

## MODUL 4

### Konstruktor



#### CAPAIAN PEMBELAJARAN

---

Dapat membuat dan menggunakan konstruktor



#### KEBUTUHAN ALAT/BAHAN/SOFTWARE

---

1. Textpad
2. JDK



#### DASAR TEORI

---

Konstruktor adalah sebuah tipe khusus dari method yang digunakan untuk membuat dan menginisialisasi sebuah object baru.

Berikut ini adalah *property* dari konstruktor :

1. konstruktor memiliki nama yang sama dengan class
2. konstruktor tidak memiliki return value
3. konstruktor tidak dapat dipanggil secara langsung, namun harus dipanggil dengan menggunakan operator **new** pada pembentukan sebuah *class*.

Untuk mendeklarasikan konstruktor, kita tulis

```
<modifier> <className> (<parameter>) {
```

**<statement>**

**}**

Perbedaan method biasa dengan konstruktor adalah bahwa konstruktor harus memiliki nama yang sama dengan nama classnya dan tidak memiliki nilai kembalian (tipe-data).

Konstruktor dijalankan pada saat sebuah object diinisialisasi (menggunakan kata new).

### ***Konstruktor default***

Apabila anda tidak mendeklarasikan satu pun konstruktor, maka Java secara otomatis menambahkan konstruktor default ke dalam class yang kita buat walaupun tidak kelihatan pada kode program. Apabila kita mendeklarasikan satu atau lebih konstruktor maka java tidak akan menambahkan konstruktor default.

### ***Konstruktor overloading***

Pada konstruktor juga berlaku overloading, artinya boleh mendeklarasikan lebih dari satu konstruktor, asalkan memiliki parameter yang berbeda – beda.

Pada konstruktor :

1. Memiliki konsep yang sama dengan overloading method.
2. Dibedakan berdasarkan paramater (jumlah atau tipe data).

Pemanggilan konstruktor tergantung pada instansiasi objek.



## **PRAKTIK**

---

### **1. Praktik 1 Membuat konstruktor**

```
public class Titik{
    int x;
    int y;
    public Titik(){
        System.out.println("Konstruktor titik dijalankan!");
    }
}
```

```
}  
}
```

## 2. Praktik 2 Membuat obyek dan pemanggilan konstruktor

```
public class TesTitik {  
    public static void main(String[] args) {  
        Titik a=new Titik();  
    }  
}
```

## 3. Praktik 3 Membuat konstruktor overloading dan pemanggilan konstruktor

Tambahkan konstruktor baru pada class Titik

```
public Titik(int x, int y) {  
    this.x = x;  
    this.y = y;  
    System.out.println("Konstruktor titik 2  
dijalankan!");  
}
```

Tambahkan skrip berikut pada class TesTitik

```
Titik b=new Titik(10,10);
```

## 4. Praktik 4 Overloading konstruktor

```
public class Pegawai  
{  
    String NamaPegawai;  
    int IdPegawai;  
    String PosisiPegawai;  
  
    Pegawai (String nama,int Id, String posisi) // konstruktor  
    {  
        NamaPegawai = nama;  
        IdPegawai = Id;  
        PosisiPegawai = posisi;  
    }  
    Pegawai () //konstruktor  
    {  
        NamaPegawai = "Azkiya";  
        IdPegawai = 2514;  
        PosisiPegawai = "Staf pengajar";  
    }  
    void Show()  
    {  
        System.out.println("Informasi Pegawai");  
        System.out.println("Nama      : "+NamaPegawai);  
        System.out.println("Id      : "+IdPegawai);  
    }  
}
```

```

        System.out.println("Posisi    : "+PosisiPegawai);
    }
    public static void main(String args[])
    {

        Pegawai pegawai1 = new Pegawai();
        Pegawai pegawai2 = new Pegawai("Zahwa",3313,"Staf
Akademik");
        pegawai1.Show();
        pegawai2.Show();

    }
}

```



## LATIHAN

---

1. Modifikasi class buku pada modul 1 dengan menggunakan konstruktor !



## TUGAS

---

1. Buatlah class Komputer lengkap dengan atribut, method dan konstruktor !



## REFERENSI

---

Wagito, Sumiyatun, Fx. Henry Nugroho. 2015. Modul Praktikum Pemrograman Berorientasi Objek. STMIK AKAKOM. Yogyakarta

