

## **BAB 3**

### **PENGENALAN PEMROGRAMAN BERORIENTASI OBJEK**

#### **A. POKOK BAHASAN**

- ⊕ Pengenalan Pemrograman Berorientasi Objek
- ⊕ Deklarasi class
- ⊕ Deklarasi atribut
- ⊕ Deklarasi metode
- ⊕ Pengaksesan anggota obyek

#### **B. TUJUAN BELAJAR**

Dengan praktikum ini mahasiswa diharapkan dapat:

- ✓ Menenal pemrograman berorientasi objek
- ✓ Mendeklarasikan suatu class
- ✓ Mendeklarasikan suatu atribut
- ✓ Mendeklarasikan suatu metode
- ✓ Mengakses anggota suatu obyek

#### **C. DASAR TEORI**

##### ➤ Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek (Object Oriented Programming/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object.

PBO memiliki beberapa karakteristik mendasar, antara lain adalah abstraksi, encapulation (pembungkusan), inheritance (pewarisan), dan polymorphism.

##### ➤ Mendeklarasikan suatu class

Class adalah wadah yang berisi abstraksi (pemodelan) dari suatu fungsi objek (benda), yang mendeskripsikan data (sifat karakteristik)

dan fungsi yang dimiliki oleh objek tersebut.

Deklarasi class dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> class <nama_class> {  
    [isi class]  
}  
  
Contoh : public class Mobil{ }
```

#### ➤ Mendeklarasikan suatu atribut

Attributes merupakan nilai (type) data yang terdapat pada suatu object yang berasal dari class. Attributes merepresentasikan karakteristik dari suatu object.

Deklarasi atribut dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <type> <nama_atribut> ;  
  
Contoh : public String warna;
```

#### ➤ Mendeklarasikan suatu metode

Metode/method adalah sesuatu yang dapat dilakukan oleh objek. Method dalam implementasi program ditulis dalam bentuk fungsi. Metode menentukan apa yang terjadi ketika objek itu dibuat serta berbagai operasi yang dapat dilakukan objek.

Deklarasi metode dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <return_type> <nama_metode>  
([daftar_argumen])  
[<statement>]  
}  
  
Contoh : public void info(){  
    System.out.println(warna); }
```

#### ➤ Mengakses anggota suatu obyek

Untuk dapat mengakses anggota-anggota dari suatu obyek, maka harus dibuat instance dari class tersebut terlebih dahulu. Berikut ini adalah contoh pengaksesan anggota-anggota dari class Mobil:

```
public class Mobil {  
    public static void main(String args[])  
    { Mobil m=new Mobil();  
      m.warna="hitam";  
      m.no_Plat="KT 2837 UE";  
      m.info();  
    }  
}
```

## D. PERCOBAAN

### ➤ Percobaan 1

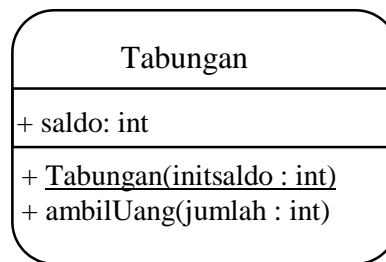
#### Mengakses anggota suatu class

Amati program dibawah ini :

```
public class Rumah {  
    String alamat_rumah;  
    public void setAlamat(String x) {  
        alamat_rumah=x;}  
    }  
    public class Test {  
        public static void main(String args[]) {  
            Rumah r=new Rumah();  
            r.setAlamat("Alamat saya di Jalan Pramuka");  
            System.out.println(r.alamat_rumah);}  
        }  
    }
```

### ○ Percobaan 2

Mengimplementasikan UML class diagram dalam program untuk class Tabungan.



Transformasikan class diagram diatas ke dalam bentuk program?  
Tulislah listing program berikut ini sebagai pengetesan.

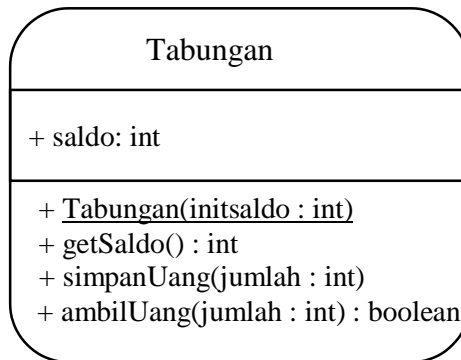
```
public class tesTabungan {  
    public static void main(String args[]) {  
        Tabungan t=new Tabungan(5000);  
        System.out.println("Saldo awal Tabungan Anda :  
        "+t.saldo);  
        t.ambiluang(1500);  
        System.out.println("Jumlah uang yang diambil : 1500");  
        System.out.println("Saldo Tabungan Anda sekarang adalah  
        : " + t.saldo);}  
    }
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar.

```
Saldo awal Tabungan Anda : 5000  
Jumlah uang yang diambil : 1500  
Saldo Tabungan Anda sekarang adalah : 3500
```

## E. LATIHAN

Mengimplementasikan UML class diagram dalam program untuk class **Tabungan**



Transformasikan class diagram diatas ke dalam bentuk program?

Tulislah listing program berikut ini sebagai pengetesan.

```
public class TesTabungan {
    public static void main(String args[]) {
        boolean status;
        Tabungan tabungan=new Tabungan(10000);
        System.out.println("Saldo awal : "+tabungan.getSaldo());
        tabungan.simpanUang(8000);
        System.out.println("Jumlah uang yang disimpan : 8000");
        status=tabungan.ambilUang(7000);
        System.out.print("Jumlah uang yang diambil : 7000");
        {if (status)
            System.out.println(" ok");
        else
            {System.out.println(" gagal");}
        tabungan.simpanUang(1000);
        System.out.println("Jumlah uang yang disimpan : 1000");
        status=tabungan.ambilUang(10000);
        System.out.print("Jumlah uang yang diambil : 10000");
        {if (status)
            System.out.println(" ok");
        else
            System.out.println(" gagal");}
        status=tabungan.ambilUang(2500);
        System.out.print("Jumlah uang yang diambil : 2500");
        {if (status)
            System.out.println(" ok");
        else
            System.out.println(" gagal");}
        tabungan.simpanUang(2000);
        System.out.println("Jumlah uang yang disimpan : 2000");
        System.out.println("Saldo          sekarang          =          "          +
        tabungan.getSaldo());
    }
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

Saldo awal : 10000
Jumlah uang yang disimpan : 8000
Jumlah uang yang diambil : 7000 ok
Jumlah uang yang disimpan : 1000
Jumlah uang yang diambil : 10000 ok
Jumlah uang yang diambil : 2500 gagal
Jumlah uang yang disimpan : 2000
Saldo sekarang = 4000