

# 2

## Modul 2 - Atribut, Behaviour, dan Constructor

**Durasi Modul = 2-3 menit**

### Capaian pembelajaran:

Setelah mempelajari modul ini, mahasiswa diharapkan mampu:

- Memahami dan menjelaskan konsep atribut dalam pemrograman berorientasi objek
- Mengidentifikasi dan menerapkan behavior melalui penggunaan method dalam sebuah class
- Menjelaskan fungsi dan jenis-jenis konstruktor dalam pemrograman berorientasi objek
- Membuat dan mengimplementasikan atribut, behavior, dan konstruktor dalam program Java
- Menganalisis perbedaan antara berbagai jenis konstruktor dan penggunaannya
- Menerapkan konsep atribut, behavior, dan konstruktor untuk merancang class yang efektif dan efisien

### Deskripsi Pembelajaran :

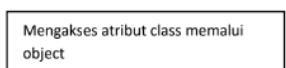
Pada modul ini, mahasiswa akan mempelajari tiga konsep dasar dalam pemrograman berorientasi objek: atribut, behavior, dan konstruktor. Materi ini akan membantu mahasiswa memahami:

- Definisi dan penggunaan atribut dalam sebuah class
- Konsep behavior dan bagaimana mendefinisikannya melalui method
- Fungsi dan jenis-jenis konstruktor dalam pemrograman berorientasi objek

Setelah mempelajari modul ini, mahasiswa diharapkan dapat menerapkan konsep-konsep tersebut dalam pembuatan program yang lebih kompleks dan efisien.

## 2.1. Atribut

```
public class MyClass {  
    int x = 5;  
  
    public static void main(String[] args) {  
        MyClass myObj = new MyClass();  
        System.out.println(myObj.x);  
    }  
}
```



Mengakses atribut class memalui object

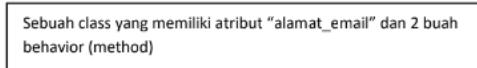
**Atribut** adalah data yang ada di dalam sebuah class dan disebut juga dengan variabel atau property

. **Atribut** dapat berada dalam **sebuah class**, atau **dalam sebuah method** yang ada dalam sebuah class.

**Atribut** juga dapat **diakses langsung** dengan menggunakan **object**.

## 2.2. Behavior

```
public class Manusia {  
    String alamat_email;  
  
    void sendEmail(String pesan){  
        System.out.println("Mengirim pesan: "+pesan+" ke alamat email "+alamat_email);  
    }  
    void verifyEmail(){  
        System.out.println("Alamat email "+alamat_email+" sudah diverifikasi");  
    }  
}
```



Sebuah class yang memiliki atribut "alamat\_email" dan 2 buah behavior (method)

**Behaviour** menggambarkan bahwa :



1. Apa saja operasi-operasi yang dapat dilakukan oleh sebuah class
2. Mengakses atribut class melalui object
3. Bagaimana sebuah object dari sebuah class "behave" atau beroperasi
4. Bagaimana sebuah object bereaksi jika diperintahkan melakukan sesuatu atau jika terjadi perubahan state Untuk mendefinisikan behavior dari sebuah class atau object, kita dapat mendefinisikannya dengan membuat method.

## 2.3. Constructor

```
class Box
{
    double width, height, depth;

    Box(double w, double h, double d)
    {
        width = w;
        height = h;
        depth = d;
    }

    Box()
    {
        width = height = depth = 0;
    }

    Box(double len)
    {
        width = height = depth = len;
    }

    double volume()
    {
        return width * height * depth;
    }
}
```

- 1. Constructor dengan 3 parameter
- 2. Constructor tanpa parameter
- 3. Constructor dengan 1 parameter

Class yang memiliki lebih dari 1 constructor seperti ini, dinamakan constructor overloading

```
// Driver code
class Test
{
    public static void main(String args[])
    {
        // create boxes using the various
        // constructors
        Box mybox1 = new Box(1, 2, 1.5);
        Box mybox2 = new Box();
        Box mycube = new Box(7);

        double vol;

        // get volume of first box
        vol = mybox1.volume();
        System.out.println(" Volume of mybox1 is " + vol);

        // get volume of second box
        vol = mybox2.volume();
        System.out.println(" Volume of mybox2 is " + vol);

        // get volume of cube
        vol = mycube.volume();
        System.out.println(" Volume of mycube is " + vol);
    }
}
```

Constructor akan dijalankan ketika sebuah object dari sebuah class bersangkutan dibuat. Karena class Box memiliki constructor dengan parameter, maka ketika object-nya dibuat, argument-nya juga harus diselipkan

**Constructor** adalah method khusus yang digunakan salah satunya untuk inisialisasi atribut sebuah class. Pada saat pembuatan suatu objek, terdapat method yang pertama kali dijalankan. Method ini digunakan untuk memberikan

nilai awal pada saat objek diciptakan. Method ini disebut dengan constructor. Constructor dipanggil secara otomatis ketika perintah new digunakan untuk membuat instan class.

Perbedaan method constructor dengan method yang biasa kita kenal adalah:



1. Nama constructor harus persis sama dengan nama class-nya.
2. Tidak boleh ada kata kunci void.
3. Tidak dapat mengembalikan nilai sehingga tidak boleh ada statement return.
4. Seperti method, constructor juga dapat memiliki parameter.
5. Suatu class dapat memiliki lebih dari satu constructor dengan syarat parameternya tidak boleh ada yang sama.
6. Suatu constructor dapat memanggil constructor lain dalam satu class.

▼ Constructor ada 3, yaitu :

- a. Constructor tanpa parameter

```
PersegiPanjang() {
```

```
}
```

- b. Constructor berparameter

```
PersegiPanjang(double panjang, double lebar) {  
    this.panjang = panjang;  
    this.lebar = lebar;  
}
```

- c. Constructor berparameter object

```
PersegiPanjang(PersegiPanjang pg) {  
    this.panjang = pg.panjang;  
    this.lebar = pg.lebar;  
}
```

## REFERENSI

Aziefah, & Nurmasari, J. (2013). Pemrograman Berorientasi Objek. Pekanbaru: Politeknik Caltex Riau.

Barclay, K., & Savage, J. (2004). Object Oriented Design with UML and Java. Elsevier.

Deitel, P., & Deitel, H. (2012). Java: How to Program (9 ed.). US: Prentice Hall.