

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Dakon**

##### **2.1.1 Pengertian Dakon**

Banyak permainan tradisional Indonesia yang sebenarnya secara tidak langsung memberikan pelajaran. Dakon atau juga biasa disebut congklak adalah salah satu permainan tradisional. Permainan ini memberikan beberapa pelajaran atau melatih anak-anak antara lain: strategi, kesabaran, dan ketelitian.

**Strategi:** dakon menuntut pemain memikirkan pemilihan agar bisa memenangkan permainan.

**Kesabaran:** pemain khususnya yang tidak sedang bermain atau melangkah harus bersabar menunggu lawannya melakukan kesalahan sehingga tiba gilirannya, pemain yang sedang bermain juga harus bersabar memasukkan satu-persatu biji-bijian dalam lubang.

**Ketelitian:** pemain yang sedang bermain harus teliti dalam memasukkan biji dakon satu per satu dalam lubang, sedangkan pemain yang sedang tidak bermain atau melangkah juga harus teliti mengawasi atau memastikan biji-bijian dimasukkan satu persatu dalam lubang jangan sampai lawan melakukan kecurangan. Dengan begitu dakon melatih motorik sekaligus sensorik.

### 2.1.2 Perlengkapan Bermain Dakon

Dakon memerlukan perlengkapan antara lain media dakon yang biasanya terbuat dari kayu berbentuk persegi panjang dengan lubang yang berjajar berpasangan dan diujung kanan dan kiri lubang tersebut terdapat satu lubang yang ukurannya lebih besar sebagai rumah masing-masing pemain. (lihat ilustrasi)



Gambar 2.1 Dakon Tradisional

Media dakon selain dari kayu yang diukir bisa juga dibuat sendiri dengan membuat lubang seperti pada gambar di tanah, sehingga mudah untuk bermainnya, selain tanpa biaya juga akan menambah kreatifitas. Selain media tersebut dibutuhkan juga biji-bijian untuk mengisi lubang-lubang tersebut. Jumlah biji-bijian disesuaikan dengan jumlah lubang, minimal masing-masing lubang berisi 5 biji. Biji-bijian bisa menggunakan kerikil atau biji buah sirsat, sawo atau biji-bijian lain yang ukurannya kecil.

### 2.1.3 Cara Bermain Dakon

Cara memainkan dakon:

1. Dakon dimainkan oleh dua orang.
2. Biji-bijian dimasukkan dalam tiap lubang kecuali lubang diujung (lubang paling besar yang disebut rumah bagi masing-masing pemain).
3. Ditentukan pemain yang akan bermain lebih dulu (biasanya dengan cara sut).
4. Pemain yang lebih dulu main memilih biji-bijian di salah satu lubang pada sisi pemain tersebut.
5. Biji-biji dimasukkan satu persatu dalam lubang searah dengan lubang besar yang menjadi rumahnya sampai habis.
6. Apabila biji-bijian habis pada rumahnya maka lawan memilih biji-bijian pada lubang disisinya lawan, tapi apabila tidak habis pada rumahnya maka biji-bijian akan berjalan terus adalah biji-bijian dalam lubang dimana biji sampai terakhir jatuh, begitu seterusnya.
7. Apabila biji terakhir jatuh di lubang kosong maka permainan dilanjutkan oleh lawan.
8. Ketika biji terakhir jatuh di lubang yang kosong maka ada dua cara, yang pertama apabila lubang itu ada disisi pemain yang sedang bermain (melangkah) maka biji-bijian yang berada di lubang pasangannya (disisi lawan) menjadi miliknya dan dimasukkan semua ke lubang besar (rumahnya), yang kedua apabila lubang kosong itu ada disisi lawan maka pemain tidak mendapatkan apa-apa.

9. Permainan dilanjutkan seperti itu terus menerus sampai biji-bijian dalam lubang masuk semua dalam rumah masing-masing pemain dan lubang-lubang kecil disisi masing-masing pemain kosong semua. Bila lubang-lubang disisi masing-masing pemain sudah kosong semua maka permainan berakhir dan dilakukan perhitungan untuk menentukan siapa pemenangnya.
10. Untuk menentukan pemenang, dihitung jumlah biji-bijian di masing-masing rumah (lubang besar) pemain. Jumlah biji yang terbanyak adalah pemenangnya.
11. Permainan bisa diulang kembali seterusnya sampai bosan.<sup>1</sup>

## **2.2. Java**

### **2.2.1 Bahasa Pemrograman Java**

Java adalah bahasa pemrograman yang berorientasi objek (OOP) yang dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat open source.

Perkembangan teknologi Java semakin pesat. Hingga saat ini Sun Microsystems membagi Java menjadi empat jenis edisi:

---

<sup>1</sup> <http://istanaide.blogspot.com/2010/04/dakon.html>, diakses 20 April 2011

1. Java Card

Teknologi java yang digunakan pada peralatan elektronik yang memiliki memori terbatas, misalnya smart card.

2. J2SE (Standard Edition)

Teknologi java edisi standar, digunakan untuk penerapan teknologi java pada komputer desktop.

3. J2EE (Enterprise Edition)

Teknologi java edisi enterprise, digunakan untuk penerapan teknologi java pada komputer server.

Untuk aplikasi web.

4. J2ME (Java 2 Micro Edition)

Teknologi java edisi mikro, digunakan untuk penerapan teknologi java pada peralatan elektronik seperti handphone dan PDA.

## 2.2.2 Teknologi Java

### 1. Sebuah Bahasa Pemrograman

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, *desktop*, *web* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan Java tidak hanya terfokus ada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

## 2. Sebuah *Development Environment*

Sebagai sebuah peralatan pembangun, teknologi Java menyediakan banyak *tools* : *compiler*, *interpreter*, penyusun dokumentasi, paket kelas dan sebagainya.

## 3. Sebuah Aplikasi

Aplikasi dengan teknologi Java secara umum adalah aplikasi serba guna yang dapat dijalankan pada seluruh mesin yang memiliki *Java Runtime Environment* (JRE).

## 4. Sebuah *Deployment Environment*

Terdapat dua komponen utama dari *Deployment Environment*. Yang pertama adalah JRE, yang terdapat pada paket J2SDK, mengandung kelas – kelas untuk semua paket teknologi Java yang meliputi kelas dasar dari Java, komponen GUI dan sebagainya. Komponen yang lain terdapat pada Web Browser. Hampir seluruh Web Browser komersial menyediakan *interpreter* dan *runtime environment* dari teknologi Java.

### 2.2.3 Keuntungan JAVA

Berdasarkan *white paper* resmi dari SUN, Java memiliki karakteristik berikut:

#### 1. Sederhana (*Simple*)

Bahasa pemrograman Java menggunakan Sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance*. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection*.

## 2. Berorientasi objek (Object Oriented)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

## 3. Terdistribusi (Distributed)

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries networking* yang terintegrasi pada Java.

## 4. Interpreted

Program Java dijalankan menggunakan interpreter yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi *Java bytecodes* dapat dijalankan pada platform yang berbeda-beda.

## 5. Robust

Java mempunyai reliabilitas yang tinggi. Compiler pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling* untuk membantu mengatasi error pada pemrograman.

## 6. Secure

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

#### 7. Architecture Neutral

Program Java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform berbeda dengan *Java Virtual Machine*.

#### 8. Portable

*Source code* maupun program Java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.

#### 9. Performance

Performance pada Java sering dikatakan kurang tinggi. Namun performance Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan *Just In Time Compilers (JIT)*.

#### 10. Multithreaded

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

#### 11. Dynamic

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan *properties* ataupun *method* dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.



## 2.2.4 Sebagian Fitur dari Java

### 1. Java Virtual Machine (JVM)

JVM adalah sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi hardware dan platform dimana kompilasi kode Java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis Java menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh JVM.

Aplikasi program Java diciptakan dengan *file* teks berekstensi *.java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi.

### 2. Garbage Collection

Banyak bahasa pemrograman lain yang mengizinkan seorang pemrogram mengalokasikan memori pada saat dijalankan. Namun, setelah menggunakan alokasi memori tersebut, harus terdapat cara untuk menempatkan kembali blok memori tersebut supaya program lain dapat menggunakannya. Dalam C, C++ dan bahasa lainnya, adalah pemrogram yang mutlak bertanggung jawab akan hal ini. Hal ini dapat menyulitkan bilamana pemrogram tersebut lupa untuk mengembalikan blok memori sehingga menyebabkan situasi yang dikenal dengan nama *memory leaks*.

Program Java melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek – objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh pemrogram dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat pada bahasa yang memungkinkan alokasi dinamis.

### 3. Code Security

*Code Security* terimplementasi pada Java melalui penggunaan Java Runtime Environment (JRE). Java menggunakan model pengamanan 3 lapis untuk melindungi sistem dari *untrusted Java Code.*, berupa :

1. Pertama, *class-loader* menangani pemuatan kelas Java ke *runtime interpreter*. Proses ini menyediakan pengamanan dengan memisahkan kelas – kelas yang berasal dari *local disk* dengan kelas – kelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas – kelas yang berasal dari *local disk* yang dimuat terlebih dahulu.
2. Kedua, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan – aturan dasar bahasa Java.
3. Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem file, *port* jaringan, proses eksternal dan sistem *windowing*.

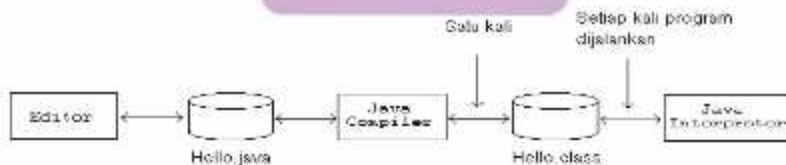
Setelah seluruh proses tersebut selesai dijalankan, barulah kode program di eksekusi.

Java juga menyediakan beragam teknik pengamanan lain :

1. Bahasa dirancang untuk mempersulit eksekusi kode perusak. Peniadaan *pointer* merupakan langkah besar pengamanan. Java tidak mengenal operasi *pointer*. Di tangan pemrogram handal, operasi *pointer* merupakan hal yang luar biasa untuk optimasi dan pembuatan program yang efisien serta mengagumkan. Namun mode ini dapat menjadi petaka di hadapan pemrogram jahat. *Pointer* merupakan sarana luar biasa untuk pengaksesan tak diotorisasi. Dengan peniadaan operasi *pointer*, Java dapat menjadi bahasa yang lebih aman.
2. Java memiliki beberapa pengamanan terhadap *applet*. Untuk mencegah program bertindak mengganggu media penyimpanan, maka *applet* tidak diperbolehkan melakukan *open*, *read* ataupun *write* terhadap berkas secara sembarangan. Karena Java *applet* dapat membuka jendela *browser* yang baru, maka jendela mempunyai logo Java dan teks identifikasi terhadap jendela yang dibuka. Hal ini mencegah jendela *pop-up* menipu sebagai permintaan keterangan *username* dan *password*.

### 2.2.5 Fase – fase Pemrograman Java

Gambar ini menjelaskan aliran proses kompilasi dan eksekusi sebuah program Java :



Gambar 2.2 Proses Kompilasi dan Eksekusi Program Java

Langkah pertama dalam pembuatan sebuah program berbasis Java adalah menuliskan kode program pada *text editor*. Contoh *text editor* yang dapat digunakan antara lain : notepad, vi, emacs dan lain sebagainya. Kode program yang dibuat kemudian tersimpan dalam sebuah berkas berekstensi *java*. Setelah membuat dan menyimpan kode program, kompilasi file yang berisi kode program tersebut dengan menggunakan Java Compiler. Hasil dari adalah berupa berkas *bytecode* dengan ekstensi *.class*. Berkas yang mengandung *bytecode* tersebut kemudian akan dikonversikan oleh Java Interpreter menjadi bahasa mesin sesuai dengan jenis dan *platform* yang digunakan.

Tabel 2.1 Proses Platform<sup>2</sup>

<i>Proses</i>	<i>Tool</i>	<i>Hasil</i>
Menulis kode program	<i>Text editor</i>	Berkas berekstensi <i>java</i>
Kompilasi program	Java Compiler	Berkas berekstensi <i>.class</i> (Java Bytecodes)
Menjalankan program	Java Interpreter	Program Output

### 2.2.6 Netbeans IDE 6.7.1

NetBeans adalah IDE (*Integrated Development Enterprise*) open source yang ditulis sepenuhnya dengan bahasa pemrograman Java. NetBeans 6.7.1 mendukung pengembangan semua tipe aplikasi Java yang dibutuhkan untuk membuat aplikasi dekstop profesional, enterprise, web dan aplikasi *mobile* dengan bahasa pemrograman

<sup>2</sup> [www.google.com](http://www.google.com) , Pengenalan Bahasa Java, diakses 20 Oktober 2009

Java, C/C++ dan even dynamic languages seperti PHP, JavaScript, Groovy dan Ruby. NetBeans 6.7.1 dapat dijalankan di beberapa platform termasuk Windows, Linux, Mac OS dan Solaris.

### 2.2.7 Java 2 Micro Edition

Java 2 Micro Edition memiliki arsitektur yang unik, dibangun dengan desain modular yang memungkinkan aplikasi yang dibuat dapat disesuaikan dengan keterbatasan perangkat telepon seluler. J2ME ini terdiri dari lapisan-lapisan yang terdapat di atas lapisan sistem operasi, dimana secara keseluruhan lapisan disebut CLDC (Connected Limited Device Configuration).<sup>3</sup>

#### 2.2.7.1 Configuration dan Profile

Dalam pengembangan aplikasi wireless dengan Java, J2ME dibagi menjadi dua buah bagian diantaranya ialah bagian *configuration* dan *profile*.

1. Lapisan Konfigurasi (*Configuration Layer*)

J2ME mempunyai dua konfigurasi yaitu *Connected Limited Device Configuration (CLDC)* dan *Connected Device Configuration (CDC)*.

2. Lapisan Profil (*Profile Layer*)

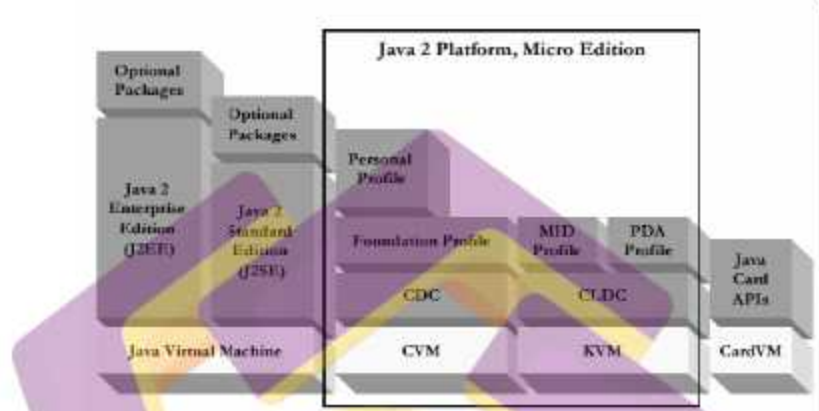
J2ME mempunyai beberapa profil antara lain :

- a. *Mobile Information Device Profile (MIDP)*
- b. *Foundation Profile (FP)*
- c. *Personal Profile*

---

<sup>3</sup> Setianto, Dwi, 2006, Panduan Lengkap Pemrograman Celluler Device dengan J2ME, Yogyakarta, Ardana Media.

d. *Personal Digital Assistance (PDA)*



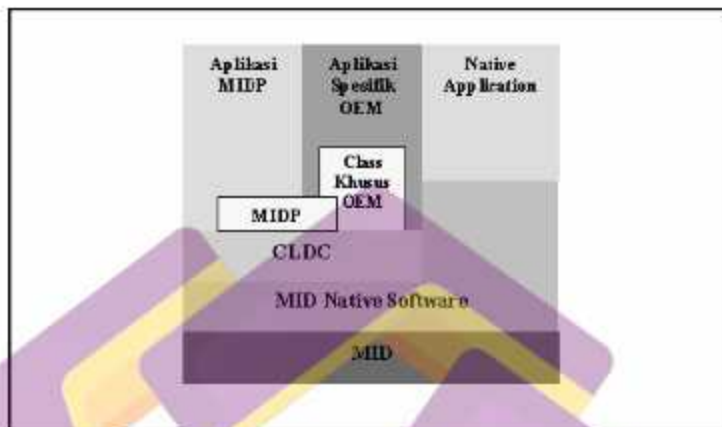
Gambar 2.3 Lapisan Konfigurasi<sup>4</sup>

Dilihat dari Gambar diatas maka J2ME mempunyai lapisan konfigurasi dan profile yang didukung oleh *Java Virtual Machine (C-Virtual Machine dan K-Virtual Machine)*.

### 2.2.7.2 MIDlet

Aplikasi yang berjalan pada sebuah perangkat yang mendukung MIDP disebut dengan MIDlet. Dengan kata lain MIDlet merupakan aplikasi dari MIDP (*Mobile Information Device Profile*).

<sup>4</sup> Knudsen Jonathan, *Wireless Java—Developing with J2ME*, Second Edition, USA, 2003

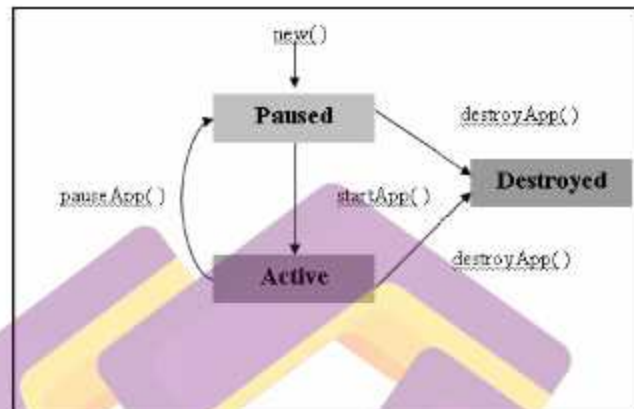


Gambar 2.4 .Arsitektur Aplikasi MIDP<sup>5</sup>

*Lifecycle* dari sebuah *MIDlet* ditangani oleh *Application Management Software (AMS)*<sup>6</sup>. *AMS* adalah sebuah lingkungan tempat siklus dari sebuah *MIDlet*, mampu untuk diciptakan, dijalankan, dihentikan maupun dihilangkan. *AMS* sering pula disebut dengan *Java Application Manager (JAM)*. *MIDlet* memiliki beberapa *state*, yaitu *Pause*, *Active* dan *Destroy*. Ketika masing-masing *state* dipanggil, beberapa *method* yang bersesuaian dipanggil. *Method-method* tersebut merupakan bawaan dari *J2ME*. Untuk menjelaskan proses *MIDlet* dalam *Java Application Manager (JAM)* adalah sebagai berikut :

<sup>5</sup> Knudsen Jonathan, *Wireless Java—Developing with J2ME*, Second Edition. USA, 2003

<sup>6</sup> Day Bill, *Days Codecamp: Developing Wireless Applications using MIDP 2.0, WMA, and MMA*, java.sun.com diakses 4 september 2011

Gambar 2.5 LifeCycle MIDlet<sup>7</sup>

### 2.3. Ponsel

Ponsel saat ini memang bukan barang yang aneh untuk masyarakat Indonesia. Dulu Ponsel digunakan untuk menelpon dan mengirim pesan saja tapi kini, ponsel bukan lagi sekadar alat untuk berkomunikasi saja, tetapi bisa untuk menyimpan data, musik, foto, *chatting*, dan lain sebagainya. Semakin berkembangnya jaman banyak teknologi pendukung yang terintegrasi dengan produk Ponsel, seperti radio FM, kamera digital dan pemutar MP3.

Dengan berkembangnya teknologi Ponsel, Ponsel tidak lagi hanya digunakan untuk alat komunikasi tapi dengan fasilitas teknologi tertentu Ponsel bisa ditambah dengan aplikasi lain.

<sup>7</sup> Knudsen Jonathan, *Wireless Java—Developing with J2ME*, Second Edition. USA, 2003



### 2.3.1 Sistem Operasi pada Ponsel

Ponsel makin cerdas saja. Itu karena urusan “otak”-nya juga mulai diperhatikan. Beberapa jenis sistem operasi siap dicangkokkan ke ponsel-ponsel keluaran terbaru. Dalam kurun waktu beberapa tahun saja, kecanggihan dan popularitas ponsel sudah mengalahkan gadget manapun yang pernah diciptakan. Berbagai aplikasi seru yang bisa dibuat dalam bentuk supermini telah dicangkokkan ke dalam handset yang satu ini. Alhasil, selain menjalankan fungsi dasarnya sebagai alat komunikasi, ponsel juga jago berperan sebagai organizer, kamera, music player, sampai komputer. Hebatnya, semua fitur canggih tersebut berhasil dibungkus dalam desain yang imut sekaligus fashion, dua elemen penting yang digila-gilai masyarakat saat ini. Untuk menjalankan berbagai aplikasi yang dibebankan kepadanya, ponsel mesti mengadopsi teknologi ala komputer yaitu sistem operasi atau biasa disebut OS (Operating System). Jadi jangan heran kalau Windows atau Linux kini sudah wara-wiri dalam wujud handset.

Selain berfungsi untuk mengkoordinir sumber daya hardware dan software ponsel seperti keypad, layar, phonebook, baterai, dan koneksi ke jaringan, OS juga memastikan agar semua aplikasi bisa berjalan stabil dan konsisten. Kalau ponsel Anda sering hang atau lelet kinerjanya, ini bisa jadi karena sistem operasinya yang kurang mantap. OS juga mesti dirancang cukup fleksibel supaya para software developer alias pembuat software bisa lebih gampang menciptakan aplikasi-aplikasi baru yang lebih canggih. Inilah asyiknya punya ponsel ber-OS. Kita bisa mendownload berbagai aplikasi tambahan yang kita butuhkan ke dalamnya. Tidak mesti

pasrah dengan fasilitas yang sudah disediakan oleh vendor. Aplikasi-aplikasi tambahan ini bisa diperoleh secara gratis dari si vendor pembuat produk tersebut, atau membeli dari pihak ketiga seperti operator atau para software developer. Saat ini ada tiga sistem operasi yang mewarnai kancah perponselan yaitu Symbian OS, Windows Mobile, dan Linux. Selain itu, masih ada OS yang diberi julukan proprietary. Sistem ini dirancang oleh masing-masing vendor dan tidak bisa dimodifikasi oleh si pengguna ponsel. Misalnya Nokia OS pada beberapa ponsel Nokia seperti seri 6230, 6610 dan 6822. Hingga saat ini, dibandingkan OS proprietary dari vendor lain, rancangan Nokia dianggap paling user friendly karena tidak ribet ketika dioperasikan.

#### **2.4. Perangkat Mobile**

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

##### **1. Ukuran yang kecil**

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

##### **2. Memory yang terbatas**

Perangkat mobile juga memiliki 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.

### 3. Daya proses yang terbatas

Sistem mobile tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti harddisk dan RAM, Anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.

### 4. Mengonsumsi 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.

### 5. Kuat dan dapat diandalkan

Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

### 6. Konektivitas yang terbatas

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

### 7. 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.

## **2.5. Unified Modeling Language (UML)**

Unified Modeling Language (UML) merupakan notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain system perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP). Dengan kata lain UML merupakan standar bahasa modeling yang dapat memodelkan sistem yang konkret dan terdistribusi. UML sudah menjadi standar industri yang dibuat dibawah pengawasan Object Management Group (OMG). OMG dibentuk untuk membuat standar-standar software yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar CORBA (Comman Object Request Broker Architecture).

Pada dunia software engineering, UML merupakan suatu metode modeling dan bahasa spesifikasi yang sifatnya non-proprietary. Sebenarnya penggunaan dari UML itu sendiri tidak terbatas hanya pada dunia software modeling, tetapi bisa pula digunakan untuk modeling hardware (engineering systems) dan sering digunakan sebagai model untuk proses bisnis dan juga modeling untuk struktur organisasi. UML adalah suatu metode terbuka yang digunakan untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifak-artifak dari suatu pengembangan sistem software yang berbasis pada obyek. UML merupakan hasil kompilasi best

engineering practice yang sudah terbukti sukses dalam pemodelan sistem yang besar, sistem yang kompleks, khususnya pada level arsitektural.

Ada tiga aspek utama dalam pemodelan sistem yang mampu didukung oleh UML:

- a. Functional Model, untuk menunjukkan fungsionalitas dari suatu sistem dari sudut pandang user atau pengguna. Ini dicapai dengan menggunakan Use Case Diagram.
- b. Object Model, untuk menunjukkan struktur dan substruktur dari suatu sistem dengan menggunakan object, atribut, operasi dan juga asosiasi. Ini dicapai dengan menggunakan Class Diagram.
- c. Dynamic Model, menunjukkan internal behavior dan suatu sistem. Ini dicapai dengan menggunakan Sequence Diagram, Activity Diagram dan juga Statechart Diagram.

Merupakan hal yang penting untuk membedakan model UML dan diagram UML, atau himpunan diagram, termasuk Use Case Diagram, Collaboration Diagram, Activity Diagram, Sequence Diagram, Deployment Diagram, Component Diagram, Class Diagram, StateChart Diagram. Suatu UML diagram adalah representasi grafis dari suatu informasi dalam suatu model, tetapi model tersebut hadir secara independen.

UML dalam posisinya saat ini, membedakan notasi dan meta-model. Notasi tersebut adalah grafik yang merupakan sintaks grafis dari bahasa pemodelan.

Contohnya, notasi class diagram menentukan bagaimana pokok-pokok dan konsep, seperti class, association, dan multiplicity, digambarkan.

UML bukan merupakan metode dan juga tidak berusaha untuk menjadi suatu metode, namun demikian ia merupakan kebutuhan untuk Rational Unified Process yang telah dibuat oleh Rational Software Corporation. Metode development lainnya seperti Dynamic Systems Development Method juga menggunakan teknik dari UML.

