PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK
"Overloading dan Overriding"

DOSEN PEMBIMBING

Septian Enggar Sukmana, S.Pd., M.T.

Disusun oleh :

Muhammad Reza Khatami

(2041720076)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG 2021

Jobsheet Pertemuan 7

1. Latihan

```
public class PerkalianKu {

 void perkalian(int a, int b){

  System.out.println(a * b);

 }

 void perkalian(int a, int b, int c){

  System.out.println(a * b * c);

 }

 public static void main(String args []){

  PerkalianKu objek = new PerkalianKu();

  objek.perkalian(25, 43);
  objek.perkalian(34, 23, 56);
 }
}
```

1. Dari source coding diatas terletak dimanakah overloading?
   Jawab :
   Overloading terdapat pada void perkalian, karena nama method sama tetapi parameter berbeda

```
void perkalian(int a, int b){

 System.out.println(a * b);

}

void perkalian(int a, int b, int c){

 System.out.println(a * b * c);

}
```

2. Jika terdapat overloading ada berapa jumlah parameter yang berbeda?
   Jawab : Pada Kode tersebut terdapat 1 parameter berbeda, yaitu pada method perkalian pertama terdapat 2 parameter (int a , int b) dan perkalian kedua terdapat 3 parameter yaitu (int a, int b, int c)

```
public class PerkalianKu {

 void perkalian(int a, int b){

  System.out.println(a * b);

 }

 void perkalian(double a, double b){

  System.out.println(a * b);

 }

 public static void main(String args []){

  PerkalianKu objek = new PerkalianKu();

  objek.perkalian(25, 43);
  objek.perkalian(34.56, 23.7);
 }
}
```

3. Dari source coding diatas terletak dimanakah overloading?
   Jawab :
   Overloading terdapat pada void perkalian, karena nama method sama tetapi parameter berbeda

```
 void perkalian(int a, int b){

  System.out.println(a * b);

 }

 void perkalian(double a, double b){

  System.out.println(a * b);

 }
```

4. Jika terdapat overloading ada berapa tipe parameter yang berbeda?
   Jawab : Pada Kode tersebut terdapat 2 parameter ang berbeda yaitu pada parameter pertama menggunakan tipe data int dan pada parameter ke 2 menggunakan tipe data double

```
  class Ikan{
   public void swim(){
      System.out.println("Ikan bisa berenang");
    }
}
class Piranha extends Ikan{
    public void swim(){
      System.out.println("Piranha bisa makan daging");
    }
}
public class Fish {
     public static void main(String[] args) {
         Ikan a = new Ikan();
        Ikan b = new Piranha();
        a.swim();
        b.swim();
     }
    }
```

5. Dari source coding diatas terletak dimanakah overriding?
   Jawab : Overriding terletak pada method swim karena pada class Piranha menggunakan nama method yang sama dengan parent class yaitu class Ikan
6. Jabarkanlah apabila sourcoding diatas jika terdapat overriding?
   Jawab : Terdapat overriding karena pada class Ikan yaitu parent class terdapat method swim dan pada class Piranha yaitu child class terdapat method swim dimana method tersebut memiliki nama yang sama dengan method yang ada pada class Ikan sehingga class Piranha menggunakan override pada method swim yang berada pada class Ikan.

2. Tugas

1. Tugas Overloading

- Segitiga.java

```java
package TugasOverloading;

public class Segitiga {

    private int sudut;

    public int totalSudut(int sudutA) {
        sudut = 180 - sudutA;
        return sudut;
    }

    public int totalSudut(int sudutA, int sudutB) {
        sudut = 180 - (sudutA + sudutB);
        return sudut;
    }

    public int keliling(int sisiA, int sisiB, int sisiC) {
        int keliling = sisiA + sisiB + sisiC;
        return keliling;
    }

    public double keliling(int sisiA, int sisiB) {
        double sisiC = Math.sqrt(Math.pow(sisiA, 2) + Math.pow(sisiB, 2));
        return sisiA + sisiB + sisiC;
    }
}
```

- MainTugasOverloading.java

```java
package TugasOverloading;

public class MainTugasOverloading {

    public static void main(String[] args) {
        Segitiga sgt1 = new Segitiga();
        System.out.println("Sudut Pertama : " + sgt1.totalSudut(10));
        System.out.println("Sudut Kedua : " + sgt1.totalSudut(6, 10));
        System.out.println("Keliling Segitiga Pertama : " + sgt1.keliling(10, 20, 15));
        System.out.println("Keliling Segitiga Kedua : " + sgt1.keliling(3, 4));
    }
}
```

- Output

```
run:
Sudut Pertama : 170
Sudut Kedua : 164
Keliling Segitiga Pertama : 45
Keliling Segitiga Kedua : 12.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Tugas Overriding
   - Manusia.java

```java
package TugasOverriding;

public class Manusia {

    public void bernafas() {
        System.out.println("Manusia bernafas");
    }

    public void makan() {
        System.out.println("Manusia makan");
    }
}
```

   - Dosen.java

```java
package TugasOverriding;

public class Dosen extends Manusia {

    public void makan() {
        System.out.println("Dosen makan");
    }

    public void lembur() {
        System.out.println("Dosen lembur");
    }
}
```

   - Mahasiswa.java

```java
package TugasOverriding;

public class Mahasiswa extends Manusia{

    public void makan(){
        System.out.println("Mahasiswa makan");
    }

    public void tidur(){
        System.out.println("Mahasiswa tidur");
    }
}
```

- MainTugasOverriding.java

```java
package TugasOverriding;

public class MainTugasOverriding {

    public static void main(String[] args) {
        Manusia man1 = new Manusia();
        man1.bernafas();
        man1.makan();

        man1 = new Dosen();
        man1.makan();

        man1 = new Mahasiswa();
        man1.makan();

        Mahasiswa mhs = new Mahasiswa();
        mhs.tidur();
        mhs.makan();
    }
}
```

- Output

```
run:
Manusia bernafas
Manusia makan
Dosen makan
Mahasiswa makan
Mahasiswa tidur
Mahasiswa makan
BUILD SUCCESSFUL (total time: 0 seconds)
```