



IPB University
— Bogor Indonesia —

KOM120C -- PEMROGRAMAN

Object Oriented Programming

OOP using Java

Tim Pengajar Pemrograman IPB University

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.

Tetapi instruksinya panjang untuk melakukan suatu task tertentu.

Java

Free for download

Unit terkecil program Java adalah Class yang terdiri dari methods (C:procedure) dan instance (C: data)

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

Perlu diperhatikan

Program Java harus disimpan ke dalam file dengan nama *.java

Nama File seharusnya sama dengan nama class public nya.

Program yang berada pada satu folder dianggap sebagai satu package (kumpulan dari satu atau lebih class).

Buat komentar secukupnya untuk memperjelas kode program.

Bandingkan

```
class Counter
{
    private:
        int c;
    public:
        Counter() { c=0; }
        void set(int n)
            { c=(n<0)?0:n; }
        void inc() { c++; }
        void dec()
            { c=(c-1<0)?0:c-1; }
        void prin()
            { cout << c << endl; }
};
```

```
public class Counter
{
    private int c;
    public Counter() { c=0; }
    public void set(int n)
        { c=(n<0)?0:n; }
    public void inc() { c++; }
    public void dec()
        { c=(c-1<0)?0:c-1; }
    public void prin()
        { System.out.println(c); }
};
```

Java User Input

Class `Scanner` digunakan untuk membaca data dari standard input, dan terdapat di dalam package `java.util`.

Beberapa method yang dimiliki class `Scanner`:

Method	Description
<code>nextBoolean()</code>	Reads a <code>boolean</code> value from the user
<code>nextByte()</code>	Reads a <code>byte</code> value from the user
<code>nextDouble()</code>	Reads a <code>double</code> value from the user
<code>nextFloat()</code>	Reads a <code>float</code> value from the user
<code>nextInt()</code>	Reads a <code>int</code> value from the user
<code>nextLine()</code>	Reads a <code>String</code> value from the user
<code>nextLong()</code>	Reads a <code>long</code> value from the user
<code>nextShort()</code>	Reads a <code>short</code> value from the user

Class Scanner

Contoh program menjumlahkan 2 bilangan bulat.

```
import java.util.Scanner;
public class Input {

    public static void main (String[] args) {
        int a, b, result;
        Scanner inp = new Scanner(System.in);
        a = inp.nextInt();
        b = inp.nextInt();
        result = a + b;
        System.out.println("Hasil penjumlahan: " + result);
    }
}
```

Standard Output

```
System.out.print(parameter);
```

```
System.out.println(parameter);
```

```
System.out.printf(parameter);
```

```
import java.io.*;

public class Tulis {
    public static void main (String[] args) {
        int a=5;
        System.out.print("Hasil: "+a);
        System.out.println(a+10);
        System.out.printf("%.2f\n", Math.PI);
    }
}
```


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];
```

Membaca Data Bentuk Matrik

Input Data:

```
3 4
1 2 3 4
5 6 7 8
9 0 1 2
```

- Baca satu baris data menggunakan class **Reader** atau **Scanner**.
- Split (pisahkan) setiap nilai yang dipisahkan oleh spasi

Class Objek dan Class Driver

Class Objek dan Class Driver dapat diletakkan dalam satu class, atau terpisah

```
class Object
{
    // elements of class Object
}

public class myClass
{
    public static void main (String[] args)
    {
        //
    }
}
```

TERPISAH

```
public class myClass {
    // elements of class Object

    public static void main (String[] args)
    {
        //
    }
}
```

DALAM SATU CLASS

Latihan

Gunakan Java OOP

Deskripsi

Buat program mengelola bilangan pecahan a b/c.

Format Masukan

Beberapa baris operasi bilangan pecahan:

set a b c	inisialisasi bilangan pecahan a b/c
p	menampilkan bilangan pecahan sesederhana mungkin
add a b c	menambah bilangan pecahan yang ada dengan a b/c
mul a b c	mengalikan bilangan pecahan yang ada dengan a b/c
inc	postfix increment
dec	postfix decrement
end	akhir dari pengolahan

Format Keluaran

Beberapa baris bilangan pecahan sesederhana mungkin.

Contoh Input

```
set 4 2 8
p
add 0 6 8
inc
p
```

Contoh Output

```
4 1/4
5
```