

Bahasa Pemrograman :: Java Class Library

Julio Adisantoso
ILKOM IPB

3 Mei 2010

Objek

- Objek sebagai dasar element dari program
- Setiap objek memiliki karakteristik berupa atribut (*properties*) dan tingkah laku (*behaviour*)
- Contoh

Objek	Properties	Behaviour
Komputer	Merk Ukuran RAM Ukuran Harddisk	Dioperasikan (on) Dimatikan (off) Menjalankan program Mendengarkan lagu
Jam	Merk Nilai jam Nilai menit Nilai detik	Ditentukan angka jam Menambah angka detik

Object and Class

- Objek dalam software mirip dengan objek dalam dunia nyata.
- Setiap objek dibuat dengan memiliki atribut (data) dan tingkah laku (prosedur).
- Pada pemrograman Java, variabel untuk menyimpan data disebut **instance variable** dan prosedur untuk melakukan tingkah laku disebut **instance method**.
- **Class** adalah struktur dasar OOP yang terdiri dari
 - atribut/field/property
 - method

Contoh Object dan Class

Class Komputer		Komputer A	Komputer B
Instance Variable	Merek RAM (MB) HD (GB)	Toshiba 2048 60	Acer 1024 120
Instance Method	On/Off Method Play Music Method Run Program Method		

Instansiasi

- Membuat sebuah objek atau sebuah instance pada sebuah kelas disebut **instansiasi** atau *instantiation*.
- Contoh:

```
String str = new String("Hello");  
String str2 = "OOP Yes";  
Komputer a = new Komputer();  
Komputer b = new Komputer();
```

Mendefinisikan Class

- Definisi class

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

- Contoh:

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

Class Time

```
public class Time {  
    private int hour;        //instance variables  
    private int minute;  
    private int second;  
  
    // default constructor  
    Time() { hour=minute=second=0; }  
  
    public int getHour() {    //accessor method  
        return hour;  
    }  
  
    // mutator methods  
    public void setHour( int hour ) {  
        this.hour=hour;  
    }  
}
```

Instance Variable

- Atribut suatu class harus didefinisikan sebagai **instance variable**.
- Contoh

```
public class Time {  
    private int hour;  
    private int minute;  
    private int second;  
    //penulisan kode selanjutnya  
}
```

- Kata kunci **private** menunjukkan bahwa variabel tersebut hanya dapat diakses oleh class itu sendiri.

Instance Method

- Deklarasi method

```
<modifier><returnType><name> (<parameter>*)  
{  
    <statement>*  
}
```

- Contoh

```
public void setHour(int h) { hour=h; }  
public void setMinute(int minute) {  
    this.minute = minute;  
}
```

- Kata kunci **this.minute** menunjukkan bahwa *minute* adalah atribut dari kelas itu sendiri.

Accessor Method

- **Accessor method**, adalah method yang digunakan untuk mengakses nilai atribut atau ekspresi dari suatu kelas.
- Contoh

```
public int getHour() {
    return hour;
}
```

- Mengapa diperlukan accessor method?

Mutator Method

- **Mutator method**, adalah method yang digunakan untuk memberi atau mengubah nilai atribut dari suatu kelas.
- Contoh

```
public void setHour(int hour) {  
    this.hour=hour;  
}
```

- Mengapa diperlukan mutator method?

Overloading Method

- Overloading method mengizinkan sebuah method dengan nama yang sama namun memiliki parameter yang berbeda sehingga mempunyai implementasi yang berbeda pula
- Contoh

```
public class Time {  
    // ...  
  
    Time() { hour=minute=second=0; }  
    public Time( int h, int m, int s ) {  
        hour=h; minute=m; second=s;  
    }  
  
    // ....  
}
```

Main Routine

- Class dapat digunakan oleh class lain atau class itu sendiri
- Contoh class lain

```
public class appTime {
    public static void main(String[] args)
    {
        Time t1=new Time();
        Time t2=new Time(10,30);
        Time t3=new Time(11,30,20);
        t1.setTime(9,20,15);
        t2.print();
    }
}
```

- Unified Modeling Language
- Cara mendeskripsikan class dalam bentuk diagram
- Contoh

