

**JOBSHEET 9**  
**OBJECT ORIENTED PROGRAMMING**



Arranged By :

Lenka Melinda Florianka

2241720074

Class 2I

**INFORMATION TECHNOLOGY**  
**D-IV INFORMATICS ENGINEERING**  
**MALANG STATE POLYTECHNIC**  
**2023**

## Practice 1

Code :

```
1 package jobsheet9;
2
3 public class PerkalianKu {
4     void perkalian (int a, int b){
5         System.out.println(a * b);
6     }
7
8     void perkalian(int a, int b, int c){
9         System.out.println(a * b * c);
10    }
11
12    public static void main(String args[]){
13        PerkalianKu objek = new PerkalianKu();
14
15        objek.perkalian(25, 43);
16        objek.perkalian(34, 23, 56);
17    }
18 }
```

Output :

```
PS C:\Dev\OOP> & '
e\User\workspaceSto
1075
43792
PS C:\Dev\OOP>
```

## Questions

1. Dari source coding diatas terletak dimanakah overloading?

Overloading terjadi pada method perkalian karena terdapat dua versi dari method tersebut dengan jumlah parameter yang berbeda.

## 2. Jika terdapat overloading ada berapa jumlah parameter yang berbeda?

Berbeda pada overloading tersebut. Pada versi pertama, method perkalian memiliki dua parameter (int a, int b), sedangkan pada versi kedua, method perkalian memiliki tiga parameter (int a, int b, int c).

### Practice 2

#### Code :

```
1 package jobsheet9;
2
3 public class PerkalianKu {
4     void perkalian (int a, int b){
5         System.out.println(a * b);
6     }
7
8     void perkalian(double a, double b){
9         System.out.println(a * b);
10    }
11
12    public static void main(String args[]){
13        PerkalianKu objek = new PerkalianKu();
14
15        objek.perkalian(25, 43);
16        objek.perkalian(34.56, 23.7);
17    }
18 }
```

#### Ouput :

```
PowerShell 7.3.9
PS C:\Dev\OOP> & 'C:\Program Files\PowerShell\PowerShell.exe' -File 'C:\Dev\OOP\jobsheet9\PerkalianKu.ps1'
1075
819.072
PS C:\Dev\OOP>
```

### Questions

#### 1. Dari source coding diatas terletak dimanakah overloading?

Dalam source code di atas, overloading terjadi pada metode perkalian. Overloading terjadi ketika ada beberapa metode dengan nama yang sama tetapi dengan tipe atau jumlah parameter yang berbeda.

#### 2. Jika terdapat overloading ada berapa tipe parameter yang berbeda?

Terdapat dua tipe parameter yang berbeda pada metode perkalian:

- Metode pertama memiliki dua parameter bertipe int.
- Metode kedua memiliki dua parameter bertipe double.

### Practice 3

#### Code :

```
1 package jobsheet9;
2
3 public class Ikan {
4     public void swim(){
5         System.out.println("Ikan bisa berenang");
6     }
7 }
8
9 class Piranha extends Ikan {
10     public void swim(){
11         System.out.println("Piranha bisa makan daging");
12     }
13 }
14
15 public class Fish {
16     public static void main(String[] args) {
17         Ikan a = new Ikan();
18         Ikan b = new Piranha();
19         a.swim();
20         b.swim();
21     }
22 }
23
```

#### Output :

```
age\2abd1bcf8435531430469a1c11
\OOP_5927979e\bin' 'jobsheet9.
Ikan bisa berenang
Piranha bisa makan daging
PS C:\Dev\OOP>
```

### Questions

#### 1. Dari source coding diatas terletak dimanakah overriding?

Dalam source code di atas, overriding terjadi pada metode swim yang dideklarasikan di kelas Piranha yang merupakan turunan dari kelas Ikan.

#### 2. Jabarkanlah apabila sourcoding diatas jika terdapat overriding?

Overriding terjadi ketika suatu metode di kelas turunan memiliki nama, tipe kembalian, dan parameter yang sama dengan metode di kelas induknya. Dalam kasus ini:

- Metode swim di kelas Ikan diberi implementasi yang mencetak "Ikan bisa berenang".
- Metode swim di kelas Piranha di-override untuk memberikan implementasi yang berbeda, yaitu mencetak "Piranha bisa makan daging".

Sebagai hasilnya, ketika objek b yang bertipe Ikan diinisialisasi dengan objek Piranha, saat memanggil metode swim, yang dipanggil adalah metode swim dari kelas Piranha, bukan dari kelas Ikan. Ini menunjukkan overriding, di mana metode dalam kelas turunan menggantikan metode dalam kelas induk dengan implementasi yang berbeda.

## Tasks

1. Overloading Implementasikan konsep overloading pada class diagram dibawah ini :

Code :

```
1 package jobsheet9;
2
3 public class Segitiga {
4     private int sudut;
5
6     public int totalSudut(int sudutA) {
7         this.sudut = 180 - sudutA;
8         return this.sudut;
9     }
10
11     public int totalSudut(int sudutA, int sudutB) {
12         this.sudut = 180 - (sudutA + sudutB);
13         return this.sudut;
14     }
15
16     public int keliling(int sisiA, int sisiB, int sisiC) {
17         return sisiA + sisiB + sisiC;
18     }
19
20     public double keliling(int sisiA, int sisiB) {
21         double c = Math.sqrt(Math.pow(sisiA, 2) + Math.pow(sisiB, 2));
22         return c;
23     }
24
25     public static void main(String[] args) {
26         Segitiga segitiga = new Segitiga();
27
28         int totalSudut1 = segitiga.totalSudut(60);
29         System.out.println("Total Sudut 1: " + totalSudut1);
30
31         int totalSudut2 = segitiga.totalSudut(60, 30);
32         System.out.println("Total Sudut 2: " + totalSudut2);
33
34         int kelilingABC = segitiga.keliling(3, 4, 5);
35         System.out.println("Keliling ABC: " + kelilingABC);
36
37         double kelilingAB = segitiga.keliling(3, 4);
38         System.out.println("Keliling AB: " + kelilingAB);
39     }
40 }
41
```

Output :

```
PowerShell 7.3.9
PS C:\Dev\OOP> & 'C:\Program Files\PowerShell\PowerShell.exe' -c 'C:\Dev\OOP\Jobsheet9\Jobsheet9.exe'
Total Sudut 1: 120
Total Sudut 2: 90
Keliling ABC: 12
Keliling AB: 5.0
PS C:\Dev\OOP>
```

2. Implementasikan class diagram dibawah ini dengan menggunakan teknik dynamic method dispatch :

Code :

```
1 package jobsheet9;
2
3 public class Main {
4     public static void main(String[] args) {
5         Manusia manusia1 = new Dosen();
6         Manusia manusia2 = new Mahasiswa();
7
8         manusia1.bernafas();
9         manusia1.makan();
10
11         manusia2.bernafas();
12         manusia2.makan();
13     }
14 }
```

```
1 package jobsheet9;
2
3 class Manusia {
4     public void bernafas() {
5         System.out.println("Manusia sedang bernafas");
6     }
7
8     public void makan() {
9         System.out.println("Manusia sedang makan");
10    }
11 }
12
13 class Dosen extends Manusia {
14
15     public void makan() {
16         System.out.println("Dosen sedang makan");
17     }
18
19     public void lembur() {
20         System.out.println("Dosen sedang lembur");
21     }
22 }
23
24 class Mahasiswa extends Manusia {
25
26     public void makan() {
27         System.out.println("Mahasiswa sedang makan");
28     }
29
30     public void tidur() {
31         System.out.println("Mahasiswa sedang tidur");
32     }
33 }
```

Output :

```
e\User\workspaceStorage\2a
Manusia sedang bernafas
Dosen sedang makan
Manusia sedang bernafas
Mahasiswa sedang makan
PS C:\Dev\OOP>
```