Pertemuan 9

OVERLOADING

OVERLOADING PADA METHOD

- Dalam satu class dapat mendefinisikan lebih dari satu method dengan nama yang sama tetapi parameter yang dideklarasikan harus berbeda baik itu jumlahnya ataupun tipe parameternya.
- Jenis-jenis method overloading:
  1. Jumlah parameter beda, tipe beda
  2. Jumlah parameter sama, tipe beda
  3. Jumlah parameter beda, tipe sama

Syarat-syarat method overloading pada method

- Dalam sebuah kelas diperbolehkan lebih dari satu method dengan nama yang sama dengn catatan method dapat dibedakan berdasarkan banyaknya parameter atau tipe data parameter.
- Overloading tidak bisa dilakukan dengan membedakan nilai kembaliannya.
- Overloading juga bisa dilakukan pada sub kelasnya.

Contoh method yang mengalami overloading

Method yang di Overload :	
myMethod(String s, int i, float f)	
Method yang mengOverload (LEGAL)	
myMethod()	Jumlah arugument berbeda
myMethod(int i)	Jumlah arugument berbeda
myMethod(String s1, String s2, String s3)	Jumlah arugument berbeda, tapi urut-urutan tipe data berbeda
myMethod(String s, float f, int i)	Jumlah arugument berbeda, tapi urut-urutan tipe data berbeda
Method yang mengoverload (ILEGAL)	
myMethod(String s, int i, float f)	Jumlah argument sama, urut-urutan tipe data sama.
myMethod(String **x**, int **y**, float **z**)	Jumlah argument sama, urut-urutan tipe data sama. Yang haru diperhatikan dari contoh ini adalah, nama variabel yang berbada tidak lantas menyebabkan argument list menjadi berbeda.

CONTOH OVERLOADING PADA METHOD

```
class mtk{
    static double kuadrat(double nilai){
        return nilai*nilai;
    }
    static int kuadrat(int nilai){
        return nilai*nilai;
    }
    static double kuadrat(String nilai){
        double bil;
        bil=Double.parseDouble(nilai);
        return bil*bil;
    }
}

public class mat{
    public static void main(String []args){
        System.out.println(mtk.kuadrat(25.0));
        System.out.println(mtk.kuadrat(25));
        System.out.println(mtk.kuadrat("25"));
    }
}
```

OVERLOADING PADA KONSTRUKTOR

- Constructor dapat melakukan overloading.
- Overloading constructor terjadi pada objek yang berbeda-beda, nilai argument berbeda, sehingga method bisa lebih dari satu.

CONTOH OVERLOADING PADA KONSTRUKTOR

```
class bentuk{
    int[]koordinat;
    bentuk(int x1, int x2){
        this.koordinat=new int[2];
        this.koordinat[0]=x1;
        this.koordinat[1]=x2;
    }
    bentuk(int x1, int y1, int x2, int y2){
        this.koordinat=new int[4];
        this.koordinat[0]=x1;
        this.koordinat[1]=y1;
        this.koordinat[2]=x2;
        this.koordinat[3]=y2;
    }
    bentuk(int x1, int y1, int x2, int y2, int x3, int y3){
        this.koordinat=new int[6];
        this.koordinat[0]=x1;
        this.koordinat[1]=y1;
        this.koordinat[2]=x2;
        this.koordinat[3]=y2;
        this.koordinat[4]=x3;
        this.koordinat[5]=y3;
    }
```

(...LANJUTAN) CONTOH OVERLOADING PADA KONSTRUKTOR

```
    public void info(){
        switch(koordinat.length){
            case 2:
            System.out.println("Bentuk TITIK");
            System.out.println("Koordinat : ");
            System.out.println(koordinat[0]+" - "+koordinat[1]);
            break;

            case 4:
            System.out.println("Bentuk GARIS");
            System.out.println("Koordinat : ");
            System.out.println(koordinat[0]+" - "+koordinat[1]);
            System.out.println(koordinat[2]+" - "+koordinat[3]);
            break;

            case 6:
            System.out.println("Bentuk SEGITIGA");
            System.out.println("Koordinat : ");
            System.out.println(koordinat[0]+" - "+koordinat[1]);
            System.out.println(koordinat[2]+" - "+koordinat[3]);
            System.out.println(koordinat[4]+" - "+koordinat[5]);
            break;
        }
        System.out.println(" ");
    }
}
```

(...LANJUTAN) CONTOH OVERLOADING PADA KONSTRUKTOR

```
public class konstruktor{
    public static void main(String[]args){
        bentuk bentuk1=new bentuk(20,30);
        bentuk1.info();
        bentuk bentuk2=new bentuk(20,30,40,60);
        bentuk2.info();
        bentuk bentuk3=new bentuk(20,30,40,60,50,30);
        bentuk3.info();
    }
}
```