

# **OBJECT ORIENTED PROGRAMMING**

## **JOBSHEET 9 OVERLOADING & OVERRIDING**



Nama :  
Gastiadirijal N.K.

NIM:  
2241720001

Kelas2I

**STATE POLYTECHNIC OF MALANG**  
**2023**

## Exercise

```
1 public class Perkalianku {
2     void perkalian(int a, int b) {
3         System.out.println(a * b);
4     }
5
6     void perkalian(int a, int b, int c) {
7         System.out.println(a * b * c);
8     }
9
10 }
```

- 1.
2. 2 parameter dan 3 parameter

```
1 public class Perkalianku {
2     void perkalian(int a, int b) {
3         System.out.println(a * b);
4     }
5
6     void perkalian(double a, double b) {
7         System.out.println(a * b);
8     }
9
10 }
```

- 3.
4. Int dan Double

```
1 class ikan {
2     public void swim() {
3         System.out.println(x:"Ikan bisa berenang");
4     }
5 }
```

```
1 class Piranha extends ikan {
2     public void swim() {
3         System.out.println(x:"Piranha bisa makan daging");
4     }
5 }
```

6. - Kelas Piranha adalah subclass dari Ikan (Piranha extends Ikan), sehingga Piranha mewarisi metode swim() dari Ikan.
  - Metode swim() dalam kelas Piranha memiliki implementasi yang berbeda dibandingkan dengan metode swim() dalam kelas Ikan.
  - Ketika Anda membuat objek dari kelas Piranha dan memanggil metode swim(), metode yang dipanggil adalah versi dari kelas Piranha karena ada overriding, yaitu implementasi dari kelas Piranha yang digunakan, bukan dari kelas Ikan.
  - Dengan kata lain, metode swim() di kelas Piranha meng-override (mengganti) perilaku metode yang sama yang diwarisi dari kelas Ikan.

## Tugas

1.

```
1 public class Segitiga {
2
3     public int totalSudut(int sudutA) {
4         return 180 - sudutA;
5     }
6
7     public int totalSudut(int sudutA, int sudutB) {
8         return 180 - (sudutA + sudutB);
9     }
10
11     public int keliling(int sisiA, int sisiB, int sisiC) {
12         return sisiA + sisiB + sisiC;
13     }
14
15     public double keliling(int sisiA, int sisiB) {
16         double sisiC = Math.sqrt(Math.pow(sisiA, 2) + Math.pow(sisiB, 2));
17         return sisiA + sisiB + sisiC;
18     }
19 }
```

```
1 public class MainSegitiga {
2     Run | Debug
3     public static void main(String[] args) {
4         Segitiga segitiga = new Segitiga();
5
6         int totalSudut1 = segitiga.totalSudut(sudutA:60);
7         System.out.println("Total Sudut 1: " + totalSudut1);
8
9         int totalSudut2 = segitiga.totalSudut(sudutA:60, sudutB:45);
10        System.out.println("Total Sudut 2: " + totalSudut2);
11
12        int keliling1 = segitiga.keliling(sisiA:3, sisiB:4, sisiC:5);
13        System.out.println("Keliling 1: " + keliling1);
14
15        double keliling2 = segitiga.keliling(sisiA:3, sisiB:4);
16        System.out.println("Keliling 2: " + keliling2);
17    }
18 }
```

```
Total Sudut 1: 120
Total Sudut 2: 75
Keliling 1: 12
Keliling 2: 12.0
```

2.

```
Manusia.java > Manusia > makan()
1 class Manusia {
2     public void bernafas() {
3         System.out.println(x:"Manusia bernafas");
4     }
5
6     public void makan() {
7         System.out.println(x:"Manusia makan");
8     }
9 }
```

```
1 class Mahasiswa2 extends Manusia {
2     public void makan() {
3         System.out.println(x:"Mahasiswa makan");
4     }
5
6     public void tidur() {
7         System.out.println(x:"Mahasiswa tidur");
8     }
9 }
```

```
1 class Dosen2 extends Manusia {
2     public void makan() {
3         System.out.println(x:"Dosen makan");
4     }
5
6     public void lembur() {
7         System.out.println(x:"Dosen lembur");
8     }
9 }
10
```

```
1 public class MainManusia {
2     Run | Debug
3     public static void main(String[] args) {
4         Manusia manusia1 = new Dosen2();
5         Manusia manusia2 = new Mahasiswa2();
6
7         manusia1.bernafas();
8         manusia1.makan();
9         manusia2.bernafas();
10        manusia2.makan();
11    }
```

```
Manusia bernafas
Dosen makan
Manusia bernafas
Mahasiswa makan
```