebuah MIDlet harus di-extend dengan class MIDlet. Dan dapat meminta parameter dari AMS seperti dirumuskan dalam application descriptor (JAD). Suatu MIDlet tidak harus memiliki (dan memang harus tidak mempunyai) sebuah method public static void main(String[] argv). Method tersebut tidak akan dikenal lagi oleh AMS sebagai titik awal sebuah program.

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). File JAD adalah suatu file teks yang berisi satu set atribut-atribut, beberapa dibutuhkan. Terdapat Software yang Digunakan Dalam mobile Programming.

Setiap MIDlet dapat berada dalam salah satu keadaan (state) berikut: **Pause**, **Active**, maupun **Destroyed**.



Gambar 2.2 Siklus Hidup MIDlet

Sumber : Tuntunan Pemograman JAVA untuk Handphone

Pada gambar di atas bahwa pada saat pembuatan MIDlet baru, mula-mula MIDlet akan berada dalam keadaan Paused. Apabila proses pembuatan MIDlet gagal atau mengakibatkan kesalahan (menimbulkan eksepsi). Maka MIDlet akan langsung berada dalam keadaan Destroyed. Namun apabila proses pembuatan MIDlet berjalan dengan baik, maka setelah MIDlet dijalankan, maka AMS secara otomatis akan mengeksekusi method startApp( ) dan hal ini akan mengubah MIDlet untuk dalam keadaaan Active. MIDlet yang dalam keadaan active dapat diubah kembali menjadi keadaan Pause melalui pemanggilan methode pauseApp( ) atau diubah menjadi keadaan Destroyed melalui pemanggilan methode destroyedApp( ).

2.4.3.1 Method startApp()

AMS akan memanggil method startApp() untuk memerintahkan MIDlet agar memperoleh fokus dan menjadikan MIDlet berada dalam keadaan Active. Hal ini

terjadi ketika MIDlet baru saja dibuat atau MIDlet yang akan kembali diaktifkan dari keadaan paused. Bentuk umum deklarasi method startApp() adalah sebagai berikut:

public static void startApp()

2.4.3.2 Method pausApp()

AMS akan memanggil method pauseApp() untuk memerintahkan MIDlet agar tidak memiliki fokus dan akan menjadikan MIDlet berada dalam keadaan paused. Dalam keadaan ini, aplikasi tidak dapat memiliki satu pun tampilan UI (User Interface). Apabila aplikasi yang yang dibuat mengandung objek *Thread* maupun *Timer*, maka objek-objek tersebut tidak akan dihentikan secara otomatis. Artinya, harus dihentikan secara manual melalui penulisan kode. Aplikasi akan kembali berada dalam keadaan Active bila diaktivasi ulang. Bentuk umum deklarasi method pauseApp() adalah sebagai berikut:

public static void pauseApp()

2.4.3.3 Method destroyApp()

AMS memanggil method destroyApp() untuk memerintahkan MIDlet agar membuang atau membebaskan semua resource (biasanya berupa file) yang digunakan sekaligus menutup dan menghentikan aplikasi sesegera mungkin. Ini berarti harus menutup semua stream yang masih terbuka serta menghentikan semua thread dan timer yang digunakan. Pemanggilan method destroyApp() akan mengakibatkan MIDlet berada dalam keadaan Destroyed sehingga pada saat tersebut tidak dapat lagi melakukan pengaksesan terhadap obek Display. Bentuk umum deklarasi methd destroyApp() adalah sebagai berikut:

public static void destroyApp(boolean unconditional)

2.4.4 Connected Limited Device Configuration (CLDC)

Didalam buku Tuntunan Pemograman java unuk Handphone CLDC adalah sebuah konfigurasi yang terdapat didalam J2ME untuk alat-alat yang memiliki keterbatasan ruang memori atau RAM (kurang dari 512KB) dan pada umumnya dioperasikan dengan menggunakan baterai, serta memiliki bandwith yang kecil

2.4.5 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.

2.4.6 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 Hewan

Hewan merupakan makhluk hidup yang sangat banyak jumlahnya dan tersebar diseluruh penjuru dunia. Di setiap negara terdapat bermacam-macam jenis hewan baik hewan darat, laut dan udara.

2.5.1 Pengertian Hewan

Hewan atau disebut juga dengan binatang merupakan kelompok organisme yang diklasifikasikan dalam kerajaan Animalia atau metazoa, adalah salah satu dari berbagai makhluk hidup di bumi. Sebutan lainnya adalah fauna dan margasatwa (atau satwa saja).

Hewan dalam pengertian sistematika modern mencakup hanya kelompok ber sel banyak (multiselular) dan terorganisasi dalam fungsi-fungsi yang berbeda (jaringan), sehingga kelompok ini disebut juga hstozoa. Semua binatang heterotrof, artinya tidak membuat energi sendiri, tetapi harus mengambil dari lingkungan sekitarnya.
Penggolongan Hewan Berdasarkan Tempat Hidupnya. Berdasarkan tempat hidupnya, hewan digolongkan menjadi :
Hewan darat, Hewan air, dan Hewan darat dan air.

2.5.2 Hewan Darat

Hewan darat merupakan hewan yang tinggal dan hidup di daratan.

2.5.3 Hewan Air

Hewan air merupakan hewan yang hidup dan bertempat tinggal di dalam air.

2.5.4 Hewan Darat dan Air

Hewan darat dan air merupakan hewan yang tempat tinggalnya ada di dua tempat hidup, yaitu di air dan darat.

Penggolongan Hewan Berdasarkan Jenis Makanan yang Dimakannya yang terdiri dari tiga jenis yaitu; Herbivora, Karnivora dan Omnivora

2.5.5 Herbivora / Herbifora

Menurut Kendo (2010:27), *Cattle feeding on hay and silage from troughs did their biting and chewing separately, while in grazing, the biting occurs with the chewing, especially on swards of short grass*.

Herbivora merupakan jenis hewan yang memakan tumbuh-tumbuhan seperti daun, kayu, biji, buah, bunga, rumputm, dan lain sebagainya. Contoh Binatang herbivor adalah Domba, Jerapah, Sapi, Kelinci, dan Kuda.

2.5.6 Carnivora / Karnivora

Carnivora merupakan jenis hewan yang memakan daging, darah, dan sebagainya. Hewan pemburu ini disebut juga sebagai hewan predator. Contoh hewan carnifora adalah singa, macan, harimau, cheetah, burung bangkai, dan buaya,.

2.5.7 Omnivora / Omnifora

Omnivora merupakan jenis hewan yang memakan tumbuhan maupun hewan. Binatang ini makan silih berganti antara keduanya. Contoh binatang omnivora adalah yakni tikus, ikan mas, dan ayam,.

2.6 Sekolah Dasar

Sekolah dasar (disingkat SD) merupakan jenjang paling dasar pada pendidikan formal di Indonesia. Sekolah dasar ditempuh dalam waktu 6 tahun, mulai dari kelas 1 sampai kelas 6. Lulusan sekolah dasar dapat melanjutkan pendidikan ke Sekolah Menengah Pertama (atau sederajat). Tingkat satuan pendidikan yang dianggap sebagai dasar pendidikan. Disekolah inilah anak- anak didik mengalami sebuah

proses pendidikan dan pembelajaran. Pendidikan ini diselenggarakan untuk anak-anak yang telah berusia enam tahun dengan asumsi bahwa anak tersebut sudah mempunyai pemahaman dan kebutuhan pendidikan yang sesuai. Pendidikan dasar diselenggarakan untuk memberikan dasar pengetahuan, sikap dan keterampilan bagi anak didik. Pendidikan dasar inilah yang selanjutnya dikembangkan untuk meningkatkan kualitas anak.

2.7 UML

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. (chonoles, 2003: bab 1) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Intinya, UML merupakan alat komunikasi yang konsisten dalam mensuport para pengembang sistem saat ini. Sebagai perancang sistem, mau tidak mau pasti akan menjumpai UML, baik kita sendiri yang membuat atau sekedar membaca diagram UML buatan orang lain (Pilone, 2005: bab1), dalam buku Menggunakan UML.

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standard. untuk merancang model sebuah sistem. Ada beberapa diagram yang disediakan dalam UML antara lain:

1. Diagram *Use Case*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan

“bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara actor dengan sistem.

(Pooley, 2003:15) mengatakan bahwa model use case dapat dijabarkan dalam diagram use case, tetapi yang perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram, sumber buku Menggunkan UML.

Konsep dasar pemodelan *use case* meliputi : *use case,* actor, relasi, diagram aktivitas dan diagram *use case.* Aktor merupakan bagian dari *use case* yang bertindak sebagai subjek (pelaku) dalam suatu proses.

2. Diagram Aktivitas

Diagram aktivitas atau *Activity diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis. Diagram ini dapat digunakan untuk menggammbarkan aliran kerja (*floe of events*) dalam *use case.*

3. Diagram Sekuensial

Diagram sekuensial atau *sequence diagram* digunakan untuk menggambarkan aliran fungsionalitas dalam use case. Diagram ini disusun berdasarkan urutan waktu.

4. Diagram Kelas

Diagram kelas atau *class diagram* menunjukkan interaksi antar kelas dalam sistem. Diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem dan relasi antar mereka. Diagram ini menunjukkan gambaran sistem secara statis. Satu diagram kelas menampilkan *subset* dari kelaskelas dan relasinya.

5. Component Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

6. Deployment diagram

Menggambarkan struktur fisik kode dari komponent. Komponent dapat berupa *source code*, komponent biner, atau *executable component*. Sebuah komponent berisi informasi tentang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.