

# 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!

**Tidak ada kesalahan**

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!

**Kalau pada baris 7 parameter double nya diganti dengan int, maka akan terjadi eror dengan method lain yang sudah ada dalam class Overloading. Karena parameternya sama dengan baris 4:**

```
public static void HitungLuas(int a, int b) {
```

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

- Method **HitungLuas** pada baris ketujuh diubah menjadi bertipe void.
- Pada method **HitungLuas** yang diubah menjadi bertipe void, saya mengganti **return nilai;** dengan **System.out.println("maka hasil luas : " + nilai);**.
- Pada method **main**, saya mengubah pemanggilan **HitungLuas(double1, double2);** agar sesuai dengan method yang sekarang bertipe void, tanpa menyertakan **System.out.println**.
- Saya menghapus pemanggilan **HitungLuas(integer1, integer2);** yang redundan di dalam **main**.

### B. Overloading Constructor

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

**Tidak ditemukan kesalahan**

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

**Dengan menamba konstruktor dengan parameter int tinggi maka, konstruktor baru yang dapat menerima nilai tinggi telah dibuat, dengan konstruktor ini, kita dapat menginputkan nilai tinggi bersamaan dengan nilai alas saat membuat objek lingkaran.**

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!

**Saya menambahkan konstruktor bertipe data String alas dan tinggi:**

```
public Lingkaran(String alas, String tinggi) {  
    this.alas = parseInt(alas);  
    this.tinggi = parseInt(tinggi);  
}
```

Konstruktor ini menerima input dalam bentuk String untuk menginputkan alas dan tinggi. Dalam konstruktor ini nilainya diubah menjadi int pakai method `parseInt(String value)` sebelum disimpan ke dalam objek Lingkaran.

**Penambahan Method `parseInt(String value)`:**

Saya menambahkan method `parseInt(String value)`. Method ini untuk mengubah nilai alas dan tinggi dari String menjadi integer.

**Penggunaan Constructor baru dalam Method Main:**

Di dalam method main, saya melakukan instansiasi objek Lingkaran baru dengan menggunakan constructor yang baru ditambahkan:

```
Lingkaran lString = new Lingkaran("5", "8");
```

Agar kita bisa menginput alas dan tinggi dalam bentuk String, lalu diubah menjadi int sebelum disimpan oleh objek Lingkaran

## 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)