

Bahasa Pemrograman

Kuliah 13 :: Pemrograman Java

Julio Adisantoso

Mengapa Java?

- **Sederhana (Simple)** - Bahasa pemrograman Java menggunakan sintaks mirip dengan C++, namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit.
- **Terdistribusi (Distributed)** - Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya libraries networking yang terintegrasi pada Java.
- **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.

Slide 1

Java

- Free for download → <http://java.sun.com>
- Unit terkecil program Java adalah Class yang terdiri dari methods (C:procedure) dan instance (C: data)

• Contoh:

```
public class Hello {
    public static void main(String[] args) {
        // menampilkan string ke layar
        System.out.println("Hello world!");
    }
}
```

Slide 2

Program Hello

- Baris pertama:

```
public class Hello
```

menunjukkan nama class yaitu Hello. Pada java, semua kode ditempatkan di dalam deklarasi class, dan kata kunci class. Kata public menunjukkan bahwa class ini mempunyai akses bebas, baik ke class yang lain maupun dari package yang lain pula (package merupakan kumpulan class-class).

- Baris selanjutnya,

```
public static void main(String[] args) {
    menunjukkan nama suatu method dalam class Hello
    yang bertindak sebagai method utama. Sama dengan
    main() dalam program C.
```

Slide 3

Program Hello

- Baris selanjutnya:

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

menampilkan teks "HelloWorld!" ke layar. Perintah **System.out.println()**, menampilkan teks yang diapit oleh tanda kutip double ("...") ke layar.

- Perlu diperhatikan

- Program Java harus disimpan dengan nama *.java
- Nama File seharusnya sama dengan nama class public nya
- Program yang berada pada satu folder dianggap sebagai satu package
- Berisi komentar secukupnya untuk memperjelas kode program

Slide 4

Standard input

- Menggunakan kelas `BufferedReader` yang berada di `java.io`

```
import java.io.*;
```

- Menyimpan input keyboard ke dalam buffer

```
BufferedReader dataIn = new
BufferedReader(new InputStreamReader(System.in));
```

- Menyimpan input ke dalam variabel sementara bertipe String

```
try {
    String temp = dataIn.readLine();
} catch (IOException e) {
    System.out.println("Error input");
}
```

- Contoh (`stdin.java`) → menghitung rata-rata dari n bilangan riil.

Slide 5

Standard output

- Tanpa format

```
System.out.print(rataan);  
System.out.println(rataan);
```

- Dengan format

```
System.out.format("%.2f \n", rataaan);
```

Slide 6

Array dalam Java

- Sama dengan program dalam C
- Mendeklarasikan variabel array

```
int []usia; atau int usia[];
```

- Membuat objek array (dalam Java disebut sebagai instantiation)

```
int usia[];  
usia = new int[100];
```

atau bisa juga ditulis sekaligus menjadi

```
int usia[] = new int[100];
```

- Dapat juga langsung didefinisikan seperti dalam C

```
boolean hasil[]={ true, false, true };
```

```
int[] nilai = {100, 90, 80, 75};
```

```
String hari[] = {"Senin", "Selasa", "Rabu"};
```

Slide 7

Mendefinisikan Class dalam Java

- Definisi class

```
<modifier> class <name> {  
    <attributeDeclaration>*  
    <constructorDeclaration>*  
    <methodDeclaration>*  
}
```

- Contoh:

```
public class Lingkaran {  
    //area penulisan kode selanjutnya  
}
```

Slide 8

Contoh class Lingkaran

```
public class Lingkaran {  
    // Instance variables  
    private double x;  
    private double y;  
    private double r;  
  
    // Instance methods  
    public void set(double x, double y, double r) {  
        this.x=x; this.y=y; this.r=r; }  
    public double luas() {  
        double phi=3.14;  
        return phi*r*r; }  
  
    // main routine  
    public static void main(String[] args) {  
        // penulisan kode program utama  
    }  
}
```

Slide 9

Contoh kasus

- Program membaca n obyek lingkaran menggunakan class Lingkaran.
- Menghitung banyaknya obyek lingkaran yang lebih besar atau sama dengan rata-ratanya.

- Input:

```
3  
1 2 5.6  
1 0 2.1  
9 3 3.6
```

- Output (rataaan luas, n, dan # lingk yg lebih besar)

```
51.00 3 1
```

- Program (luasLingkaran.java)

Slide 10