
Perkenalan PBO (Java)

I. Pengenalan Java

Java adalah suatu teknologi di dunia software komputer, yang merupakan suatu bahasa pemrograman tingkat tinggi Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++, Java juga merupakan bahasa pemrograman berorientasi objek dan dirancang agar dapat dijalankan di semua platform.

II. Jenis – Jenis Java

- **Enterprise Java (J2EE)**

Dipergunakan untuk aplikasi berbasis web, aplikasi sistem tersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi.

- **Standar Java (J2SE)**

Dipergunakan untuk aplikasi berbasis desktop dan inilah yang biasa dikenal sebagai bahasa Java.

- **Micro Java (J2ME)**

Merupakan subset dari J2SE dan salah satu aplikasinya yang banyak dipakai adalah untuk wireless device / mobile device.

III. Fitur - Fitur Java

a) **Applet**

Program Java yang dapat berjalan di atas browser, yang dapat membuat halaman HTML lebih dinamis dan menarik.

b) **Java Networking**

Sekumpulan API (Application Programming Interface) yang menyediakan fungsi – fungsi untuk aplikasi – aplikasi jaringan, seperti penyediaan akses untuk TCP, UDP, IP Address dan URL. Tetapi Java Networking tidak menyediakan akses untuk ICMP dikarenakan alasan sekuriti dan pada kondisi umum hanya administrator (root) yang bisa memanfaatkan protokol ICMP.

c) Java Database Connectivity (JDBC)

JDBC menyediakan sekumpulan API yang dapat digunakan untuk mengakses database seperti Oracle, MySQL, PostgreSQL, Microsoft SQL Server.

d) Java Security

Java Security menyediakan sekumpulan API untuk mengatur security dari aplikasi Java baik secara high level atau low level, seperti public/private key management dan certificates.

e) Java Swing

Java Swing menyediakan sekumpulan API untuk membangun aplikasi – aplikasi GUI (Graphical User Interface) dan model GUI yang diinginkan bisa bermacam – macam, bisa model Java, model Motif/CDE atau model yang dependent terhadap platform yang digunakan.

f) Java RMI

Java RMI menyediakan sekumpulan API untuk membangun aplikasi – aplikasi Java yang mirip dengan model RPC (Remote Procedure Call) jadi object - object Java bisa di call secara remote pada jaringan komputer.

g) Java 2D/3D

Java 2D/3D menyediakan sekumpulan API untuk membangun grafik – grafik 2D/3D yang menarik dan juga akses ke printer.

h) Java Server Pages

Berkembang dari Java Servlet yang digunakan untuk menggantikan aplikasi – aplikasi CGI, JSP (Java Server Pages) yang mirip ASP dan PHP merupakan alternatif terbaik untuk solusi aplikasi Internet.

i) JNI (Java Native Interface)

JNI menyediakan sekumpulan API yang digunakan untuk mengakses fungsi – fungsi pada library (*.dll atau *.so) yang dibuat dengan bahasa pemrograman yang lain seperti C, C++, dan Basic.

j) Java Sound

Java Sound menyediakan sekumpulan API untuk manipulasi sound.

k) Java IDL + CORBA

Java IDL (Interface Definition Language) menyediakan dukungan Java untuk implementasi CORBA (Common Object Request Broker) yang

merupakan model distributed-Object untuk solusi aplikasi besar di dunia networking.

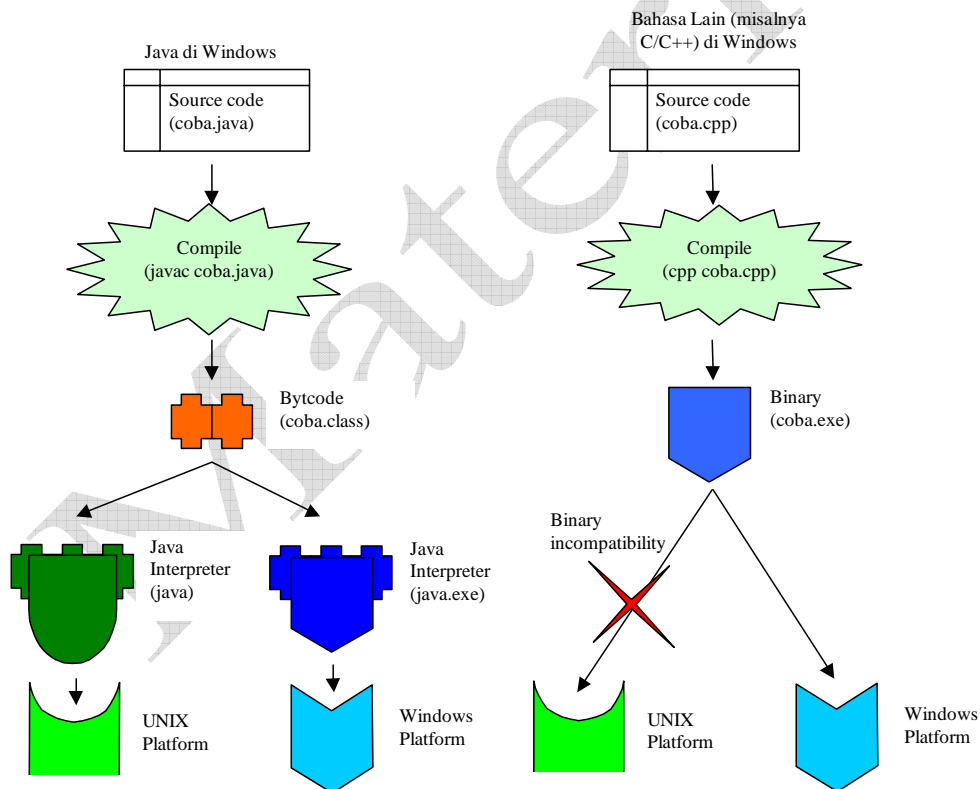
l) Java Card

Java Card utamanya digunakan untuk aplikasi – aplikasi pada smart card, yang sederhana wujudnya seperti SIM Card pada handphone.

m) JTAPI (Java Telephony API)

Java Telephony API menyediakan sekumpulan API untuk memanfaatkan devices – devices telephony, sehingga akan cocok untuk aplikasi – aplikasi CTI (Computer Telephony Integration) yang dibutuhkan seperti ACD (Automatic Call Distribution), PC-PBX dan lainnya.

IV. Metode Compiler Java



V. OOP Java

Pemrograman berorientasi objek diciptakan untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada dalam kehidupan nyata. Dalam paradigma ini, sesuai dengan model kehidupan nyata, segala bagian (entiti) dari suatu permasalahan adalah objek. Objek-objek ini kemudian juga dapat berupa gabungan dari beberapa objek yang lebih kecil. Sebagai contoh, tengoklah sebuah mobil. Mobil adalah sebuah objek dalam kehidupan nyata. Namun mobil sendiri terbentuk dari beberapa objek yang lebih kecil seperti roda ban, mesin, jok, dll. Mobil sebagai objek yang merupakan gabungan dari objek yang lebih kecil dibentuk dengan membentuk hubungan antara objek-objek penyusunnya. Begitu juga dengan sebuah program. Objek besar dapat dibentuk dengan menggabungkan beberapa objek-objek dalam bahasa pemrograman. Objek-objek tersebut berkomunikasi dengan saling mengirim pesan kepada objek lain.

a) Objek

Baik dalam dunia nyata atau dalam sebuah program, sebuah objek memiliki dua karakteristik, yaitu :

1. **State** adalah keadaan dari sebuah objek, seperti mobil memiliki state warna, model, tahun pembuatan, kondisi, dll
2. **Behaviour** adalah kelakuan dari objek tersebut, seperti mobil dapat melaju, membelok, membunyikan klakson, dll.

Objek menyimpan statenya dalam satu atau lebih variabel, dan mengimplementasikan behaviournya dengan metode. Dengan penjelasan di atas, dapat disimpulkan bahwa objek adalah bagian software yang dibentuk dengan variabel-variabel dan metode-metode yang berhubungan dengan variabel tersebut.

Dengan karakteristik tersebut, kita dapat memodelkan berbagai objek dalam kehidupan nyata ke dalam objek-objek dalam sebuah program. Lebih lanjut kita dapat memodelkan objek-objek abstrak ke dalam sebuah program. Contoh umum untuk konsep abstrak seperti ini adalah objek Event, yaitu objek untuk mewakili peristiwa klik atau tombol ditekan.

b) Message (Pesan)

Objek-objek yang bekerja sama membentuk suatu sistem harus saling berkomunikasi untuk menjalankan sistem tersebut. Dalam sebuah program, objek-objek berkomunikasi satu sama lain dengan mengirimkan pesan. Sebagai contoh, jika sebuah objek ingin memanggil metode dari objek lain, maka objek ini akan mengirimkan sebuah pesan yang meminta objek tujuan untuk menjalankan metode yang dikehendaki. Pesan ini akan berisi informasi-informasi yang dibutuhkan objek tujuan untuk dapat menunaikan permintaan tadi. Sebuah pesan dibentuk oleh informasi berikut ini: 1) objek yang dituju; 2) nama metode yang ingin dipanggil; 3) parameter yang dibutuhkan metode tersebut.

Misalnya : `anotherObject.aMethod(parameter1);`

c) Kelas

Kelas adalah semacam cetakan, atau template, untuk membuat objek. Ibaratkan sebuah rancangan rumah yang digunakan untuk membangun ratusan rumah. Rumah yang dibangun tersebut adalah objek dari kelas rancangan rumah. Hal ini dapat dilakukan karena semua objek rumah yang dibangun memiliki karakteristik yang sama, sehingga dapat dibuatkan semacam blueprintnya. Tetapi objek-objek yang dibangun tetap akan memiliki bentuk fisik tertentu sendiri-sendiri, seperti variabel dalam sebuah program, atau pintu sebuah objek rumah. Dengan penjelasan ini, kelas dapat kita definisikan kembali menjadi sebuah blueprint, atau prototipe, yang mendefinisikan variabel dan metode yang sama untuk semua objek sejenis.

d) Pewarisan

Terminologi asing untuk pewarisan adalah inheritance. Mungkin dalam literatur lain Anda akan sering menjumpai istilah ini. Secara gamblang, pewarisan berarti sebuah kelas mewarisi state dan behaviour dari kelas lain. Sebagai contoh, sebuah kelas `RumahMewah` akan mewarisi state dan behaviour dari kelas `Rumah`. Begitu juga dengan kelas `RumahSederhana`. Kelas `RumahMewah` dan `RumahSederhana` disebut subkelas, atau kelas anak, dari kelas `Rumah`, yang disebut superkelas, atau kelas induk.

VI. Pengenalan Aplikasi Java

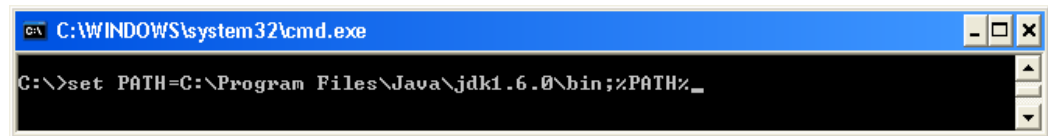
a) Setting Path Java

Pada sistem operasi Windows, perintah untuk setting path yaitu:

```
C:\>set PATH=c:\folderinstalasi\bin;
```

Contoh pada modul ini adalah :

```
C:\> set PATH=C:\Program Files\Java\jdk1.6.0\bin;%PATH%
```



b) Tulis Source Code

Untuk menuliskan code program bisa menggunakan teks editor apa saja, misalnya Notepad. Untuk latihan pertama coba ketikkan syntax dibawah ini dan simpan dengan nama **hello.java** :

```
public class hello
{
    public static void main (String[] args)
    {
        System.out.println("Latihan Java Pertama");
    }
}
```

Keterangan Syntax

- Public, salah satu macam modifier.
- Static, tipe method.
- Void, menunjukkan bahwa method tidak mengembalikan nilai atau objek.
- Main, nama method utama dari program java.
- String, tipe argumen yang diterima untuk parameter dari command java.
- Args, array argumen yang bisa ditambahkan pada saat menggunakan command java untuk menjalankan program java.
- System.out.println("Latihan Java Pertama"), untuk mencetak kata "Latihan Java Pertama" dilayar.

c) Compile Source Code

Untuk mengcompile source code yang telah dibuat menggunakan syntax dibawah ini yang diketikkan pada **CommandPrompt**

javac (nama file yang akan dcompile)

contoh :

c:\>javac hello.java



```
C:\WINDOWS\system32\cmd.exe
C:\>javac hello.java
```

Hasil compile source code ini akan membuat sebuah file dengan nama **hello.class**, file ini selanjutnya yang akan dijalankan.

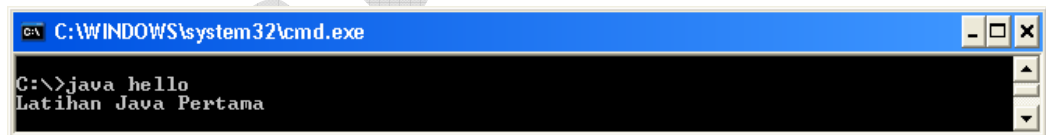
d) Menjalankan Program

Untuk menjalankan file dengan extensi **.class** yang merupakan hasil kompilasi dari file java menggunakan syntax dibawah ini yang diketikkan pada **CommandPrompt**.

java (nama file hasil kompilasi)

contoh :

c:\>java hello



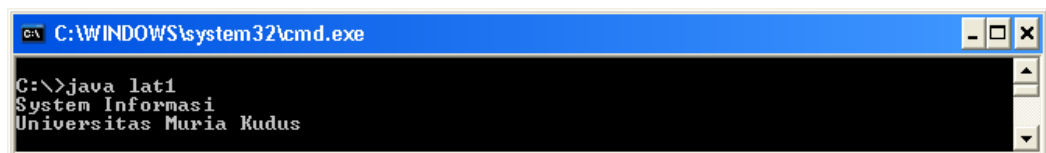
```
C:\WINDOWS\system32\cmd.exe
C:\>java hello
Latihan Java Pertama
```

e) Latihan

Buat sebuah program java untuk menampilkan kata seperti dibawah ini :

System Informasi

Universitas Muria Kudus



```
C:\WINDOWS\system32\cmd.exe
C:\>java lat1
System Informasi
Universitas Muria Kudus
```