

# Pemrograman Berorientasi Objek

MODUL 1 OBJEK DAN KELAS

ASISTEN LABORATORIUM

Tujuan Pembelajaran :

1. Mengetahui apa itu kelas dan objek pada Pemrograman Berorientasi Objek
2. Mengetahui apa saja komponen yang dimiliki oleh kelas
3. Mengetahui bagaimana membuat sebuah objek
4. Mengetahui bagaimana memanggil kelas

---

**Objek** – Objek adalah Entitas yang memiliki atribut atau elemen yang menyatakan bentuk objek tersebut serta memiliki kebiasaan yang dilakukan oleh entitas tersebut

**Kelas** – Kelas adalah sebuah cetak biru yang menjelaskan kebiasaan/elemen yang dimiliki oleh sebuah objek yang dibuatnya

### Objek pada Java

Objek adalah entitas yang menampilkan benda/mahluk/konsep yang berada di dunia asli ini yang dapat dijelaskan serta memiliki elemen/atribut dan kebiasaan/kegiatan yang dilakukan objek tersebut.

Objek pada software dibagi menjadi 2 juga miliki atribut dan kebiasaan/kegiatan yang terdefinisi sebagai berikut :

1. Atribut = Variabel pada field
2. Kebiasaan/Kegiatan = Fungsi/Metode

Fungsi/Metode ini digunakan untuk melakukan operasi internal atribut dan melakukan komunikasi antar objek

### Kelas pada Java

```
public class Cat{  
    String name;  
    int age;  
    String type;  
  
    void walking(){
```

```
    }  
  
    void eating(){  
  
    }  
  
    void sleeping(){  
  
    }  
  
}
```

Sebuah kelas berisikan variable seperti berikut :

- Local Variables – Variable yang dibuat didalam sebuah fungsi/method, konstruktor
- Instances Variables – Variable yang didalam kelas tapi diluar dari method
- Class Variables – Variable yang diluar dari kelas tapi di deklarasi dengan static keyword

Di sebuah kelas juga dapat dieklarasikan sejumlah fungsi yang mengidentifikasi / mengoperasikan nilai yang berada di dalam kelas seperti walking(), eating(), sleeping().

## Konstruktor

Setiap kelas memiliki konstruktor bahkan jika kita tidak mendefinisikannya akan terbuat secara default konstruktor tersebut. Konstruktor adalah sebuah fungsi yang menjalankan sebuah kelas, jika kita membuat objek maka konstruktor tersebutlah yang akan terpanggil/terbentuk. Sebuah kelas dapat memiliki lebih dari satu konstruktor. Contoh :

```
public class Cat{  
  
    String name;  
    int age;  
    String type;  
  
    public Cat(){  
        //arguments  
    }  
  
    public Cat(String name){
```

```
        //arguments
    }
    ...
}
```

### Membuat Objek

Kelas adalah blueprint untuk sebuah objek, Maka, objek terbuat dari sebuah kelas. Di Java keyword untuk membuat objek adalah **new**.

Ada 3 tahap untuk membuat sebuah objek:

- Deklarasi (Declaration) – Pembuatan variable menggunakan tipe objek.
- Instansiasi (Instatiation) – kata kunci new saat membuat objek.
- Inisialisasi (Initialization) – pemanggilan konstruktor untuk menjadi objeknya.

Contoh :

```
public class Cat {
    public Cat(String name) {
        // This constructor has one parameter, name.
        System.out.println("Passed Name is :" + name );
    }

    public static void main(String []args) {
        // Following statement would create an object myPuppy
        Cat mycat = new Cat( "tommy" );
    }
}
```

### Pengaksesan variable objek

```
/* First create an object */
ObjectReference = new Constructor();

/* Now call a variable as follows */
ObjectReference.variableName;

/* Now you can call a class method as follows */
ObjectReference.MethodName();
```

Contoh :

```
public class Puppy {
    int puppyAge;

    public void setAge( int age ) {
        this.puppyAge = age;
    }

    public static void main(String []args) {
        /* Object creation */
        Puppy myPuppy = new Puppy();

        /* Call class method to set puppy's age */
        myPuppy.setAge( 2 );

        /* Call another class method to get puppy's age */
        System.out.println("Variable Value :" + myPuppy.puppyAge );
    }
}
```

### Import Kelas

Di Java untuk membuat kelas biasanya disertakan dengan nama packagenya sehingga saat kelas akan diimport akan mudah di baca oleh compiler untuk dicari filenya. Untuk import menggunakan katakunci **import**.

Contoh :

Buat Class bernama Puppy

```
public class Puppy {
    public int puppyAge;

    public void setAge( int age ) {
        this.puppyAge = age;
    }
}
```

Kemudian buat kelas Result

```
import <packagename>.Puppy;

public class Result {
    public static void main(String []args) {
        /* Object creation */
        Puppy myPuppy = new Puppy();

        /* Call class method to set puppy's age */
        myPuppy.setAge( 2 );

        /* Call another class method to get puppy's age */
        System.out.println("Variable Value :" + myPuppy.puppyAge );
    }
}
```