

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

BAB : OVERLOAD DAN OVERLOADING METHOD
NAMA : NAMA PRAKTIKAN
NIM : NIM PRAKTIKAN
ASISTEN : Tengku Muhammad Rafi Rahardiansyah
Muhammad Bin Djafar Almasyhur
TGL PRAKTIKUM : 15 Maret 2023

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

Ketikkan main classnya

1	<code>public class LIngkaran</code>
2	<code> public</code>
3	
4	<code> l.setTinggi(10);</code>
5	<code> l.displayMessage();</code>
6	<code> lingkaran l2 = new lingkaran(4, 10);</code>
7	<code> l2.displayMessage();</code>
8	<code> }</code>
9	<code>}</code>

Data dan Analisis hasil percobaan

A. Overloading Method

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Source code

1	<code>Tulis source code di sini pake courier new 12</code>
---	--

Output

Penjelasan

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!

Source code

1	<code>Tulis source code di sini pake courier new 12</code>
---	--

Output

Penjelasan

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

Source code

1	<code>Tulis source code di sini pake courier new 12</code>
---	--

Output

Penjelasan

B. Overloading Constructor

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Source code

1	Tulis source code di sini pake courier new 12
---	---

Output

Penjelasan

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

Source code

1	Tulis source code di sini pake courier new 12
---	---

Output

Penjelasan

3. Pada class lingkaran tambahkan constrctor 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!

Source code

1	Tulis source code di sini pake courier new 12
---	---

Output

Penjelasan

████████████████████
