

Praktikum 3: Constructor

Default Constructor

- Class Mobil di bawah ini menggunakan default constructor, pada class ini default constructor tidak dituliskan.

```
class Mobil {  
    String warna = "merah";  
  
    // penamaan method di bawah ini menggunakan lowerCamelCase  
    void lihatWarna() {  
        System.out.println(warna);  
    }  
}
```

- Pada class Mobil di bawah ini menggunakan default constructor dengan menuliskan default constructor tersebut.

```
class Mobil {  
    String warna = "merah";  
  
    // default constructor  
    public Mobil() {}  
  
    void lihatWarna() {  
        System.out.println(warna);  
    }  
}
```

- Pada class Mobil di bawah ini, value warna diberikan pada default constructor sehingga saat constructor dipanggil pada waktu instansiasi, maka state warna pada instance (object) akan secara otomatis berisi String "red".

```
class Mobil {  
    String warna;  
  
    public Mobil() {  
        warna = "red";  
    }  
  
    void lihatWarna() {  
        System.out.println(warna);  
    }  
}
```

- .Pada driver class di bawah ini, yaitu class TestMobil, sebuah instance (object) dibuat dengan cara memanggil constructor Mobil().

```
// penamaan class di bawah ini menggunakan UpperCamelCase
class TestMobil {
    public static void main(String[] args) {
        Mobil mbl = new Mobil();
        mbl.lihatWarna();
    }
}
```

Constructor with Parameter

- Pada class Mobil di bawah ini, constructor memiliki parameter String wrn, parameter tersebut nantinya akan men-set field String warna dengan nilai yang dimasukkan pada saat constructor dipanggil.

```
class Mobil {
    String warna;

    public Mobil(String wrn) {
        warna = wrn;
    }

    void lihatWarna() {
        System.out.println(warna);
    }
}
```

- Karena constructor pada class Mobil di atas memiliki parameter, maka pada saat constructor dipanggil harus dengan mengisi parameter sesuai dengan tipe data dari parameter tersebut, yaitu String.

```
class TestMobil {
    public static void main(String[] args) {
        Mobil mbl = new Mobil("merah");
        mbl.lihatWarna();
    }
}
```

Constructor with more Parameters

```
class Mobil {
    String merek;
    String warna;
    String tahun;

    public Mobil(String mrk, String wrn, int thn) {
        merek = mrk;
        warna = wrn;
        tahun = thn;
    }

    void lihatMerek() {
        System.out.println(merek);
    }

    void lihatWarna() {
        System.out.println(warna);
    }

    void lihatTahun() {
        System.out.println(tahun);
    }
}

class TestMobil {
    public static void main(String[] args) {
        Mobil mbl = new Mobil("Honda", "merah", 2020);
        mbl.lihatMerek();
        mbl.lihatWarna();
        mbl.lihatTahun();
    }
}
```

Catatan:

- Penamaan class (begitu juga dengan constructor) menggunakan UpperCamelCase jika lebih dari 1 kata
- Penamaan object, field, method menggunakan lowerCamelCase jika lebih dari 1 kata