

Jobsheet 02

Class dan Object

1. Kompetensi

- Mahasiswa dapat memahami deskripsi dari class dan object
- Mahasiswa memahami implementasi dari class
- Mahasiswa dapat memahami implementasi dari attribute
- Mahasiswa dapat memahami implementasi dari method
- Mahasiswa dapat memahami implementasi dari proses instansiasi

2. Pendahuluan

2.1 Class dan Object

Secara singkat class adalah abstraksi dari sebuah object (nyata ataupun tidak nyata) (Roger S Pressman). Apabila kita ingin membuat class **Mahasiswa**, maka kita perlu melakukan identifikasi terhadap objek mahasiswa mengenai ciri/atribut dan perilaku/aksi yang merepresentasikan objek tersebut. Salah satu contoh atribut dari mahasiswa adalah **NIM** (Nomor Induk Mahasiswa) dan perilaku/aksi yang bisa dilakukan oleh mahasiswa adalah **mengikuti UAS**.

Setelah kita memahami makna dari class dan object, langkah selanjutnya adalah melakukan implementasi class melalui pendekatan Object Oriented Programming (dalam mata kuliah ini menggunakan bahasa pemrograman java). Berikut adalah sintaks dari deklarasi class pada pemrograman java:

```
<modifier> class <nama_class>{  
    //deklarasi atribut dan method  
}
```

Aturan penulisan class adalah sebagai berikut:

1. Berupa kata benda,
2. Diawali dengan **huruf kapital**,
3. Jika terdiri dari lebih dari 1 kata, maka setiap kata disambungkan, dan huruf awal dari tiap kata menggunakan **huruf kapital**.

Untuk Access Modifier tidak dibahas pada jobsheet ini, tetapi akan dibahas pada jobsheet berikutnya.

Contoh deklarasi class:

```
public class Mahasiswa {  
  
}
```

2.2 *Attribute*

Untuk melakukan pendeklarasian *attribute* dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <tipe_data> <nama_atribut>;
```

Aturan penulisan atribut adalah sebagai berikut:

1. Berupa kata benda atau kata sifat,
2. Diawali dengan **huruf kecil**,
3. Jika terdiri dari lebih dari 1 kata, maka setiap kata disambungkan, dan huruf awal dari tiap kata menggunakan **huruf kapital**

Contoh deklarasi *attribute*:

```
public String nim;  
public String nama;  
public String alamat;
```

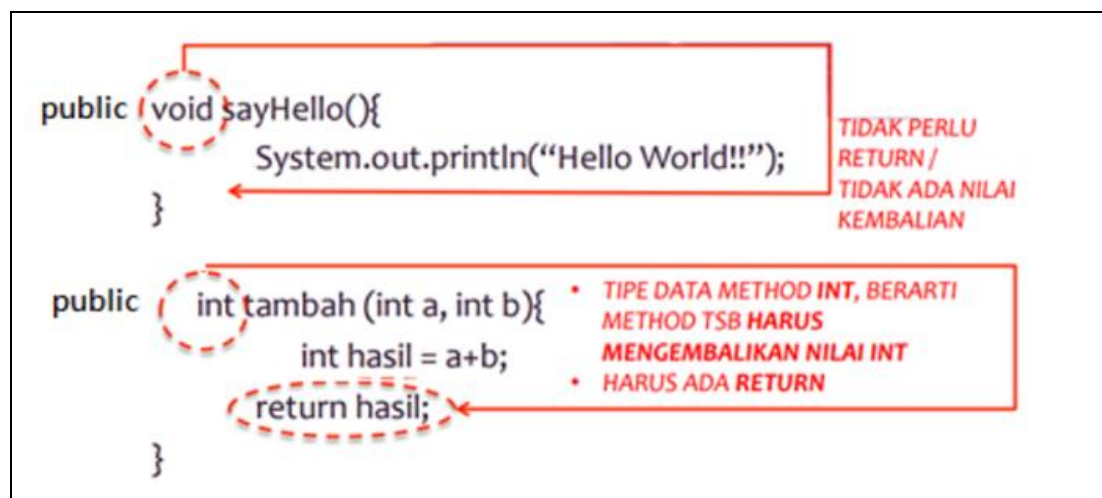
2.3 *Method*

Method pada suatu objek merepresentasikan perilaku dari objek tersebut atau aksi/fungsi/prosedur/proses yang dapat dilakukan. Method diimplemetasikan sebagai suatu blok yang berisi statement atau baris kode program.

Method dideklarasikan dengan syntax sebagai berikut:

```
<modifier> <return_type> <nama method>(param1, param2, ...)  
{  
    //statements  
}
```

Method dengan *return type* **void**, berarti tidak memiliki *return value* (nilai kembalian), sehingga tidak memerlukan kata kunci *return* di dalamnya. Sedangkan method dengan *return type* **selain void**, berarti memerlukan suatu *return value*, sehingga harus ada nilai yang dikembalikan dengan kata kunci *return* di dalamnya.



Aturan penulisan method adalah sebagai berikut:

1. Berupa kata kerja,
2. Diawali dengan **huruf kecil**,
3. Jika terdiri dari lebih dari 1 kata, maka setiap kata **disambungkan**, dan huruf awal dari tiap kata menggunakan **huruf kapital**.

2.4 Object

Setelah class dibuat, langkah selanjutnya adalah membuat object. Proses pembuatan Object dari suatu Class disebut **instansiasi** menggunakan kata kunci *new*. Syntax dasar instansiasi adalah sebagai berikut:

```
NamaClass namaObject = new NamaClass();
```

Contoh:

```
Mahasiswa mhs = new Mahasiswa();
Mahasiswa ani = new Mahasiswa();
Mahasiswa mahasiswa = new Mahasiswa();
Random r = new Random();
Pegawai pegawai1 = new Pegawai();
```

Pada baris pertama contoh di atas, dibuat objek baru dengan nama *mhs* yang bertipe *Mahasiswa*.

3. Percobaan

3.1 Percobaan 1: Instansiasi objek, mengakses atribut, memanggil method

Langkah kerja:

1. Bukalah development tools, misalkan Netbeans, Visual Code, dsb
2. Ketikkan kode program berikut ini:

```
public class Mahasiswa {
    public String nim;
    public String nama;
    public String alamat;
    public String kelas;

    public void displayBiodata(){
        System.out.println("NIM      : " + nim);
        System.out.println("Nama      : " + nama);
        System.out.println("Alamat   : " + alamat);
        System.out.println("Kelas   : " + kelas);
    }
}
```

3. Simpan dengan nama file Mahasiswa.java.
4. Untuk membuat objek baru dengan tipe mahasiswa, dilakukan instansiasi class Mahasiswa seperti contoh berikut:

```
public class MahasiswaDemo {
    public static void main(String[] args) {
        Mahasiswa m1 = new Mahasiswa();
        m1.nim = "023432";
        m1.nama = "Yansy Ayuningtyas";
        m1.alamat = "Nias, Sumatera Utara";
        m1.kelas = "2A";

        m1.displayBiodata();
    }
}
```

5. Simpan file dengan MahasiswaDemo.java
6. Jalankan class MahasiswaDemo.java
7. Pada bagian mana proses pendeklarasian atribut pada program di atas?
8. Pada bagian mana proses pendeklarasian method pada program di atas?
9. Berapa banyak objek yang di instansiasi pada program di atas?
10. Apakah yang sebenarnya dilakukan pada sintaks program “m1.nim=101” ?
11. Apakah yang sebenarnya dilakukan pada sintaks program “m1.displayBiodata()” ?
12. Instansiasi 2 objek mahasiswa baru pada class MahasiswaDemo

4.2 Percobaan 3: Method dengan return value

Langkah kerja:

1. Bukalah text editor atau IDE, misalnya Notepad ++ / netbeans.
2. Ketikkan kode program berikut ini:

```
public class Barang {
    public String kode;
    public String nama;
    public double hargaKotor;
    public double diskon;
}
```

3. Simpan dengan nama file Barang.java
4. Buat method yang menghitung dan mengembalikan nilai harga bersih berdasarkan atribut diskon dan hargaKotor

```
public double getHargaBersih() {
    return hargaKotor - diskon * hargaKotor;
}
```

5. Buat method untuk mencetak info dari suatu barang. Nilai harga bersih diperoleh dengan memanggil method getHargaBersih()

```
public void displayInfo() {
    System.out.println("Kode      : " + kode);
    System.out.println("Nama      : " + nama);
    System.out.println("Harga Kotor : " + hargaKotor);
    System.out.println("Diskon      : " + diskon);
    System.out.println("Harga Bersih: " + getHargaBersih());
}
```

6. Buat file baru BarangDemo.java kemudian instansiasi objek barang baru

```
public class BarangDemo {  
    public static void main(String[] args) {  
        Barang barang1 = new Barang();  
        barang1.kode = "ATK01";  
        barang1.nama = "Bolpoin Pilot Hitam";  
        barang1.hargaKotor = 3500;  
        barang1.diskon = 0.1;  
  
        barang1.displayInfo();  
    }  
}
```

7. Jalankan program tersebut!
8. Ambil kesimpulan tentang kegunaan dari kata kunci *return*, kapan suatu method harus memiliki kata kunci *return*?

4.3 Tugas

1. Implementasikan studi kasus berikut dengan paradigman PBO.

Class **PersegiPanjang** memiliki atribut **panjang dan lebar** dengan tipe data integer
Class tersebut juga memiliki tiga method:

- Method displayInfo() untuk menampilkan data panjang dan lebar
- Method getLuas() untuk menghitung luas dari
- Method getKeliling() untuk menghitung keliling

Tampilkan data persegi, nilai luas persegi dan keliling persegi pada class **DemoPersegi**.

2. Implementasikan **salah satu** class yang telah dibuat pada tugas Teori PBO 02 ke dalam java dengan paradigma PBO. Instansiasikan 2 objek dari class tersebut pada class lain. Lakukan update nilai atribut dari setiap objek dan eksekusi method yang dimiliki.