MODUL 8

MULTI INHARITANCE

A. Tujuan

1. Mahasiswa memahami dan bisa mengimplementasikan konsep penurunan sifat dari beberapa class induk dengan mengguanakan Interface.
2. Memahami jenis Multi Inheritance

B. Dasar Teori

B.1 *Abstract Method*

Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan. Abstraksi merupakan kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan (Ristianti, 2019 : 19)

Abstract method adalah metode dalam sebuah kelas yang dideklarasikan tanpa implementasi konkret.

```
public abstract class Bentuk {
public abstract double hitungLuas();
}

public class PersegiPanjang extends Bentuk {
private double panjang;
private double lebar;

public PersegiPanjang(double panjang, double lebar) {
this.panjang = panjang;
this.lebar = lebar;
}
```

B.2 *Interface*

Interface adalah merupakan sesuatu yang berbeda dari class, yang hanya berisi deklarasi method tanpa memiliki implementasi dan semua property yang dimilikinya bersifat final. Interface mirip dengan class abstrak, tetapi interface tidak terikat dengan class hierarki.Interface mendefinisikan sebuah(signature) dari sebuah kumpulan method tanpa tubuh.Interface perlu digunakan dalam kerja team, karena programer yang lain tidak perlu tahu bagaimana detail code ditulis.

Pembuatan interface

1. Buat interface dengan nama Interface

```
short x;
int umur;
float gaji;
double data;
```

```
interface Interface {

    public int hitungLuas();
    public int hitungKeliling();
}
```

InterfaceInterface diatas merupakan perubahan dari class Abstrak berikut

```
short x;
int umur;
float gaji;
double data;
```

```
public abstract class Abstrak {

    abstract int hitungLuas();

    int hitungKeliling(){
        int x = 0;
        return x;
    }
}
```

Bandingkan interface dan class Abstrak

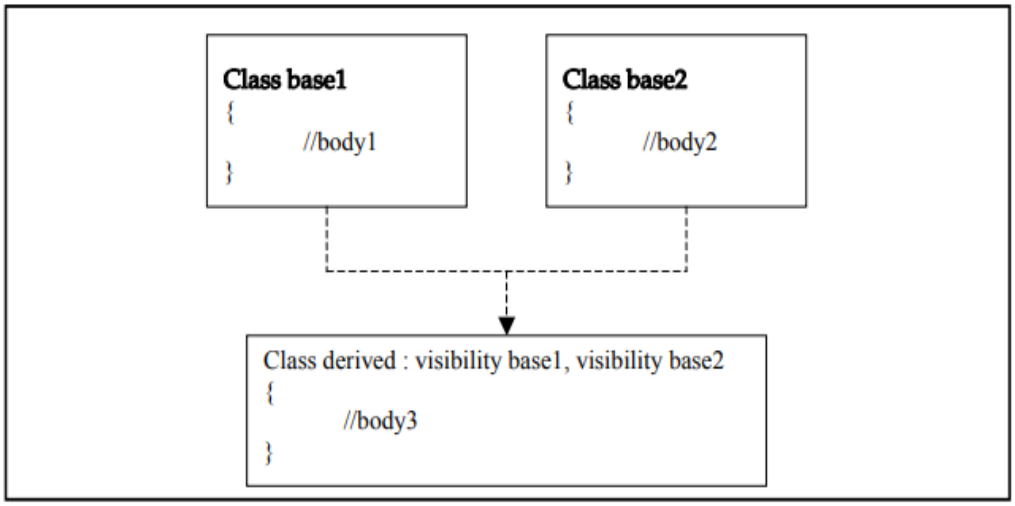
2. Buat Class Persegei yang merupakan implements dari Interface

```
short x;
int umur;
float gaji;
double data;
```

```
11  public class Persegi implements Interface{
12      int sisi;
13
14      public Persegi(int s) {
15          this.sisi = s;
16      }
17
18      @Override
19      public int hitungLuas() {
20          //method ini harus diberi implementasi
21          int luas = sisi * sisi;
22          return luas;
23      }
24
25      @Override
26      public int hitungKeliling() {
27          return 0;
28      }
29
30      public static void main(String[] args) {
31          Persegi p = new Persegi(5);
32          int luas = p.hitungLuas();
33          System.out.println(luas);
34          System.out.println(p.hitungKeliling());
35      }
36  }
```

B.3 *Multiple Inheritance*

Sebuah kelas dapat mewarisi atribut dari dua kelas atau lebih. Mekanisme ini dikenal sebagai *Multiple Inheritance*. *Multiple Inheritance* memungkinkan kita untuk menggabungkan fitur dari beberapa fitur yang sudah adakelas sebagai titik awal untuk mendefinisikan kelas baru. (dubey, 2015 : 178-179)



Gambar 8.1 *Multiple Inheritance*

Dimana visibilitas mengacu pada penentu akses yaitu publik, pribadi atau dilindungi. Berikut program menunjukkan *Multiple Inheritance*

```
interface Shape {
    double getArea();
    double getPerimeter();
}

interface Color {
    String getColor();
}

class Circle implements Shape, Color {
    private double radius;
    private String color;

    public Circle(double radius, String color) {
        this.radius = radius;
        this.color = color;
    }

    @Override
    public double getArea() {
        return Math.PI * radius * radius;
    }

    @Override
    public double getPerimeter() {
        return 2 * Math.PI * radius;
    }

    @Override
    public String getColor() {
        return color;
```

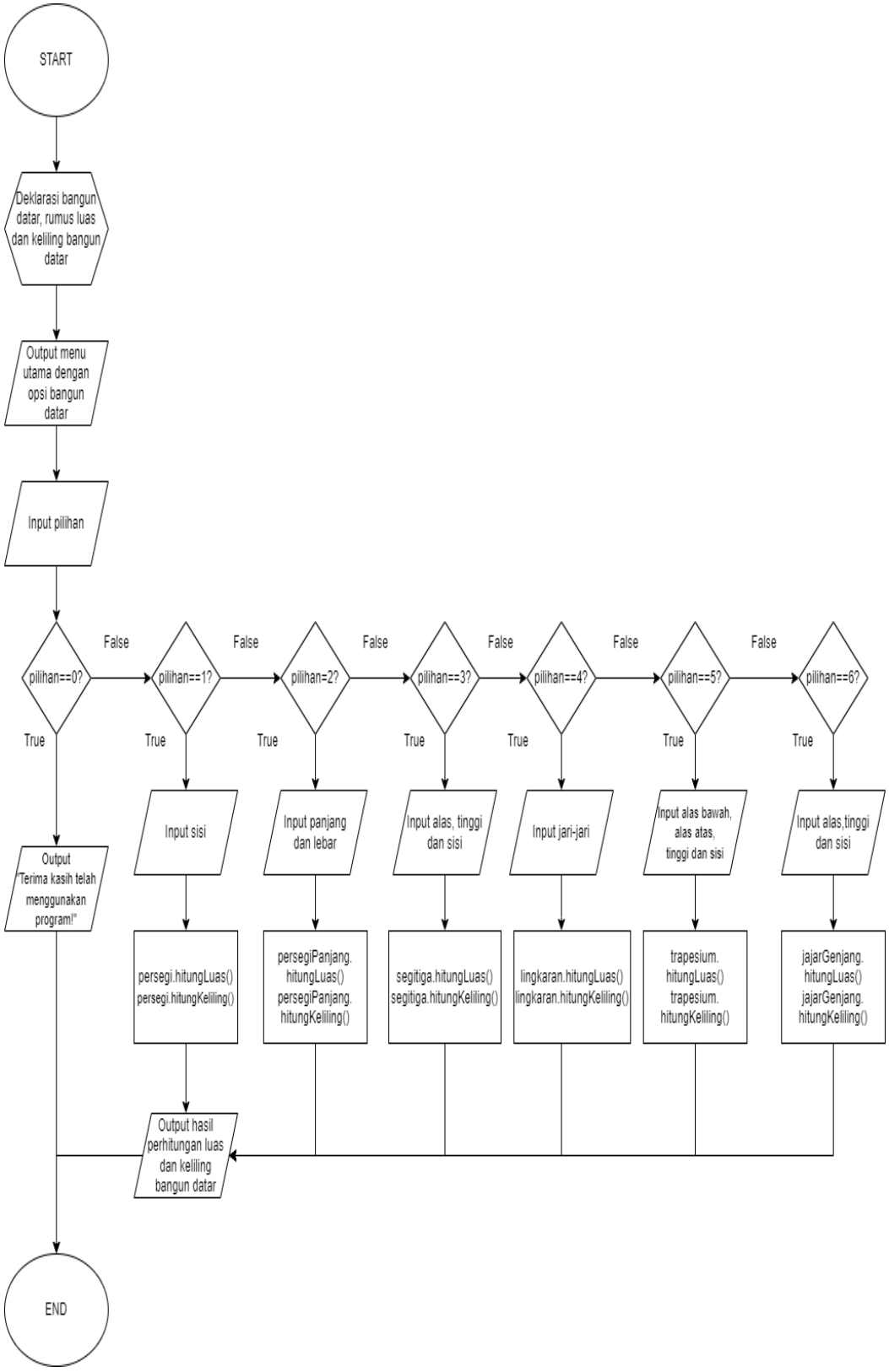
```
        }
    }

    public class Main {
        public static void main(String[] args) {
            Circle circle = new Circle(5.0, "Red");
            System.out.println("Area:          "          +
    circle.getArea());
            System.out.println("Perimeter:       "       +
    circle.getPerimeter());
            System.out.println("Color:         "         +
    circle.getColor());
        }
}
```

C. Studi Kasus / Permasalahan

1. Buatlah aplikasi kalkulator untuk menghitung luas dan keliling bangun datar yang terdiri dari persegi, persegi panjang, segitiga, lingkaran, trapesium dan jajar genjang. Program dibuat dengan dinamis dengan menggunakan menu seperti switch case atau if else.

D. Hasil (*flowchart, script* program, hasil program)

D.1. Flowchart



D.2. *Script Program*

```
package StudiKasus;

interface BentukLuas {
    double hitungLuas();
}
```

```
package StudiKasus;

interface BentukKeliling {
    double hitungKeliling();
}
```

```
package StudiKasus;

abstract class BangunDatar implements BentukLuas,
BentukKeliling {
    public double hitungLuas(){
        return 0.0;
    }

    public double hitungKeliling(){
        return 0.0;
    }
}
```

```
package StudiKasus;

class Persegi extends BangunDatar {
    private double sisi;

    public Persegi(double sisi){
        this.sisi = sisi;
    }

    @Override
    public double hitungLuas() {
        return sisi * sisi;
    }

    @Override
    public double hitungKeliling() {
        return 4 * sisi;
    }
}
```

```
package StudiKasus;

class PersegiPanjang extends BangunDatar {
    private double panjang;
    private double lebar;

    public PersegiPanjang(double panjang, double lebar){
        this.panjang = panjang;
        this.lebar = lebar;
    }

    @Override
```

```
    public double hitungLuas(){
        return panjang * lebar;
    }

    @Override
    public double hitungKeliling(){
        return 2 * (panjang + lebar);
    }
}
```

```
package StudiKasus;

class Segitiga extends BangunDatar{
    private double tinggi;
    private double alas;
    private double sisi1;
    private double sisi2;
    private double sisi3;

    public Segitiga(double alas, double tinggi, double sisi1,
double sisi2, double sisi3) {
        this.alas = alas;
        this.tinggi = tinggi;
        this.sisi1 = sisi1;
        this.sisi2 = sisi2;
        this.sisi3 = sisi3;
    }

    @Override
    public double hitungLuas(){
        return 0.5 * alas * tinggi;
    }

    @Override
    public double hitungKeliling(){
        return sisi1 + sisi2 + sisi3;
    }
}
```

```
package StudiKasus;

class Lingkaran extends BangunDatar {
    private double jariJari;

    public Lingkaran(double jariJari){
        this.jariJari = jariJari;
    }

    @Override
    public double hitungLuas(){
        return Math.PI * jariJari * jariJari;
    }

    @Override
    public double hitungKeliling(){
        return 2 * Math.PI * jariJari;
    }
```

```
}
```

```
package StudiKasus;

class Trapesium extends BangunDatar {
    private double alasAtas;
    private double alasBawah;
    private double tinggi;
    private double sisi1;
    private double sisi2;

    public Trapesium(double alasAtas, double alasBawah, double
tinggi, double sisi1, double sisi2) {
        this.alasAtas = alasAtas;
        this.alasBawah = alasBawah;
        this.tinggi = tinggi;
        this.sisi1 = sisi1;
        this.sisi2 = sisi2;
    }

    @Override
    public double hitungLuas() {
        return 0.5 * (alasAtas + alasBawah) * tinggi;
    }

    @Override
    public double hitungKeliling() {
        return alasAtas + alasBawah + sisi1 + sisi2;
    }
}
```

```
package StudiKasus;

class JajarGenjang extends BangunDatar {
    private double alas;
    private double tinggi;
    private double sisi1;
    private double sisi2;

    public JajarGenjang(double alas, double tinggi, double
sisi1, double sisi2) {
        this.alas = alas;
        this.tinggi = tinggi;
        this.sisi1 = sisi1;
        this.sisi2 = sisi2;
    }

    @Override
    public double hitungLuas() {
        return alas * tinggi;
    }

    @Override
    public double hitungKeliling() {
        return 2 * (sisi1 + sisi2);
    }
}
```

```
package StudiKasus;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("====================");
            System.out.println("Pilih bangun datar:");
            System.out.println("1. Persegi");
            System.out.println("2. Persegi Panjang");
            System.out.println("3. Segitiga");
            System.out.println("4. Lingkaran");
            System.out.println("5. Trapesium");
            System.out.println("6. Jajar Genjang");
            System.out.println("0. Keluar");
            System.out.println("====================");

            System.out.print("Masukkan pilihan: ");
            int pilihan = scanner.nextInt();

            if (pilihan == 0) {
                System.out.println("Terima kasih telah
menggunakan program!");
                break;
            }

            switch (pilihan) {
                case 1:
                    System.out.print("Masukkan panjang sisi
persegi: ");
                    double sisiPersegi = scanner.nextDouble();
                    Persegi persegi = new Persegi(sisiPersegi);
                    System.out.println("Luas\t:
"+persegi.hitungLuas());
                    System.out.println("Keliling:
"+persegi.hitungKeliling());
                    break;

                case 2:
                    System.out.print("Masukkan panjang persegi
panjang: ");
                    double panjangPersegiPanjang =
scanner.nextDouble();
                    System.out.print("Masukkan lebar persegi
panjang: ");
                    double lebarPersegiPanjang =
scanner.nextDouble();
                    PersegiPanjang persegiPanjang = new
PersegiPanjang(panjangPersegiPanjang, lebarPersegiPanjang);
                    System.out.println("Luas\t: " +
persegiPanjang.hitungLuas());
                    System.out.println("Keliling: " +
```

```
persegiPanjang.hitungKeliling());
                    break;

                case 3:
                    System.out.print("Masukkan alas segitiga:
");
                    double alasSegitiga = scanner.nextDouble();
                    System.out.print("Masukkan tinggi segitiga:
");
                    double tinggiSegitiga =
scanner.nextDouble();
                    System.out.print("Masukkan sisi1 segitiga:
");
                    double sisi1Segitiga =
scanner.nextDouble();
                    System.out.print("Masukkan sisi2 segitiga:
");
                    double sisi2Segitiga =
scanner.nextDouble();
                    System.out.print("Masukkan sisi3 segitiga:
");
                    double sisi3Segitiga =
scanner.nextDouble();
                    Segitiga segitiga = new
Segitiga(alasSegitiga, tinggiSegitiga, sisi1Segitiga,
sisi2Segitiga, sisi3Segitiga);
                    System.out.println("Luas\t:
"+segitiga.hitungLuas());
                    System.out.println("Keliling:
"+segitiga.hitungKeliling());
                    break;

                case 4:
                    System.out.print("Masukkan jari-jari
lingkaran: ");
                    double jariJariLingkaran =
scanner.nextDouble();
                    Lingkaran lingkaran = new
Lingkaran(jariJariLingkaran);
                    System.out.println("Luas\t: " +
lingkaran.hitungLuas());
                    System.out.println("Keliling: " +
lingkaran.hitungKeliling());
                    break;

                case 5:
                    System.out.print("Masukkan alas atas
trapesium: ");
                    double alasAtasTrapesium =
scanner.nextDouble();
                    System.out.print("Masukkan alas bawah
trapesium: ");
                    double alasBawahTrapesium =
scanner.nextDouble();
                    System.out.print("Masukkan tinggi
```

```
trapesium: ");
                    double tinggiTrapesium =
scanner.nextDouble();
                    System.out.print("Masukkan sisi1 trapesium:
");
                    double sisi1Trapesium =
scanner.nextDouble();
                    System.out.print("Masukkan sisi2 trapesium:
");
                    double sisi2Trapesium =
scanner.nextDouble();
                    Trapesium trapesium = new
Trapesium(alasAtasTrapesium, alasBawahTrapesium,
tinggiTrapesium, sisi1Trapesium, sisi2Trapesium);
                    System.out.println("Luas\t: " +
trapesium.hitungLuas());
                    System.out.println("Keliling: " +
trapesium.hitungKeliling());
                    break;

                case 6:
                    System.out.print("Masukkan alas jajar
genjang: ");
                    double alasJajarGenjang =
scanner.nextDouble();
                    System.out.print("Masukkan tinggi jajar
genjang: ");
                    double tinggiJajarGenjang =
scanner.nextDouble();
                    System.out.print("Masukkan sisi1 jajar
genjang: ");
                    double sisi1JajarGenjang =
scanner.nextDouble();
                    System.out.print("Masukkan sisi2 jajar
genjang: ");
                    double sisi2JajarGenjang =
scanner.nextDouble();
                    JajarGenjang jajarGenjang = new
JajarGenjang(alasJajarGenjang, tinggiJajarGenjang,
sisi1JajarGenjang, sisi2JajarGenjang);
                    System.out.println("Luas\t: " +
jajarGenjang.hitungLuas());
                    System.out.println("Keliling: " +
jajarGenjang.hitungKeliling());
                    break;
            }
        }
        scanner.close();
    }
}
```

D.3. Hasil Program

```
====================
Pilih bangun datar:
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Lingkaran
5. Trapesium
6. Jajar Genjang
0. Keluar
====================
Masukkan pilihan: 1
Masukkan panjang sisi persegi: 5
Luas    : 25.0
Keliling: 20.0
====================
Pilih bangun datar:
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Lingkaran
5. Trapesium
6. Jajar Genjang
0. Keluar
====================
Masukkan pilihan: 4
Masukkan jari-jari lingkaran: 10
Luas    : 314.1592653589793
Keliling: 62.83185307179586
====================
Pilih bangun datar:
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Lingkaran
5. Trapesium
6. Jajar Genjang
0. Keluar
====================
Masukkan pilihan: 0
Terima kasih telah menggunakan program!
```

E. Analisa

F. Kesimpulan

DAFTAR PUSTAKA