BAB II

LANDASAN TEORI

Pada Bab II akan menjelaskan beberapa konsep yang berhubungan dengan permasalahan di dalam penelitian yang dimana akan digunakan sebagai dasar pemahaman dalam mengimplementasikan konsep-konsep kedalam semua kegiatan pembangunan sebuah aplikasi

2.1 Pengertian Aplikasi Mobile

2.1.1 Pengertian Aplikasi

Menurut Buyens (2001) aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas. Dari pengertian diatas dapat disimpulkan bahwa aplikasi adalah sebuah program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu.

Software aplikasi digolongkan berdasarkan beberapa jenis, berikut jenis – jenis software aplikasi :

1. Packaged Software : Software yang telah memiliki Copyright yang biasa dibutuhkan oleh user pribadi ataupun perusahaan. Contoh : Ms. Office 2007, Adobe Photoshop, dll

2. Custom Software : Software yang memiliki fungsi tertentu. Biasanya software ini dipesan oleh suatu perusahaan kepada programmer dengan fungsi yang biasa digunakan di perusahaan tersebut. contoh : Aplikasi Penerimaan siswa baru, dll
3. Web-based Software : software yang ada di website. pengguna atau user dapat menggunakannya dari komputer manapun yang terhubung ke Internet. Contoh : e-mail, web browser game , dll
4. Open Source software : software yang dapa kita gunakan, modifikasi dan mendistribusikannya ulang.
5. Shareware : software yang memiliki copyright dan didistribusikan secara gratis dengan periode tertentu.
6. Freeware : software yang memiliki copyright dan gratis dapat kita pakai tapi kita tidak dapat memodifikasinya seperti open source.
7. Public – domain software : software yang memiliki copyright dan di donasikan untuk publik serta tidak dibatasi penggunaanya. Setiap orang dapat mengcopy dan mendistribusikannya ke orang lain. Contoh : GNU/Linux.

Macam-macam data yang digunakan untuk membuat aplikasi adalah:

1. Data Sumber (source data), ialah fakta yang disimpan di dalam basis data, misalnya: nama, tempat lahir, tanggal lahir, dan lain-lain
2. Meta Data, digunakan untuk menjelaskan struktur dari basis data, type dan format penyimpanan data item dan berbagai pembatas (constraint) pada data.
3. Data Dictionary atau Data Repository, digunakan untuk menyimpan informasi katalog skema dan pembatas serta data lain seperti: pembakuan, deskripsi program aplikasi dan informasi pemakai.
4. Overhead Data, berisi linked list, indeks dan struktur data lain yang digunakan untuk menyajikan relationship record.

2.1.2 Pengertian Mobile

Mobile berasal dari bahasa inggris yang artinya berpindah. Mobile dapat di artikan sebagai perpindahan dari suatu tempat ke tempat yang lain.

Pada konsep ini, mobile lebih cenderung dengan aplikasi yang dapat digunakan kapanpun dan dimanapun dengan menggunakan perangkat mobile seperti telepon seluler, pager, PDA, smartphine dan sejenisnya.

2.1.3 Pengertian Aplikasi Mobile

Menurut Buyens (2001) aplikasi mobile berasal dari kata application dan mobile. Application yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan mobile dapat di artikan sebagai perpindahan dari suatu tempat ke tempat yang lain.

Maka aplikasi mobile dapat di artikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah – pindah dari satu tempat ke tempat yang lain serta mempunyai ukuran yang kecil. Aplikasi mobile ini dapat di akses melalui perangkat nirkabel, pager, PDA, telepon seluler, smartphone, dan perangkat sejenisnya.

Menurut Budi Darytamo dkk (2007) pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman konvensional pada PC. Aspek karakteristik dari perangkat bergerak sering mempengaruhi 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.

2.2 Karakteristik Perangkat Mobile

Perangkat mobile memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi mereka memiliki karakteristik yang sangat berbeda dari sistem desktop.
Berikut karakteristik perangkat mobile, diantaranya :

a. Ukuran yang kecil

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

b. Memory yang terbatas

Perangkat mobile juga memiliki memory 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 jumlah dari memory, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.

c. 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, Anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.

d. 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-baterai.

e. 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.

f. Konektivitas yang terbatas

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

g. Masa hidup yang pendek

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. Coba ambil kasus sebuah handphone, mereka booting dalam hitungan detik dan kebanyakan orang tidak mematikan handphone mereka bahkan ketika malam hari. PDA akan menyala jika anda menekan tombol power mereka.

Aplikasi mobile terdapat beberapa tipe, berikut beberapa tipe aplikasi mobile :

A. Berdasarkan Teknologi yang digunakan :

1. Server-Request Based

   Mobile Equipment (ME) merequest data dari server, WAP. menggunakan WML, sebuah markup language khusus berbasis XML.

   Keuntungan :

   a. Platform Independent
   b. Mudah diterapkan
   c. Jenis aplikasi Beragam

   Kerugian :

   a. Masih banyak ME yang tidak support WAP.
   b. Pembayaran untuk data.
   c. Interface terbatas

   Implementasi :

   a. Menggunakan Web Server dengan setting tambahan di configurasi
   b. File WML dibuat selayaknya membuat file HTML

2. SMS based

   SMS : Short Message Service. Pengiriman pesan singkat antar ME / dengan sebuah server Terbatas ± 160 characters.

   Keuntungan :

   a. Platform Independent
   b. ME sebagian besar sudah dapat menerapkan SMS

   Kerugian :

   a. Pembayaran untuk setiap aksi.
   b. Interface hanya text, user harus ingat perintah
   c. Panjang text sangat terbatas

   Implementasi :

   a. Menggunakan interface antara ME dengan server
   b. Aplikasi menunggu data dari interface dan mengirim data melalui interface tersebut

3. SIM-card based

   Aplikasi diletakkan di atas SIM-card, aplikasi dapat diakses melalui menu baru yang akan muncul di ponsel. Javacard, khusus untuk javacard aplikasi bersifat downloadable, pengiriman data lewat SMS

   Keuntungan :

   a. Platform Independent
   b. Data disimpan secara lokal dan dapat digunakan bila berganti ME

Kerugian :

a. Size terbatas.
b. Aplikasi harus sangat kecil
c. Sulit diterapkan

Implementasi :

a. Download Javacard.
b. Gunakan Simulator

4. Mobile Equipment Based

Aplikasi diletakkan di Mobile Equipment, contoh Pemrograman di atas Symbian OS, pengiriman data SMS.

Keuntungan :

a. Interface sangat mudah
b. Jenis aplikasi sangat beragam, bahkandapat digunakan untuk aplikasi berat

Kerugian :

a. Platform Dependent
b. Bila mengubah ME kemungkinan aplikasi dan data tidak dapat digunakan

B. Berdasarkan fungsi :

1. Information

   Pengiriman informasi ke ME user

2. Communications

   Komunikasi antara user dengan satu atau lebih user lain.

3. Entertainment

   Aplikasi menghibur seperti Games.

4. Productivity Tools

   Aplikasi yang dapat meningkatkan produktivitas walaupun sedang berpindah

5. Commerce and Banking

   Jual beli melalui ME.

6. Location based

   Aplikasi yang mengubah data sesuai dengan lokasi user. Lokasi dapat ditentukan oleh GPS dan sel yang ditempati oleh user.

2.3 Pengertian M - Learning

Istilah M – Learning mengacu pada penggunaan perangkat teknologi informasi genggam dan bergerak seperti PDA, Hp, Laptop dan Tablet PC dalam pengajaran dan pembelajaran. M – Learning adalah pembelajaran yang unik karena pembelajaran, arahan dan aplikasi yang berkaitan dengan pembelajaran dapat diakses kapanpun dan dimanapun. Hal ini akan meningkatkan perhatian pada materi pembelajaran, membuat pembelajaran menjadi pervasif dan dapat mendorong motivasi belajar.

Beberapa kelebihan M – Learning dibandungkan pembelajaran lain adalah :

a. Dapat digunakan kapanpun dan dimanapun
b. Kebanyakan perangkat bergerak memiliki harga yang relatif lebih murah dibandingkan dengan harga PC atau Laptop
c. Ukuran perangkat yang kecil dan ringan

Meski memiliki kelebihan, akan tetapi M – Learning juga memiliki kekurangan diantaranya sebagai berikut :

a. Kemampuan prosessor
b. Kapasitas Memori
c. Layar Tampilan
d. Kemampuan Sumber Daya Listrik
e. Perangkat I / O

2.4 Bahasa Pemograman JAVA

2.4.1 Sejarah Singkat 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 oleh 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 di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan 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, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "bapak java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak 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.

2.4.2 Edisi Java

Menurut M. Shalahuddin (2008) Java adalah bahasa yang dapat di jalankan dimanapun dan disembarang platform apapun, diberagam lingkungan : internet, intranet, consumer Electronic products dan computer Applications. The Java 2 platform tersedia dalam 3 edisi untuk keperluan berbeda. Untuk beragam aplikasi yang dibuat dengan bahasa java, java dipaketkan dalam edisi – edisi berikut :

1. Java 2 Standard Edition (J2SE)

Menurut M. Shalahuddin (2008) The Java 2 Standard Edition menyediakan lingkungan pengembangan yang kaya akan fitur, stabil, aman dan cross – platform. Edisi ini mendukung konektivitas basis data, rancangan antar muka pemakai, masukan / keluaran dan pemograman jaringan dan termasuk sebagai paket – paket dasar bahasa Java.

2. Java 2 Enterprise Edition (J2EE)

Menurut M. Shalahuddin (2008) The Java Enterprise Edition menyediakan tool untuk membangun dan menjalankan multitier enterprise applications. J2EE bersisi paket – paket di J2SE ditambah paket – paket untuk mendukung pengembangan Enterprise JavaBeeans, Java Servlets, Java Server Pages, XML dan kendali transaksi yang fleksibel

3. Java 2 Micro Edition (J2ME)

Menurut M. Shalahuddin (2008) The Java 2 Micro Edition selain menyedikan bahasa Java yang sama, unggul dalam portabilitas (kemampuan dapat dijalankan dimanapun), *safe network delivery*, seperti J2SE dan J2EE. Aplikasi-aplikasi dapat diskalakan (dimampukan) agar dapat bekerja dengan J2SE dan J2EE.

J2ME untuk beragam consumer electronic product seperti pager, smart card, cell phone, handled PDA, dan set top box.

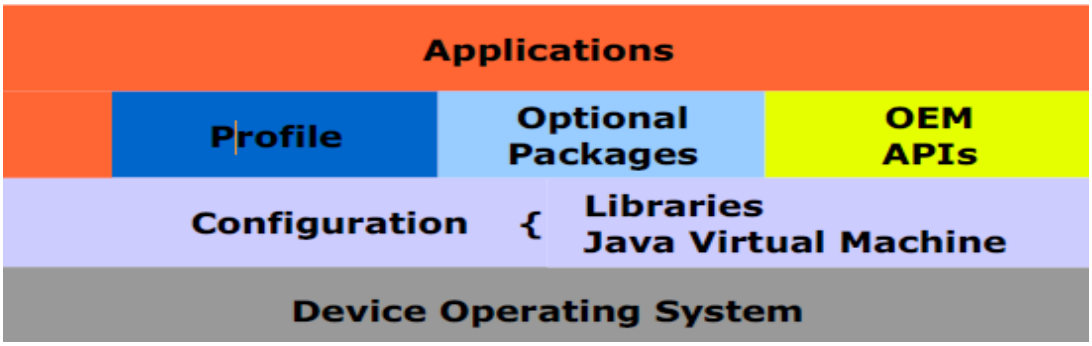
2.5 Gambaran J2ME

Menurut M. Shalahuddin (2008) J2ME adalah 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.

Dengan perkembangbiakan perangkat mobile konsumer dari telepon, PDA, kotak permainan ke peralatan-peralatan rumah, Java menyediakan suatu lingkungan yang portable untuk mengembangkan dan menjalankan aplikasi pada perangkat ini. Program J2ME, seperti semua program JAVA adalah diterjemahkan oleh VM. Program-program tersebut dikompile ke dalam bytecode dan diterjemahkan denga Java Virtual Machine (JVM). Ini berarti bahwa program-program tersebut tidak berhubungan langsung dengan perangkat.

J2ME menyediakan suatu interface yang sesuai dengan perangkat.

Aplikasi-aplikasi tersebut tidak harus dikompile ulang supaya mampu dijalankan pada mesin yang berbeda. 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.



Gambar 2.1 Arsitektur J2ME

Sebuah profile memberikan library tambahan untuk suatu kelas tertentu pada sebuah perangkat. profile-profile menyediakan user interface (UI) API,persistence, messaging library, dan sebagainya.

Satu set library tambahan atau package tambahan menyediakan kemampuan program tambahan. Pemasukan package ini ke dalam perangkat J2ME dapat berubah-ubah karena tergantung pada kemampuan sebuah perangkat. Sebagai contoh, beberapa perangkat MIDP tidak memiliki Bluetooth built-in, sehingga Bluetooth API tidak disediakan dalam perangkat ini

Arsitektur J2ME dapat dilihat pada gambar 1 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*.

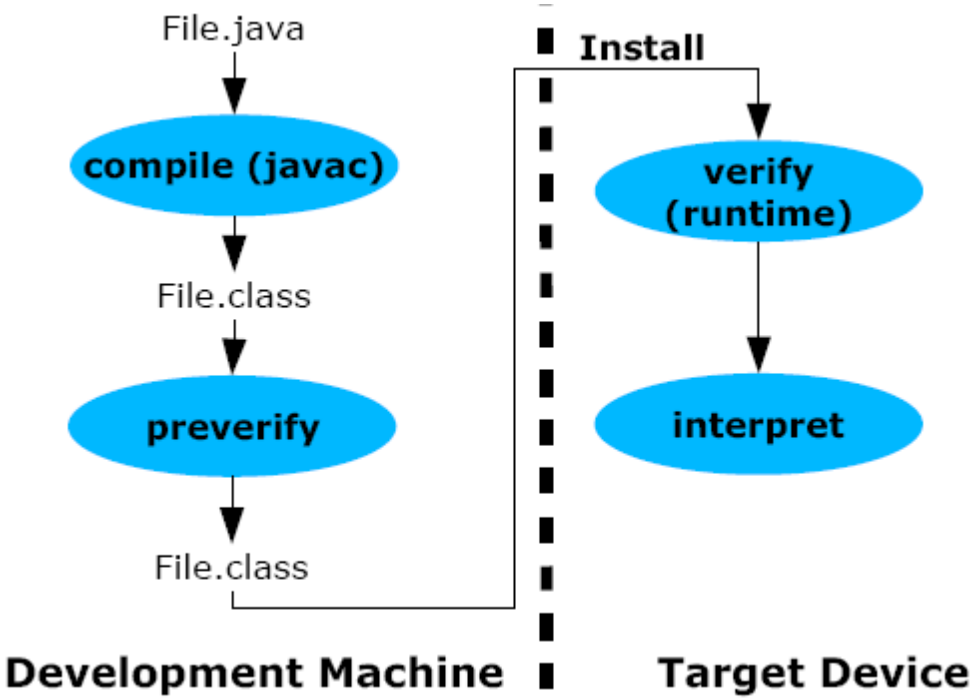
Yaitu *profile* yang menyediakan library Java untuk implementasi dasar antarmuka (GUI), jaringan (*Networking*), basisdata (*database*), penyimpanan tetap (*persistant storage*), daur hidup aplikasi (*Application life cicle*), dan timer (*Personal Information Management*). MIDP dirancang khusus untuk *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.

2.5.1 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, seperti ditunjukkan gambar 2.



Gambar 2.2 Proses Verifikasi Dua Tingkat

2.5.2 KVM (Kilobyte *Virtual Machine*)

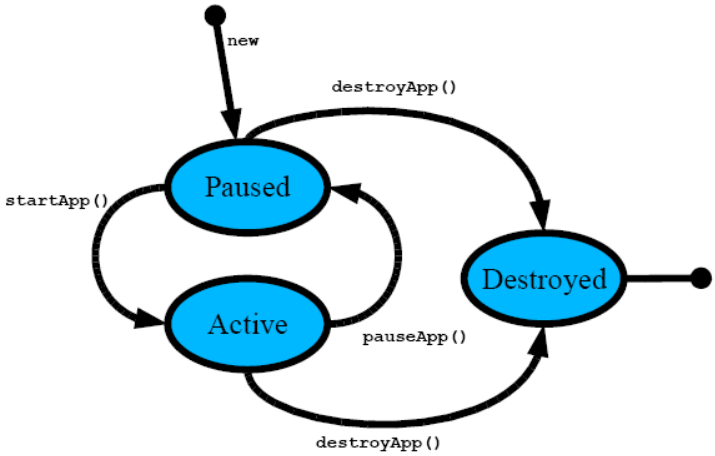
KVM adalah paket JVM yang didesain untuk perangkat yang kecil. KVM mendukung sebagian dari fitur-fitur JVM, seperti misalnya KVM tidak mendukung operasi *floating-point* dan finalisasi objek. KVM diimplemen-tasikan dengan menggunakan C sehingga sangat mudah beradaptasi pada tipe *platform* yang berbeda.

2.5.3 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.*

2.5.4 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 2.3 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 yang ada 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).

2.5.5 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.*

2.5.6 *MMAPI (Mobile Media Application Programming Interface)*

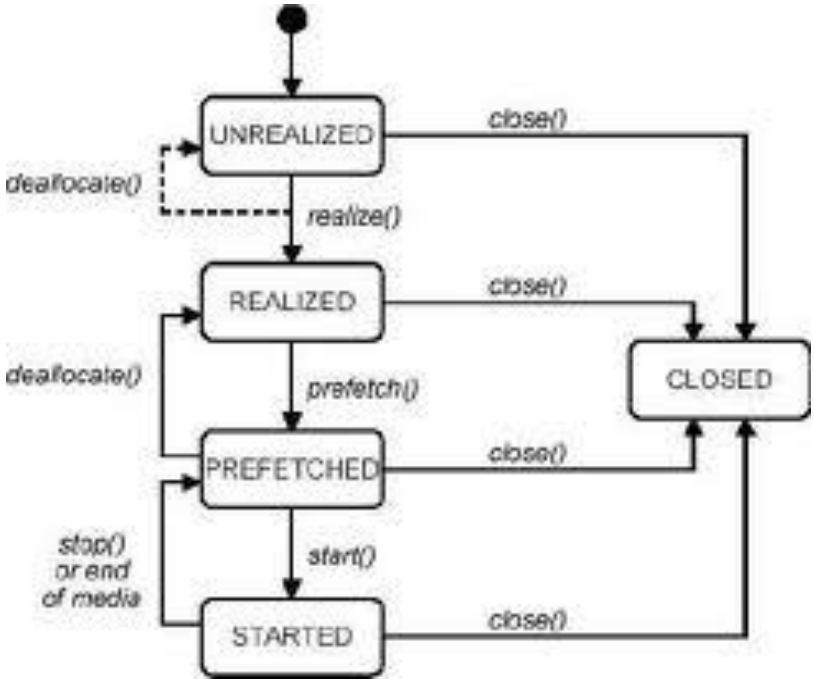
Pada J2ME, profil MMAPI memperluas cakupan kearah pengembangan mobile multimedia. MMAPI menyediakan sekumpulan interfaceyang mendukung pemutaranberbagai tipe media. Didalam MIDP 2.0 terdapat subset spesifikasi MMAPI yang dibatasi. MMAPI mendefinisikankelas dan interface dalam package. MMAPI memberikan dukungan lebih pada midlet dalam penggunaan media seperti Audio dan Video.



Gambar 2.4 Hirarki Objek Pada MMAPI

Terdapat dua kelas utama dalam hirarki MMAPI. Kelas tersebut adalah sebagai berikut :

a. Kelas Manager merupakan salah satu kelas factory yang menyediakan method untuk mendapatkan objek player dan melakukan query terhadap protokol dan tipe content yang didukung oleh perangkat.
b. Kelas player menangani playback suatu content



Gambar 2.5 Model Player State

2.6 Konsep Perancangan Berorientasi Objek

2.6.1 Pengenalan UML

Menurut Prabowo Pudjo Widodo & Herlawati (2011) UML (Unified Modeling Language) merupakan pengganti dari metode analisis berorientasi object dan design berorientasi object (OOA & OOD) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. UML merupakan gabungan dari metode Grady Booch (Booch Method), James Rumbaugh (OMT) dan Ivar Jacobson (OOSE). Tetapi UML ini akan mencakup lebih luas daripada OOA&D. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.

Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka Anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain. UML merupakan bahasa standar untuk penulisan Blueprint Software yang digunakan untuk Visualisasi (*Visualize*), Spesifikasi (*Specify*), Pembentukan (*Construct*) dan Pendokumentasian (*Documentation*) alat-alat dari sistem perangkat lunak.

2.6.2 Sejarah UML

UML dimulai secara resmi pada oktober 1994, ketika Rumbaugh bergabung dengan Booch pada Relational Software Corporation. Proyek ini memfokuskan pada penyatuan metode Booch dan OMT. UML versi 0.8 merupakan metode penyatuan yang dirilis pada bulan Oktober 1995. Dalam waktu yang sama, Jacobson bergabung dengan Relational dan cakupan dari UML semakin luas sampai diluar perusahaan OOSE. Dokumentasi UML versi 0.9 akhirnya dirilis pada bulan Juni 1996. Meskipun pada tahun 1996 ini melihat dan menerima feedback dari komunitas Software Engineering . Dalam waktu tersebut, menjadi lebih jelas bahwa beberapa organisasi perangkat lunak melihat UML sebagai strategi dari bisnisnya. Kemudian dibangunlah UML Consortium dengan beberapa organisasi yang akan menyumbangkan sumber dayanya untuk bekerja, mengembangkan, dan melengkapi UML.

Di sini beberapa partner yang berkontribusi pada UML 1.0, diantaranya Digital Equipment Corporation, Hewlett-Packard, I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Relational, Texas Instruments dan Unisys. Dari kolaborasi ini dihasilkan UML 1.0 yang merupakan bahasa pemodelan yang ditetapkan secara baik, expressive, kuat, dan cocok untuk lingkungan masalah yang luas. UML 1.0 ditawarkan menjadi standarisasi dari Object Management Group (OMG). Dan pada Januari 1997 dijadikan sebagai standar bahasa pemodelan.

Antara Januari–Juli 1997 gabungan group tersebut memperluas kontribusinya sebagai hasil respon dari OMG dengan memasukkan Adersen Consulting, Ericsson, ObjectTimeLimeted, Platinum Technology, Ptech, Reich Technologies, Softeam, Sterling Software dan Taskon. Revisi dari versi UML (versi 1.1) ditawarkan kepada OMG sebagai standarisasi pada bulan Juli 1997.

Dan pada bulan September 1997, versi ini dierima oleh OMG Analysis dan Design Task Force (ADTF) dan OMG ArchitectureBoard. Dan Akhirnya pada Juli 1997 UML versi 1.1 menjadi standarisasi.

Pemeliharaan UML terus dipegang oleh OMG Revision Task Force (RTF) yang dipimpin oleh Cris Kobryn. RTP merilis editorial dari UML 1.2 pada Juni 1998. Dan pada tahun 1998 RTF juga merilis UML 1.3 disertai dengan user guide dan memberikan technical cleanup.

2.6.3 Pengertian UML

Menurut Prabowo Pudjo Widodo & Herlawati (2011) UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object.

UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corp. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.6.4 Diagram UML

Untuk merancang model sebuah sistem. Ada beberapa diagram yang disediakan dalam UML antara lain:

2.6.4.1 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 aktor dengan sistem. 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. *Use case* adalah proses-proses yang terjadi dalam suatu *software*. *Use case* juga menggambarkan apa yang sedang dilakukan oleh seorang *actor*. Relasi menggambarkan hubungan antara *actor* dan *use case*.

2.6.4.2 Diagram Aktifitas

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 menggambarkan aliran kerja (*floe of events*) dalam *use case.*

2.6.4.3 Diagram Sekuensial

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

2.6.4.4 Diagram Kelas

Diagram kelas atau *class diagram* menunjukkan interaksi antar objek 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 kelas - kelas dan relasinya.

2.6.4.5 Diagram Deployment

Diagram deployment bagian – bagian perangkat lunak suatu sistem ke perangkat keras yang akan mengeksekusinya. Elemen – elemen perangkat lunak seperti komponen, kelas, paket dan sebagainya dimanifestasikan menggunakan artifak serta dipetakan ke perangkat keras yang akan menjalankannya dengan nodes.

2.7 Netbeans IDE 7.0.1

Pengembangan NetBeans diawali dari Xelfi, sebuah proyek mahasiswa tahun 1997 di bawah bimbingan Fakultas Matematika dan Fisika Universitas Charles, Praha. Sebuah perusahaan kemudian dibentuk untuk proyek tersebut dan menghasilkan versi komersial NetBeans IDE hingga kemudian dibeli oleh Sun Microsystem pada tahun 1999. Sun kemudian menjadikan NetBeans open source pada bulan Juni tahun 2000. Sejak itu komunitas NetBeans terus berkembang. Platform NetBeans adalah framework yang dapat digunakan kembali (*reusable*) untuk menyederhanakan pengembangan aplikasi desktop. Ketika aplikasi berbasis platform NetBeans dijalankan, kelas Main dari platform dieksekusi. Modul-modul yang tersedia ditempatkan di sebuah *registry* di dalam memori, dan tugas *startup* modul dijalankan. Secara umum, kode modul dimuatkan ke dalam memori hanya ketika ia diperlukan.

NetBeans IDE adalah IDE open source yang ditulis sepenuhnya dengan bahasa pemrograman Java menggunakan platform NetBeans. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (J2SE, web, EJB, dan aplikasi *mobile*). Fitur lainnya adalah sistem proyek berbasis Ant, kontrol versi, dan *refactoring*. *NetBeans Mobility Pack* adalah alat untuk mengembangkan aplikasi yang berjalan pada perangkat bergerak (*mobile*), umumnya telepon seluler, tetapi juga mencakup PDA, dan lain-lain.

NetBeans Mobility Pack digunakan untuk menulis, menguji, dan *debugging* aplikasi untuk perangkat bergerak yang menggunakan teknologi berplatform Java Micro Edition (platform Java ME). Paket ini mengintegrasikan dukungan terhadap *Mobile Information Device Profile* (MIDP) 2.0, *Connected Limited Device Configuration* (CLDC) 1.1, dan *Connected Device Configuration (CDC). Emulator dari pihak ketiga dapat diintegrasikan dengan mudah untuk lingkungan pengujian yang lebih kokoh. NetBeans Mobility Pack* saat ini tersedia dalam dua klaster yang berbeda, yang satu memuat *CDC* dan yang lainnya *CLDC.*

2.8 Konsep Dasar Doa

2.8.1 Pengertian Doa

Doa adalah memohon atau meminta suatu yang bersifat baik kepada Allah SWT seperti meminta keselamatan hidup, rizki yang halal dan keteguhan iman. Sebaiknya kita berdoa kepada Allah SWT setiap saat karena akan selalu didengar olehNya.

Berdoa yang secara etimologis berarti "meminta kepada Allah", mempunyai tujuan-tujuan yang bukan saja bersifat ukhrawi, melainkan juga bersifat duniawi. karena doa bukanlah untuk kepentingan Allah melainkan untuk kepentingan manusia itu sendiri. Kalaupun kita berdoa untuk memohon segala "sesuatu yang kita butuhkan", "yang kita inginkan" ataupun hanya "untuk menenangkan diri dari segala kesusahan", namun doa mempunyai beberapa faidah yang tak terhingga.

2.8.2 Manfaat Doa

Manfaat doa sangatlah banyak, berikut salahsatu manfaat doa diantaranya :

a. Doa berfungsi untuk menunjukkan keagungan Allah swt kepada hamba - hambaNya yang lemah. Dengan doa seorang hamba menyadari bahwa hanya Allah yang memberinya nikmat, menerima taubat, yang memperkenankan doa-doanya.

b. Doa mengajarkan kita agar merasa malu kepada Allah. Sebab manakala ia tahu bahwa Allah akan mengabulkan doa-doanya, maka tentu saja ia malu untuk mengingkari nikmat-nikmatNya.

   Bahkan manakala manusia sudah berada dalam puncak keimanan yang kuat sekalipun, maka ia akan lebih dekat lagi (taqarrub) untuk mensyukuri nikmatNya

c. Doa mengajarkan kita agar selalu ingat kepada Allah SWT serta meningkatkan keimanan seseorang

2.9 Konsep Dasar Pembelajaran

Menurut Dimyati dan Mudjiono (Sagala, 2005) mengemukakan bahwa belajar merupakan tindakan dan perilaku siswa yang kompleks, sebagai tindakan belajar hanya dialami oleh siswa sendiri

Istilah pembelajaran berhubungan erat dengan pengertian belajar dan mengajar. Belajar, mengajar dan pembelajaran terjadi bersama-sama. Belajar dapat terjadi tanpa guru atau tanpa kegiatan mengajar dan pembelajaran formal lain. Sedangkan mengajar meliputi segala hal yang guru lakukan di dalam kelas.

Pada jaman sekarang proses pembelajaran tidak harus terjadi di dalam kelas atau pun di lingkungan sekolah / instansi pendidikan. Seiring perkembangan teknologi dan informasi yang begitu cepat, proses pembelajaran pun dapat memanfaatkan teknologi yang ada sekarang seperti melalui internet sampai ke perangkat mobile yang kita pegang seperti smartphone, tablet, dan sejensinya.