

Praktikum 5

Kelas Abstrak Dan Interface

1. Tujuan

- Mahasiswa mampu menerapkan konsep kelas abstrak dan implementasinya ke dalam program.
- Mahasiswa mampu menerapkan konsep interface dan implementasinya ke dalam program

2. Landasan Teori

Kelas abstrak merupakan kelas yang memiliki metode abstrak, yaitu metode tanpa implementasi. Hal tersebut dilakukan untuk memfasilitasi kelas yang lain yang memiliki implementasi yang berbeda-beda. Contoh implementasi kelas abstrak dalam pemrograman Java dari notasi algoritma adalah sebagai berikut :

```
{notasi algoritma}  
abstract class BangunDatar  
    luas : real,protected  
    abstract function hitungLuas() → real  
end abstract class BangunDatar
```

```
{program Java}  
public abstract class BangunDatar{  
    protected double luas;  
    public abstract double hitungLuas();  
}
```

Interface adalah kontrak yang harus dijalankan oleh sebuah kelas ketika kelas tersebut membuat implementasi sebuah interface. Contoh implementasi interface dalam pemrograman Java dari notasi algoritma adalah sebagai berikut :

```
{notasi algoritma}  
interface IArea  
    function hitungLuas() → real  
end interface IArea  
  
class BujurSangkar realize IArea  
    function hitungLuas() → real  
    {.....}  
end class BujurSangkar
```

```
{Program Java}  
public interface IArea{  
    public double hitungLuas();  
}  
  
class BujurSangkar implements IArea{  
    public double hitungLuas(){  
        //....  
    }  
}
```

3. Langkah Praktikum

3.1. Kelas Abstrak

Implementasikan 3 kelas berikut ini :

```
1  /**
2   * BangunDatar.java 12/11/11
3   * Penulis : Panji Wisnu Wirawan
4   * Deskripsi : kelas abstrak, berisi abstraksi bangun datar
5   *
6   */
7
8  public abstract class BangunDatar{
9      protected double luas;
10
11      public abstract double hitungLuas(double sisi);
12
13      public void setLuas(double l){
14          luas = l;
15      }
16
17      public double getLuas(){
18          return luas;
19      }
20  }
```

```
1  /**
2   * BujurSangkar.java 12/11/11
3   * Penulis : Panji Wisnu Wirawan
4   * Deskripsi : kelas yang membuat implementasi metode abstrak
5   *             pada bangun datar
6   *
7   */
8
9  public class BujurSangkar extends BangunDatar{
10      public double hitungLuas(double sisi){
11          luas = sisi*sisi;
12          return luas;
13      }
14  }
```

```

1  /**
2   * MBujurSangkar.java 12/11/11
3   * Penulis : Panji Wisnu Wirawan
4   * Deskripsi : kelas yang mengimplementasikan cara menghitung
5   *             luas bujur sangkar
6   *
7   */
8   import java.util.Scanner;
9
10  class MBujurSangkar{
11      public static void main(String[] args){
12          Scanner scan = new Scanner(System.in);
13          BujurSangkar bs = new BujurSangkar();
14          System.out.print("Masukkan sisi bujur sangkar : ");
15          double sisi = scan.nextDouble();
16          System.out.println("Luas bujur sangkar dengan sisi "+sisi+
17                             " satuan adalah "+bs.hitungLuas(sisi));
18      }
19  }

```

Praktekkan : apa yang terjadi apabila kelas BujurSangkar tidak membuat implementasi metode abstrak yang ada pada kelas BangunDatar ? jelaskan !

3.2. Interface

Implementasikan ketiga file berikut :

```

1  /**
2   * IArea.java 12/11/11
3   * Penulis : Panji Wisnu Wirawan
4   * Deskripsi : interface yang mendefinisikan kontrak
5   *             bahwa setiap kelas yang membuat implementasi
6   *             dari interface ini harus memiliki cara untuk
7   *             menghitung luas
8   *
9   */
10
11  public interface IArea{
12      public double hitungLuas();
13  }

```

```

1  /**
2   * Lingkaran.java 12/11/11
3   * Penulis : Panji Wisnu Wirawan
4   * Deskripsi : kelas implementasi IArea berupa
5   *             cara menghitung luas lingkaran
6   *
7   */
8   //mengambil konstanta yang ada di kelas java.lang.Math
9   import static java.lang.Math.PI;
10
11  class Lingkaran implements IArea{
12      private double jejari;
13
14      public Lingkaran(double r){
15          jejari = r;
16      }
17
18      public double hitungLuas(){
19          return PI*jejari*jejari;
20      }
21  }

```

```

1  /**
2   * MLingkaran.java 12/11/11
3   * Penulis : Panji Wisnu Wirawan
4   * Deskripsi : implementasi cara menghitung luas lingkaran
5   *
6   */
7
8  public class MLingkaran{
9      public static void main(String[] args){
10         Lingkaran l = new Lingkaran(10.2);
11         System.out.println("Luas lingkaran dengan"+
12             "jejari 10.2 satuan adalah "+l.hitungLuas());
13     }
14 }

```

Praktekan:

Buatlah MLingkaran sedemikian hingga dapat menerima input jejari dari pengguna !