IK22

Praktikum Object dan Class Bagian-1

DiSusun Oleh :

Iklima Mardiana, 2008765

PROGRAM STUDI PENDIDIKAN ILMU KOMPUTER

FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PENDIDIKAN INDONESIA

25 MARET 2022

Implementasi dan Hasil

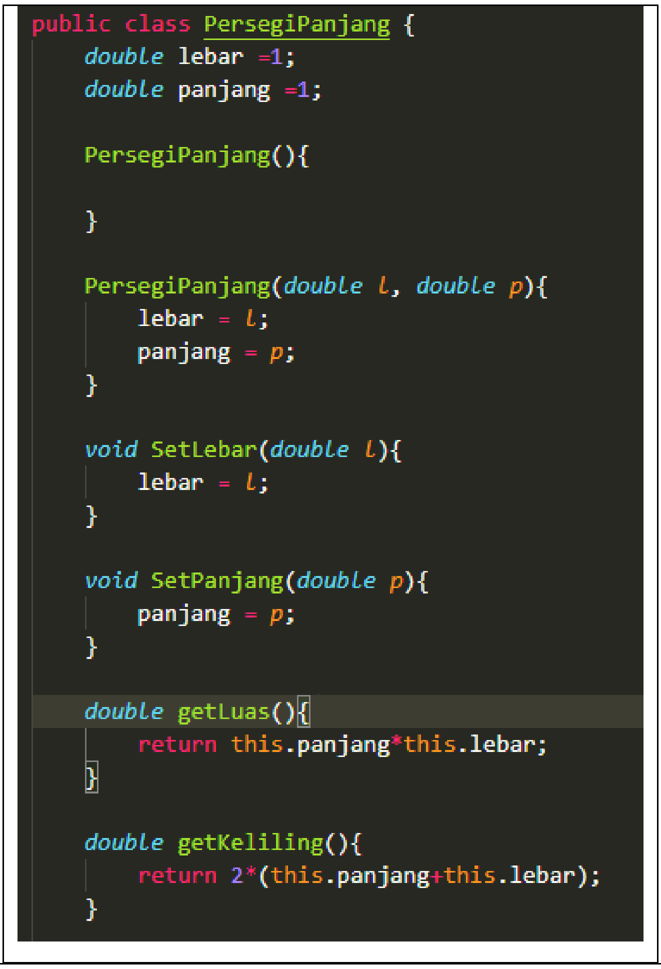
Implementasi dan Hasil

1. Class PersegiPanjang

   - UML Diagram

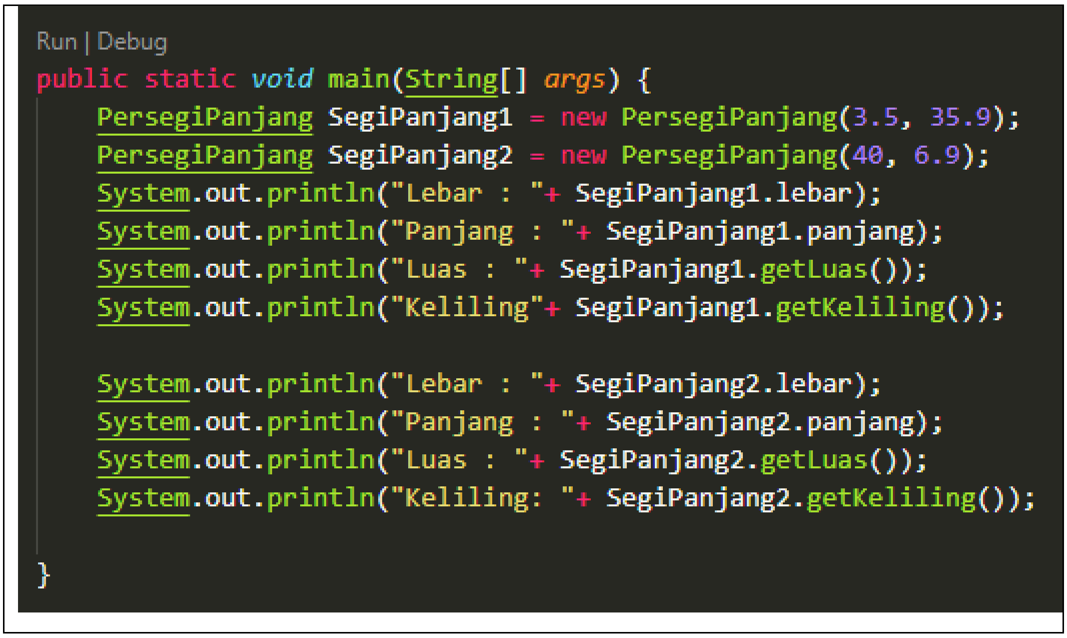
Persegi Panjang
Panjang : double Lebar : double
PersegiPanjang() PersegiPanjang(p: double, l: double) SetPanjang(P: double): void SetLebar(l : double): void getLuas(): double getKeliling(): double

   - Sourcecode (Class buatan)



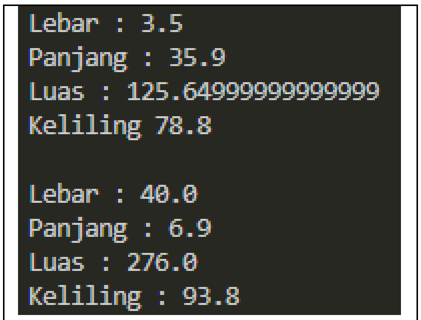
```java
public class PersegiPanjang {
    double lebar =1;
    double panjang =1;

    PersegiPanjang(){

    }

    PersegiPanjang(double l, double p){
        lebar = l;
        panjang = p;
    }

    void SetLebar(double l){
        lebar = l;
    }

    void SetPanjang(double p){
        panjang = p;
    }

    double getLuas(){
        return this.panjang*this.lebar;
    }

    double getKeliling(){
        return 2*(this.panjang+this.lebar);
    }
```

- Sourcecode (Class Utama)



```
public static void main(String[] args) {
    PersegiPanjang SegiPanjang1 = new PersegiPanjang(3.5, 35.9);
    PersegiPanjang SegiPanjang2 = new PersegiPanjang(40, 6.9);
    System.out.println("Lebar : "+ SegiPanjang1.lebar);
    System.out.println("Panjang : "+ SegiPanjang1.panjang);
    System.out.println("Luas : "+ SegiPanjang1.getLuas());
    System.out.println("Keliling"+ SegiPanjang1.getKeliling());

    System.out.println("Lebar : "+ SegiPanjang2.lebar);
    System.out.println("Panjang : "+ SegiPanjang2.panjang);
    System.out.println("Luas : "+ SegiPanjang2.getLuas());
    System.out.println("Keliling: "+ SegiPanjang2.getKeliling());

}
```

- Output



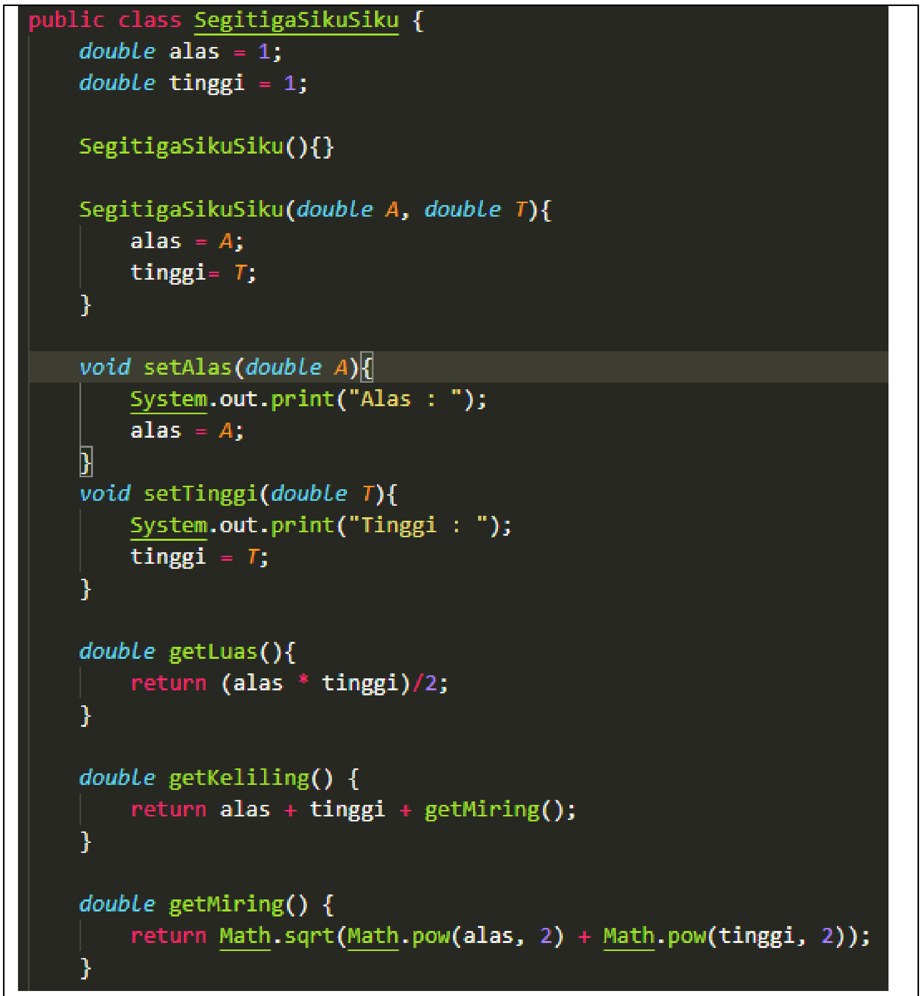
```
Lebar : 3.5
Panjang : 35.9
Luas : 125.64999999999999
Keliling 78.8

Lebar : 40.0
Panjang : 6.9
Luas : 276.0
Keliling : 93.8
```

2. Class SegitigaSikuSiku

- UML Diagram

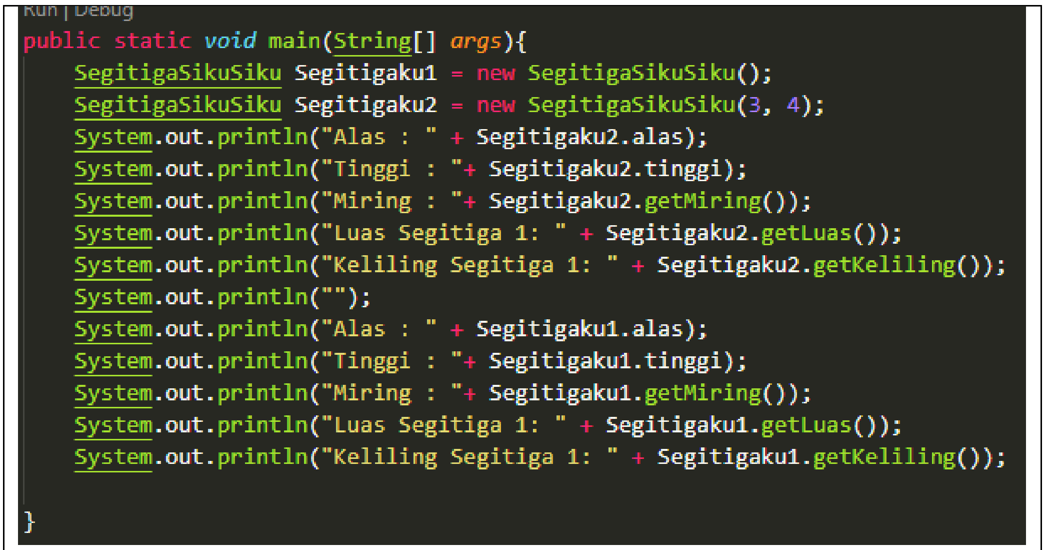
SegitigaSikuSiku
Alas : double Tinggi : double
SegitigaSikuSiku() SegitigaSikuSiku(A: double, T: double) SetAlas(A: double): void SetTinggi(T : double): void getLuas(): double getKeliling(): double getMiring(): double

- Sourcecode (Class Buatan)



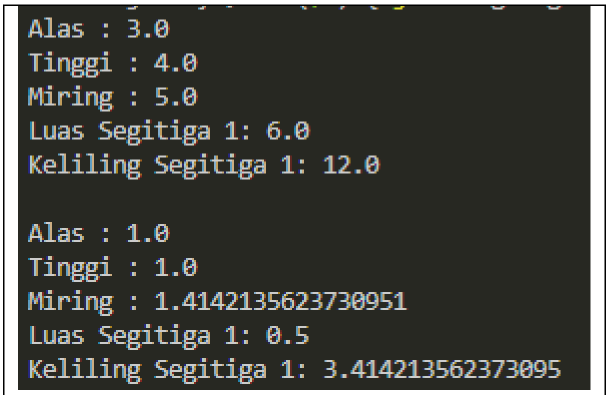
```java
public class SegitigaSikuSiku {
    double alas = 1;
    double tinggi = 1;

    SegitigaSikuSiku(){}

    SegitigaSikuSiku(double A, double T){
        alas = A;
        tinggi= T;
    }

    void setAlas(double A){
        System.out.print("Alas : ");
        alas = A;
    }
    void setTinggi(double T){
        System.out.print("Tinggi : ");
        tinggi = T;
    }

    double getLuas(){
        return (alas * tinggi)/2;
    }

    double getKeliling() {
        return alas + tinggi + getMiring();
    }

    double getMiring() {
        return Math.sqrt(Math.pow(alas, 2) + Math.pow(tinggi, 2));
    }
```

- Sourcecode (Class Utama)



```java
Run | Debug
public static void main(String[] args){
    SegitigaSikuSiku Segitigaku1 = new SegitigaSikuSiku();
    SegitigaSikuSiku Segitigaku2 = new SegitigaSikuSiku(3, 4);
    System.out.println("Alas : " + Segitigaku2.alas);
    System.out.println("Tinggi : "+ Segitigaku2.tinggi);
    System.out.println("Miring : "+ Segitigaku2.getMiring());
    System.out.println("Luas Segitiga 1: " + Segitigaku2.getLuas());
    System.out.println("Keliling Segitiga 1: " + Segitigaku2.getKeliling());
    System.out.println("");
    System.out.println("Alas : " + Segitigaku1.alas);
    System.out.println("Tinggi : "+ Segitigaku1.tinggi);
    System.out.println("Miring : "+ Segitigaku1.getMiring());
    System.out.println("Luas Segitiga 1: " + Segitigaku1.getLuas());
    System.out.println("Keliling Segitiga 1: " + Segitigaku1.getKeliling());

}
```

- Output



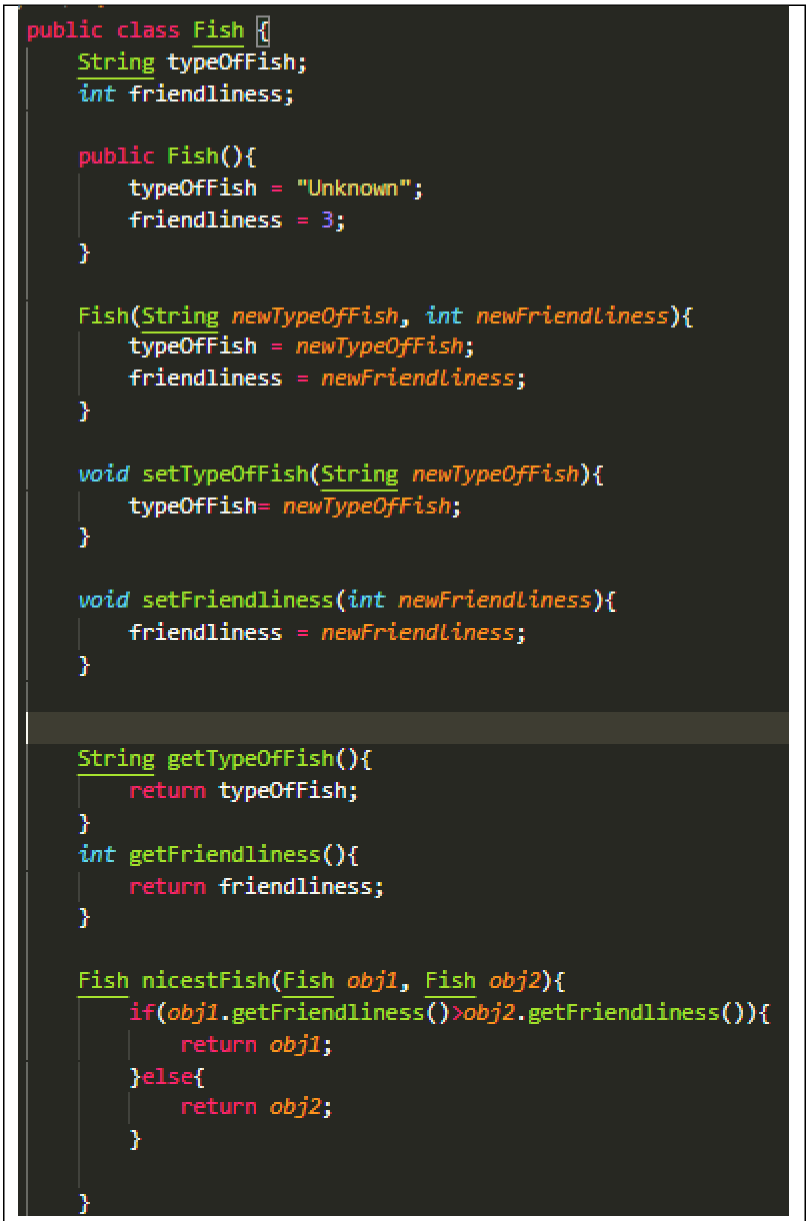
```
Alas : 3.0
Tinggi : 4.0
Miring : 5.0
Luas Segitiga 1: 6.0
Keliling Segitiga 1: 12.0

Alas : 1.0
Tinggi : 1.0
Miring : 1.4142135623730951
Luas Segitiga 1: 0.5
Keliling Segitiga 1: 3.414213562373095
```

3. Class Fish

- UML Diagram

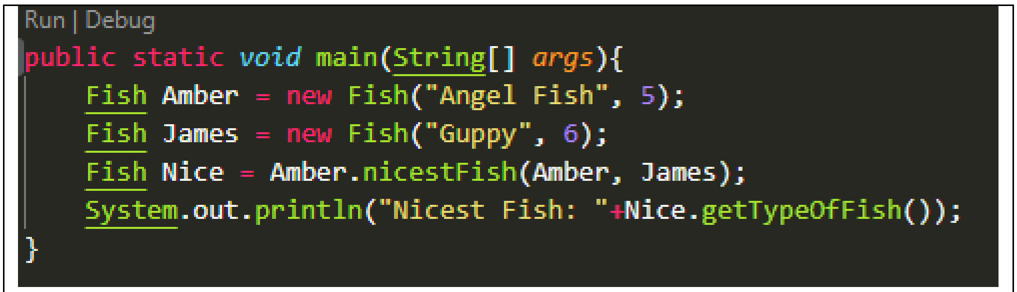
Fish
TypeOfFish : String friendliness : int
Fish() Fish(newTypeOfFish: String, newFriendliness: String) SetTypeOfFish (newTypeOfFish : String): void SetFriendliness (newFriendliness: int): void geTypeOfFish(): String getFriendliness(): int getMiring(): double nicestFish (obj1: Fish, obj2: Fish): Fish

- Sourcecode (Class Buatan)



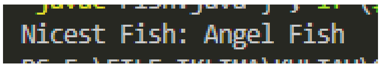
```
public class Fish {
    String typeOfFish;
    int friendliness;

    public Fish(){
        typeOfFish = "Unknown";
        friendliness = 3;
    }

    Fish(String newTypeOfFish, int newFriendliness){
        typeOfFish = newTypeOfFish;
        friendliness = newFriendliness;
    }

    void setTypeOfFish(String newTypeOfFish){
        typeOfFish= newTypeOfFish;
    }

    void setFriendliness(int newFriendliness){
        friendliness = newFriendliness;
    }


    String getTypeOfFish(){
        return typeOfFish;
    }
    int getFriendliness(){
        return friendliness;
    }

    Fish nicestFish(Fish obj1, Fish obj2){
        if(obj1.getFriendliness()>obj2.getFriendliness()){
            return obj1;
        }else{
            return obj2;
        }

    }
```

- Sourcecode (Class Utama)



```
public static void main(String[] args){
    Fish Amber = new Fish("Angel Fish", 5);
    Fish James = new Fish("Guppy", 6);
    Fish Nice = Amber.nicestFish(Amber, James);
    System.out.println("Nicest Fish: "+Nice.getTypeOfFish());
}
```

- Output



```
Nicest Fish: Angel Fish
```