

MODUL PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

PENGENALAN KEPADA JAVA

Deskripsi Singkat

Praktikum pemrograman berorientasi objek adalah praktikum yang menggunakan bahasa Java sebagai bantuan dalam memahami konsep pemrograman berorientasi objek. Materi praktikum berisi teori, latihan dan soal pemrograman.

Tujuan

1. Mengetahui cara instalasi Java sebagai bahasa implementasi praktikum pemrograman berorientasi objek.
2. Memahami struktur dasar / sintaks bahasa Java.
3. Membuat program bahasa Java sederhana.
4. Membuat program yang berkaitan dengan tipe data dan variabel.
5. Membuat program untuk memasukkan input dari pengguna.

Prasyarat

Siswa telah mendapat pelajaran mata kuliah Algoritma dan Pemrograman.

Materi 1 : Pengenalan JDK

JDK (Java Development Kit) adalah produk dari Oracle Corporation yang ditujukan untuk pengembang bahasa Java. JDK sebenarnya merupakan Software Development Kit yang khusus dikeluarkan oleh Oracle. JDK menggunakan lisensi *GNU General Public License* (GPL) yang menjadikannya software yang free dan open-source.

JDK terdiri dari banyak komponen tool pemrograman, diantaranya:

- `appletviewer`, yang digunakan untuk menjalankan applet Java tanpa web browser
- `java`, digunakan untuk menginterpretasi file class yang dihasilkan oleh compiler `javac`.
- `javac`, adalah java compiler yang mengubah *source code* menjadi Java bytecode.
- `javadoc`, adalah penghasil dokumentasi otomatis dari komentar yang dibatasi oleh `/** ... */` pada *source code*.
- `jar`, merupakan pengumpul file *source code* dan *library* dalam satu paket `jar`.

Materi 2 : Instalasi JDK

Download file JDK dari website Oracle:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Lakukan instalasi

Materi 3 : Setting Path pada Windows

Setting path dan classpath berguna agar anda dapat menggunakan tool pemrograman JDK di direktori dan folder manapun pada komputer anda.

Buka Control Panel – System

Pilih Advanced.

Pilih button: Environment Variables

Di System variables lakukan setting:

- Variable: Path
- Value: C:\Program Files\Java\jdk1.6.0_18\bin;
Jika telah ada nilai pada value, jangan dihapus, namun tambahkan. Value classpath ini merupakan folder tempat anda menginstal software jdk dan masukkan nilai folder bin, karena disitulah tempat semua tool pemrograman JDK.

Berikutnya anda dapat mencoba hasil setting path, dengan cara:

- Buka windows command prompt (cmd)
- Pada directory manapun anda berada, ketikkan: java -version
- Tekan enter
- Bila keluar versi dari Java, maknanya proses setting path telah berhasil. Ini bermakna anda dapat menyimpan program Java dimana saja, dan dapat mengkompilasi program tersebut di dalam folder tempat anda menyimpan program Java.

Materi 4 : Tips Kompilasi dan Menjalankan Program Java

- Untuk memudahkan organisasi file program Java yang anda buat, buatlah folder dalam direktori apapun yang anda pilih. Hal ini juga akan memudahkan proses kompilasi dan menjalankan program.

- Buka editor (notepad, notepad++, emacs, gedit, jedit, dll).
- Ketik program pada editor.
- Simpan dengan nama file haruslah sama dengan nama class pada program anda. Nama file ini *case sensitive* sehingga haruslah sama huruf kapital dan huruf non-kapital. Contoh jika nama class Hallo, maka simpan dengan nama file: Hallo.java
- Masuk ke direktori dan folder tempat anda menyimpan file program Java dan lakukan kompilasi dengan mengetik:
javac namafile.java
Contoh: javac Hallo.java
Jika muncul error kompilasi, maka perbaiki dulu program anda, baru kemudian compile kembali.
- Setelah proses kompilasi akan terhasil file .class pada folder yang sama. Andaikan nama class anda adalah Hallo, maka akan muncul file Hallo.class dalam folder file program Java anda. Folder .class ini merupakan Java bytecode. Jadi yang anda jalankan (run) adalah file bytecedonya bukan file source code. Cara menjalankan (run) program Java yang anda buat yaitu dengan mengetik:
java namafileclass
Contoh: java Hallo

Materi 5 : Download Java API documentation

Java menyediakan API (Application Programming Interface) untuk semua paket library dan kelas-kelas yang telah siap sedia anda pakai dalam setiap program Java. Oleh karena itu sangatlah **dianjurkan** agar anda mendownload Java API documentation.

Untuk mendownload Java API yang terbaru, silakan buka :

<http://www.oracle.com/technetwork/java/javase/documentation/jdk8-doc-downloads-2133158.html>

Atau anda dapat mengakses API secara online di:

<https://docs.oracle.com/javase/8/docs/api/>

Java API ini adalah kamus rujukan lengkap jika anda ingin membuat program dengan bahasa Java.

Materi 6 : Error

Secara umum, terdapat 2 jenis error dalam pemrograman Java. Pertama ada compile-time error atau yang disebut juga sebagai syntax error dan yang kedua adalah runtime error.

Syntax error biasanya terjadi karena kesalahan penulisan. Mungkin disebabkan kekurangan sebuah perintah di Java atau lupa untuk menulis tanda titik-koma pada akhir pernyataan. Java mencoba untuk mengisolasi error tersebut dengan cara menunjukkan baris dari kode dan terlebih dahulu menunjuk

karakter yang salah dalam baris tersebut. Bagaimanapun juga, error belum tentu berada pada titik yang ditunjuk. Kesalahan umum lainnya adalah dalam kapitalisasi, ejaan, penggunaan dari karakter khusus yang tidak benar, dan penghilangan dari pemberian tanda baca yang sebenarnya.

Runtime error merupakan error yang tidak ditampilkan sampai program dijalankan. Bahkan program yang dikompilasi dengan sukses dapat menampilkan jawaban yang salah. Runtime error ini juga terkait dengan exception, yang akan dipelajari kemudian.

Materi 7 : Tipe Data

Terdapat 2 jenis tipe data di Java yaitu tipe data primitif dan tipe data reference/objek. Java memiliki 8 tipe data primitif. Detil tipe data primitif tersebut dapat dilihat pada gambar di bawah. Masing-masing tipe data umumnya berbeda dari sisi ukuran data yang bisa ditampung.

Tipe data reference merupakan cara untuk mencipta dan mengakses objek dengan memanggil method constructornya. Class yang digunakan bebas tergantung apa yang diperlukan. Objek class dan variabel array termasuk ke dalam tipe data reference. Nilai default dari tipe data reference adalah null.

PRIMITIVE DATA TYPE in JAVA

Type	Contains	Default	Size	Range
byte	Signed integer	0	8 bits	-128 to 127
short	Signed integer	0	16 bits	-32768 to 32767
int	Signed integer	0	32 bits	-2147483648 to 2147483647
float	IEEE 754 floating point	0.0f	32 bits	$\pm 1.4\text{E}-45$ to $\pm 3.4028235\text{E}+38$
long	Signed integer	0L	64 bits	-9223372036854775808 to 9223372036854775807
double	IEEE 754 floating point	0.0d	64 bits	$\pm 4.9\text{E}-324$ to $\pm 1.7976931348623157\text{E}+308$
boolean	true or false	FALSE	1 bit	NA
char	Unicode character	'\u0000'	16 bits	\u0000 to \uFFFF

Materi 8 : Variabel

Semua komponen Java memerlukan nama. Identifier adalah nama yang digunakan untuk nama class, method dan variabel . Ada beberapa syarat identifier pada Java, yaitu:

- Setiap identifier hanya boleh diawali oleh alfabet (A-Z, a-z), \$, _
- Sesudah karakter pertama, karakter berikutnya bebas
- Keyword pada Java tidak bisa digunakan sebagai identifier
- Identifier bersifat case sensitive (huruf kapital dan huruf kecil dianggap berbeda)

Variabel akan memesan tempat di memory untuk menyimpan data. Setiap variabel haruslah memiliki tipe data dan dideklarasikan di dalam program Java. Terdapat 3 jenis variabel dalam Java yaitu:

- Variabel lokal. Variabel lokal dideklarasikan di dalam method, constructor atau blok. Variabel lokal hanya tampak di dalam method, constructor atau blok dan variabel tersebut akan hilang sesudah keluar dari nya. Variabel lokal tidak boleh menggunakan access modifier. Variabel lokal tidak memiliki nilai default sehingga harus dideklarasikan dan diinisialisasi (diberi nilai awal).
- Variabel instance. Variabel instance dideklarasikan di dalam class, namun di luar method, constructor atau blok. Penjelasan detilnya akan dijelaskan pada materi class.
- Variabel static/class. Variabel class/static juga dideklarasikan di dalam class, namun di luar method, constructor atau blok. Penjelasan detilnya akan dijelaskan pada materi yang lain.

Selain variabel, Java juga memiliki modifier. Modifier ada yang berupa access modifier dan ada yang berupa non-access modifier.

Access modifier ada 4 tingkatan (dari yang paling tertutup hingga yang paling terbuka), yaitu:

- private. Hak akses ini membuat variabel hanya tampak di dalam class itu saja.
- protected. Hak akses ini membuat variabel dapat tampak oleh sub-class pada hirarki inheritance.
- default. Default ini tidak ditulis atau bermakna jika variabel instance tidak memiliki access modifier akan dianggap sebagai default. Nilai variabel default akan tampak di dalam satu paket yang sama.
- public. Hak akses ini tidak ada batasan, variabel akan tampak dan bisa diakses tanpa ada batasan.

Sedangkan non-access modifier diantaranya adalah:

- static
- final
- abstract
- synchronized dan volatile

Materi 9 : Input

Terdapat banyak cara untuk melakukan proses Input/Output (I/O) dalam Java. Cara output yang paling sederhana dan telah kita coba pada praktikum 1 adalah `System.out.print` dan `System.out.println`.

`Println` sebenarnya sama seperti `print` hanya, `println` akan menambahkan baris baru (new line) sesudah mencetak pada layar.

Proses input dalam Java juga banyak. Namun yang umumnya akan digunakan dalam praktikum ini adalah dengan memanfaatkan class `Scanner` yang terdapat pada package `java.util` sehingga saat digunakan haruslah diimport terlebih dahulu.

```
import java.util.Scanner;
```

Class `Scanner` memiliki method yang dapat digunakan untuk membaca data untuk semua tipe data primitif dan juga tipe data `String`. Methodnya seperti `next()`, `nextInt()`, `nextBoolean()`, `nextLong()`, `nextDouble()`.

LATIHAN 1

Ketik program berikut:

```
/**
 * The HelloWorldApp class implements an application
 * that displays "Hello World!" to the standard output
 */
public class HelloWorldApp {
    public static void main(String[] args) {
        // Display "Hello World!"
        System.out.println("Hello World!");
    }
}
```

Simpan dengan namafile `HelloWorldApp.java`. Kompilasi dan jalankan program tersebut. Apakah hasilnya?

Nama file haruslah sama dengan nama class. Perhatikan sesudah anda compile, akan muncul `.class` file. File `.class` tersebut lah yang dijalankan.

`public static void main(String[] args)` merupakan method main. Method main adalah method yang akan dipanggil jika program Java dijalankan. Method main memiliki parameter string array yang dapat digunakan untuk menyimpan input dari user saat aplikasi dijalankan dengan perintah Java.

LATIHAN 2

Ubah latihan 1, berikutnya cetak nama anda pada layar output.

LATIHAN 3

Lihat kembali latihan 1, kemudian ubah perintah cetak menjadi:

```
system.out.println("Hello World!");
```

Apakah error yang muncul? Mengapa?

Kemudian ubah lagi menjadi:

```
System.out.println("Hello World!");
```

Apakah error yang muncul? Mengapa?

Kemudian ubah lagi menjadi:

```
System.out.println("Hello World!");
```

Apakah error yang muncul? Mengapa?

LATIHAN 4

Berikutnya kita coba membaca data yang tersimpan pada parameter method main.

```
public class HelloWorldArgs {  
    public static void main(String[] args) {  
        // Tampilkan semua data yang disimpan pada variabel args  
        System.out.println(args[0]);  
        System.out.println(args[1]);  
        System.out.println(args[2]);  
    }  
}
```

```
        System.out.println(args[3]);  
    }  
}
```

Simpan kode di atas dengan nama HelloWorldArgs.java. Berikutnya compile dengan javac.

Lalu jalankan program dengan mengetik:

```
java HelloWorldArgs bulan ini adalah maret
```

Apa yang dapat anda simpulkan?

Semua teks sesudah anda mengetik `java namafile`, akan disimpan pada parameter string array `args` dalam method `main`. Jadi tampaklah disini bahwa method **main** lah yang akan dipanggil saat anda menggunakan perintah **java**.

Namun bukan berarti semua class harus memiliki method `main`. Seperti yang dikatakan sebelumnya, method `main` hanya untuk menjalankan program. Jika class yang anda buat akan digunakan pada class lain, maka class tersebut tidak harus memiliki method `main`. Namun class lain itulah yang akan memiliki method `main`.

LATIHAN 5

Ketik program berikut:

```
/**  
 * Program berikut untuk mengecek tipe data primitif pada Java  
 */  
public class prak1Lat2 {  
    public static void main(String[] args) {  
        // Tipe data primitif  
        long data1 = 767226531;  
        int data2 = 2235641;  
        short data3 = 714;  
        byte data4 = 34;  
        float data6 = (float) 1.733; // tipe data pecahan  
        double data5 = 4.967; // tipe data pecahan  
        char data7 = 'C';  
        boolean data8 = true;  
  
        System.out.println("Nilai Long : "+ data1);  
    }  
}
```


Praktikum 1
Pengenalannya kepada Java

```
        System.out.println("Nilai Int : "+ data2);  
        System.out.println("Nilai Short : "+ data3);  
        System.out.println("Nilai Byte : "+ data4);  
        System.out.println("Nilai Double : "+ data5);  
        System.out.println("Nilai Float : "+ data6);  
        System.out.println("Nilai Char : "+ data7);  
        System.out.println("Nilai Boolean : "+ data8);  
    }  
}
```

Simpan, kompilasi dan jalankan program tersebut dan lihat hasilnya.

Coba anda ubah-ubah isi data yang disimpan. Misal data4 menyimpan 500, apakah yang muncul?
Kemudian data8 menyimpan 1, apa yang terjadi?

LATIHAN 6

Ketik program berikut:

```
/**  
 * Aplikasi untuk melakukan penambahan 2 nilai integer  
 * yang diambil dari parameter method main  
 */  
public class prak1Lat3 {  
    public static void main(String[] args) {  
        int nilai1 = Integer.parseInt(args[0]);  
        int nilai2 = Integer.parseInt(args[1]);  
        int hasil = nilai1 + nilai2;  
  
        System.out.println("Hasil " + nilai1 + " + " + nilai2 + " = "  
+ hasil);  
    }  
}
```

Simpan dan kompilasi program di atas. Jalankan aplikasi dengan menambahkan argumen 2 buah bilangan integer.

Contoh: java prak1Lat3 10 4

Catatan:

Method main pada Java memiliki parameter array bertipe string. Nilai yang disimpan pada parameter dikirimkan sewaktu kita melakukan running program. Contoh: java namafile parameterYangDikirim

Nilai parameter yang dikirimkan tersebut akan diakses dengan menggunakan indeksinya (sama seperti array biasanya). Pada program di atas nilai parameter pertama diakses sebagai args[0], dan nilai parameter kedua diakses sebagai args[1].

Seperti yang dijelaskan sebelumnya, parameter tersebut bertipe String. Karena pada program di atas yang kita perlukan adalah nilai integer, maka nilai String tersebut dapat diubah menjadi integer dengan menggunakan bantuan kelas Integer. Kelas Integer memiliki method parseInt(String nilai) yang memiliki fungsi untuk mengubah nilai String menjadi integer.

LATIHAN 7

Ketik program berikut:

```
/**
 * Aplikasi untuk melakukan penambahan 2 nilai integer
 * yang dimasukkan melalui layar konsol sewaktu run program
 * dengan memanfaatkan method pada class Scanner
 */

import java.util.Scanner;

public class prak1Lat4{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        int nilai1, nilai2, hasil;

        System.out.print("Masukkan integer pertama:");
        nilai1 = input.nextInt();

        System.out.print("Masukkan integer kedua:");
        nilai2 = input.nextInt();

        hasil = nilai1 + nilai2;
        System.out.println("Hasil " + nilai1 + " + " + nilai2 + " = "
+hasil);
    }
}
```

```
}
```

Simpan, kompilasi dan jalankan program tersebut. Bandingkan dengan latihan sebelumnya.

Catatan:

Input dari konsol pengguna dapat menggunakan berbagai cara. Salah satu cara dengan bantuan kelas Scanner. Kelas Scanner memiliki banyak method untuk input tergantung keperluan. Karena pada program ini kita memerlukan nilai integer maka method yang kita gunakan `nextInt()`. Untuk menggunakan kelas Scanner, kita perlu melakukan import paket `java.util` atau langsung import kelas paket Scanner dengan import `java.util.Scanner`.

SOAL-SOAL

1. Buatlah program untuk menghitung harga total suatu barang yang sudah mendapat diskon 15%, dimana jumlah barangnya dan harga perunit akan diinput oleh pengguna. Jumlah harga akan ditampilkan sebagai output.