

# BAB 3

## OVERLOAD DAN OVERLOADING METHOD

### Tujuan

1. Praktikan mampu memahami konsep static method yang ada di java
2. Mampu membedakan perbedaan method yang menggunakan kata kunci static atau tidak
3. Mampu memahami dan mengimplementasikan method overloading

### Ringkasan Materi

#### A. Overloading Method

Penamaan method pada OOP (Object Oriented Programming) menjadi sangat penting terutama pada pemrograman menggunakan bahasa java. Dalam penamaan method, terkadang tanpa sadar kita memberi nama yang sama pada method yang berbeda sehingga dapat mengakibatkan kesalahan pada saat program dijalankan. Untuk mengatasi hal ini, Java memperkenalkan istilah *overloading*, Overloading adalah teknik penamaan method dengan nama yang sama namun memiliki tipe dan jumlah argumen atau parameter yang berbeda. Sebagai contoh adalah method Hitung pada class Lingkaran, dimana pada class ini terdapat method bernama Hitung dengan parameter a dengan tipe integer.

```
public class Lingkaran{
    public static void Hitung(int a){
        //kode program
    }
}
```

Kemudian pada class tersebut dibuat method baru bernama Hitung namun parameternya bertipe double dengan nama value

```
public static void Hitung(double value){
    //kode program
}
```

Kedua method ini disebut overloading method karena memiliki nama yang sama tetapi tipe dari argumennya berbeda.

#### B. Overloading Constructor

Sama halnya dengan Overloading Method, Overloading Constructor juga mempunyai karakteristik yang serupa, namun hanya saja peletakannya yang berbeda yaitu pada constructornya saja.

```
public Lingkaran(int alas){
    //kode program
}
public Lingkaran(int alas, int tinggi){
    //kode program
}
```

### Pelaksanaan Percobaan

#### A. Overloading Method

Ketikkan program di bawah ini

1	import java.util.Scanner;
---	---------------------------

```

2 public class Overloading {
3     public static void HitungLuas(int a,int b){
4         int nilai = a*b;
5         System.out.println("maka hasil luas : "+nilai);
6     }
7     public static double HitungLuas(double value, double value2){
8         double nilai = value* value2;
9         return nilai;
10    }
11    public static void main(String[] args) {
12        Scanner in = new Scanner(System.in);
13        System.out.print("masukkan nilai integer 1 : ");
14        int integer1 = in.nextInt();
15        System.out.print("masukkan nilai integer 2 : ");
16        int integer2 = in.nextInt();
17        HitungLuas(integer1, integer2);
18        System.out.print("masukkan nilai double 1 : ");
19        double double1 = in.nextDouble();
20        System.out.print("masukkan nilai double 2 : ");
21        double double2 = in.nextDouble();
22        HitungLuas(integer1, integer2);
23        System.out.println("Maka          hasil          luas          :
24        "+HitungLuas(double1, double2));
25    }
26 }

```

## B. Overloading Constructor

Ketikkan program dibawah ini

```

1 public class lingkaran{
2     int alas, tinggi;
3     public lingkaran(int alas){
4         this.alas = alas;
5     }
6     public lingkaran(int alas, int tinggi){
7         this.alas = alas;
8         this.tinggi = tinggi;
9     }
10    public void setAlas(int alas){
11        this.alas = alas;
12    }
13    public void setTinggi(int tinggi){
14        this.tinggi = tinggi;
15    }
16    public int getAlas(){
17        return alas;
18    }
19    public int getTinggi(){
20        return tinggi;
21    }
22    public double hitungLuas(){
23        double hasil = (double) (getTinggi() *getAlas()) /2;
24        return hasil;
25    }

```

```

26     public void displayMessage(){
27         System.out.println("Hitung Luas : "+hitungLuas());
28     }
29 }

```

Ketikkan main classnya

```

1 public class LIngkaranMain{
2     public static void main(String[] args){
3         lingkaran l = new lingkaran(3);
4         l.setTinggi(10);
5         l.displayMessage();
6         lingkaran l2 = new lingkaran(4, 10);
7         l2.displayMessage();
8     }
9 }

```

## Data dan Analisis hasil percobaan

### A. Overloading Method

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

```

package OverloadingMethod;

import java.util.Scanner;

public class Overloading {
    public static void HitungLuas(int a,int b){
        int nilai = a*b;
        System.out.println("maka hasil luas : "+nilai);
    }

    public static double HitungLuas(double value, double value2){
        double nilai = value* value2;
        return nilai;
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print(s:"masukkan nilai integer 1 : ");
        int integer1 = in.nextInt();
        System.out.print(s:"masukkan nilai integer 2 : ");
        int integer2 = in.nextInt();
        HitungLuas(integer1, integer2);
        System.out.print(s:"masukkan nilai double 1 : ");
        double double1 = in.nextDouble();
        System.out.print(s:"masukkan nilai double 2 : ");
        double double2 = in.nextDouble();
        HitungLuas(integer1, integer2);
        System.out.println("Maka hasil luas : "+HitungLuas(double1, double2));
    }
}

```

Pada kode program OverloadingMethod tidak ditemukan kesalahan sehingga tidak ada yang perlu dibenahi

2. Jika pada baris 7, pada parameter double value dan double value2 di hapus dan di ganti menjadi int a dan int b apa yang terjadi? Jelaskan!

```

package OverloadingMethod;

import java.util.Scanner;

public class Overloading {
    public static void HitungLuas(int a,int b){
        int nilai = a*b;
        System.out.println("maka hasil luas : "+nilai);
    }

    public static double HitungLuas(int a, int b){
        double nilai = value* value2;
        return nilai;
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print(s:"masukkan nilai integer 1 : ");
        int integer1 = in.nextInt();
        System.out.print(s:"masukkan nilai integer 2 : ");
        int integer2 = in.nextInt();
        HitungLuas(integer1, integer2);
        System.out.print(s:"masukkan nilai double 1 : ");
        double double1 = in.nextDouble();
        System.out.print(s:"masukkan nilai double 2 : ");
        double double2 = in.nextDouble();
        HitungLuas(integer1, integer2);
        System.out.println("Maka hasil luas : "+HitungLuas(double1, double2));
    }
}

```

Jika parameter double value dan double value2 dihapus dan diganti menjadi int a dan int b maka kode program menjadi error. Karena dalam method HitungLuas terdapat kelas yang overloading bernilai integer sedangkan tipe data pada method tersebut tertulis double. Oleh karena itu, untuk mengatasi error tersebut maka harus merubah tipe data dalam method tersebut menjadi integer semua atau double semua. Sehingga, parameter tersebut tidak perlu diganti dan tetap menggunakan double value dan value 2.

```

public static void HitungLuas(double value, double value2){
    double nilai = value* value2;
    System.out.println("maka hasil luas : " +nilai);
}

```

3. Rubah method pada baris ketujuh menjadi method bertipe void, dan lakukan juga perubahan main method.

```

public static void HitungLuas(double value, double value2){
    double nilai = value* value2;
    System.out.println("maka hasil luas : " +nilai);
}

```

```

HitungLuas(integer1, integer2);
HitungLuas(double1, double2);
}
}

```

## B. Overloading Constructor

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

```
package OverloadingConstructor;

public class LingkaranMain {
    Run | Debug
    public static void main(String[] args){
        lingkaran l = new lingkaran(alas:3);
        l.setTinggi(tinggi:10);
        l.displayMessage();
        lingkaran l2 = new lingkaran(alas:4, tinggi:10);
        l2.displayMessage();
    }
}

public class lingkaran {
    int alas, tinggi;

    public lingkaran(int alas){
        this.alas = alas;
    }

    public lingkaran(int alas, int tinggi){
        this.alas = alas;
        this.tinggi = tinggi;
    }

    public void setAlas(int alas){
        this.alas = alas;
    }

    public void setTinggi(int tinggi){
        this.tinggi = tinggi;
    }

    public int getAlas(){
        return alas;
    }

    public int getTinggi(){
        return tinggi;
    }

    public double hitungLuas(){

```

2. Pada class lingkaran Tambahkan constructor dengan parameter int tinggi, apa yang terjadi dan jelaskan!

```
public lingkaran(int alas){
    this.alas = alas;
}

public lingkaran(int tinggi){
    this.tinggi = tinggi;
}
```

Jika pada class lingkaran ditambahkan constructor dengan parameter int tinggi maka kode program menjadi error. Karena pada konstruktor tersebut memiliki nama yang sama yaitu lingkaran sehingga menyebabkan error. Maka dari itu, jika ingin membuat konstruktor dengan nama yang sama maka harus memasukkan tipe data operasi yang ada di dalam konstruktor sebelumnya. Sehingga, seharusnya tidak perlu ditambahkan int tinggi

```

public lingkaran(int alas){
    this.alas = alas;
}

public lingkaran(int alas, int tinggi){
    this.alas = alas;
    this.tinggi = tinggi;
}

```

3. Pada class lingkaran tambahkan constructor dengan tipe data String alas dan String tinggi, kemudian tambahkan method untuk melakukan parsing atau perubahan tipe data dari String menjadi integer. Setelah itu pada method main lakukan instansiasi objek dengan nama objek Lstring dengan memanggil constructor bertipe data String. Jelaskan!

```

public lingkaran(String alas, String tinggi){
    rubahString(alas, tinggi);
}

public void rubahString(String alas, String tinggi){
    this.alas = Integer.parseInt(alas);
    this.tinggi = Integer.parseInt(tinggi);
}

```

```

public static void main(String[] args){
    lingkaran l = new lingkaran(alas:3);
    l.setTinggi(tinggi:10);
    l.displayMessage();
    lingkaran l2 = new lingkaran(alas:4, tinggi:10);
    l2.displayMessage();
    lingkaran Lstring = new lingkaran(alas:"5", tinggi:"5");
    Lstring.displayMessage();
}

```

## Tugas Praktikum

Melanjutkan dari bab 1 dan 2, buatlah method berikut :

1. Method `bacaFile("pathFile")`  
Fungsinya adalah memberi nilai pada kelas kalian dengan nilai atribut sesuai yang ada di file txt. Apabila kelas kalian pada bab 1 memiliki atribut judul dan penulis, maka pada file txt akan ada 2 nilai untuk judul dan nilai penulis (isi txt : DILAN 1990; Pidi Baiq)
2. Method `simpanFile("namaFile")`  
Fungsinya adalah menyimpan nilai atribut kelas.
3. Method `hitungRoyalti` dengan parameter harga 1 buku, yang mengembalikan nilai 10% dari penjualan bulan itu. (1 parameter)
4. Method `hitungRoyalti` dengan parameter harga 1 buku dan nilai persen yang ingin diambil dari penjualan bulan itu. (2 parameter)