

OBJECT ORIENTED PROGRAM PRACTICUM
JOB SHEET 9 : OVERLOADING DAN OVERRIDING



BY :

ATHRIYA GENFERIN

D4 INFORMATICS ENGINEERING (2I)

2241720075

(03)

State Polytechnic of Malang

Soekarno Hatta street No.9, Malang, East Java 65141

2022/2023



Praktikum 1

```
package jobsheet9;

public class Manager extends Karyawan {
    private double tunjangan;
    private String bagian;
    private Staff st[];

    public void setTunjangan (double tunjangan)
    {
        this.tunjangan=tunjangan;
    }
    public double getTunjangan ()
    {
        return tunjangan;
    }
    public void setBagian (String bagian)
    {
        this.bagian=bagian;
    }
    public String getBagian ()
    {
        return bagian;
    }
    public void setStaff (Staff st[])
    {
        this.st=st;
    }

    public void viewStaff()
    {
        int i;
        System.out.println("-----");
        for (i=0;i<st.length;i++)
        {
            st[i].lihatInfo();
        }
        System.out.println("-----");
    }
    public void lihatInfo()
    {
        System.out.println("Manager :"+this.getBagian());
        System.out.println("NIP :"+this.getNip());
        System.out.println("Nama :"+this.getNama());
    }
}
```



```
System.out.println("Golongan :"+this.getGolongan ());  
System.out.printf("Tunjangan :%.0f\n", this.getTunjangan ());  
System.out.printf("Gaji :.0f\n", this.getGaji());  
System.out.println("Bagian :"+this.getBagian ());  
  
this.viewStaff();  
}  
public double getGaji ()  
{  
return super.getGaji() +tunjangan;  
}  
}
```

```
package jobsheet9;  
  
public class Staff extends Karyawan {  
    private int lembur;  
    private double gajiLembur;  
  
    public void setLembur (int Lembur)  
    {  
        this.lembur=Lembur;  
    }  
    public int getLembur ()  
    {  
        return lembur;  
    }  
    public void setGajiLembur (double gajiLembur)  
    {  
        this.gajiLembur=gajiLembur;  
    }  
    public double getGajiLembur ()  
    {  
        return gajiLembur;  
    }  
    public double getGaji (int Lembur, double gajiLembur)  
    {  
        return super.getGaji () +Lembur*gajiLembur;  
    }  
    public double getGaji ()  
    {  
        return super.getGaji()+lembur*gajiLembur;  
    }  
    public void lihatInfo()  
    {  
        System.out.println("NIP :"+this.getNip());  
        System.out.println("Nama :"+this.getNama ());  
    }  
}
```



```
System.out.println("Golongan :"+this.getGolongan());  
System.out.println("Jml Lembur :"+this.getLembur());  
System.out.printf("Gaji Lembur :%.0f\n", this.getGajiLembur());  
System.out.printf("Gaji :%.0f\n", this.getGaji());  
}  
}
```

```
package jobsheet9;  
  
public class Karyawan {  
    private String nama;  
    private String nip;  
    private String golongan;  
    private double gaji;  
  
    public void setNama (String nama)  
    {  
        this.nama=nama;  
    }  
    public void setNip (String nip)  
    {  
        this.nip=nip;  
    }  
    public void setGolongan (String goLongan)  
    {  
        this.golongan=goLongan;  
  
        switch (goLongan.charAt(0)) {  
            case '1':this.gaji=5000000;  
                break;  
            case 2: this.gaji=3000000;  
                break;  
            case '3':this.gaji=2000000;  
                break;  
            case '4':this.gaji=1000000;  
                break;  
            case '5': this.gaji=750000;  
                break;  
        }  
    }  
  
    public void setGaji (double gaji)  
    {  
        this.gaji=gaji;  
    }  
}
```



```
public String getNama ()
{
    return nama;
}
public String getNip()
{
    return nip;
}
public String getGolongan ()
{
    return golongan;
}
public double getGaji ()
{
    return gaji;
}
}
```

```
package jobsheet9;

public class Utama {
public static void main(String[] args)
{
    System.out.println("Program Testing Class Manager & Staff");
    Manager man[]=new Manager [2];
    Staff staff1[]=new Staff [2];
    Staff staff2[]=new Staff [3];

    //pembuatan manager
    man[0]=new Manager();
    man[0].setNama ("Tedjo");
    man[0].setNip ("101");
    man[0].setGolongan ("1");
    man[0].setTunjangan (5000000);
    man[0].setBagian ("Administrasi");

    man[1]=new Manager();
    man[1].setNama ("Atika");
    man[1].setNip ("102");
    man[1].setGolongan ("1");
    man[1].setTunjangan (2500000);
    man[1].setBagian ("Pemasaran");

    staff1[0]=new Staff();
    staff1[0].setNama ("Usman");
    staff1[0].setNip ("0003");
}
```



```
staff1[0].setGolongan ("2");
staff1[0].setLembur (10);
staff1[0].setGajiLembur (10000);

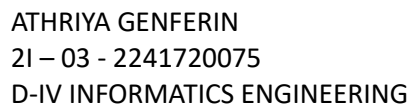
staff1[1]=new Staff();
staff1[1].setNama ("Anugrah");
staff1[1].setNip ("0005");
staff1[1].setGolongan ("2");
staff1[1].setLembur (10);
staff1[1].setGajiLembur (55000);
man[0].setStaff (staff1);

staff2[0]=new Staff();
staff2[0].setNama ("Hendra");
staff2[0].setNip ("0004");
staff2[0].setGolongan ("3");
staff2[0].setLembur (15);
staff2[0].setGajiLembur (5500);

staff2[1]=new Staff();
staff2[1].setNama ("Arie");
staff2[1].setNip ("0006");
staff2[1].setGolongan ("4");
staff2[1].setLembur (5);
staff2[1].setGajiLembur (100000);

staff2[2]=new Staff();
staff2[2].setNama ("Mentari");
staff2 [2].setNip ("0007");
staff2 [2].setGolongan ("3");
staff2 [2].setLembur (6);
staff2[2].setGajiLembur (20000);
man [1].setStaff (staff2);

//cetak informasi dari manager + staffnya
man [0].lihatInfo();
man [1].lihatInfo(); }
}
```



```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Nama :Anugrah
Golongan :2
Jml Lembur :10
Gaji Lembur :55000
Gaji :550000
-----
Manager :Pemasaran
NIP :102
Nama :Atika
Golongan :1
Tunjangan :250000
Gaji :.0f
Bagian :Pemasaran
-----
NIP :0004
Nama :Hendra
Golongan :3
Jml Lembur :15
Gaji Lembur :5500
Gaji :2082500
NIP :0006
Nama :Arie
Golongan :4
Jml Lembur :5
Gaji Lembur :100000
Gaji :1500000
NIP :0007
Nama :Mentari
Golongan :3
Jml Lembur :6
Gaji Lembur :20000
Gaji :2120000
-----
PS E:\SEMESTER 3\OOP\OOP>

```

```
package jobsheet9;

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);
    }
}
```

```
1075
43792
PS E:\SEMESTER 3\OOP\OOP>
```

questions

1. Dari source coding diatas terletak dimanakah overloading?

Overloading terjadi pada method perkalian dalam class Perkalianku. Dua metode dengan nama yang sama (perkalian) ada di dalam class tersebut.

2. Jika terdapat overloading ada berapa jumlah parameter yang berbeda?

Terdapat dua metode perkalian yang berbeda dalam class tersebut. Salah satunya memiliki dua parameter (int a dan int b), sementara yang lainnya memiliki tiga parameter (int a, int b, dan int c). Jadi, terdapat dua jumlah parameter yang berbeda pada metode perkalian.

PraKtiKum 2

```
package jobsheet9;

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();  
    }  
}
```

```
9.Fish'  
Ikan bisa berenang  
Piranha bisa makan daging  
PS E:\SEMESTER 3\OOP\OOP>
```

Questions

1. Dari source coding diatas terletak dimanakah overriding?

Overriding terjadi pada method swim di kelas Piranha. Kelas Piranha meng-override (mengganti) implementasi method swim yang ada di kelas induk Ikan

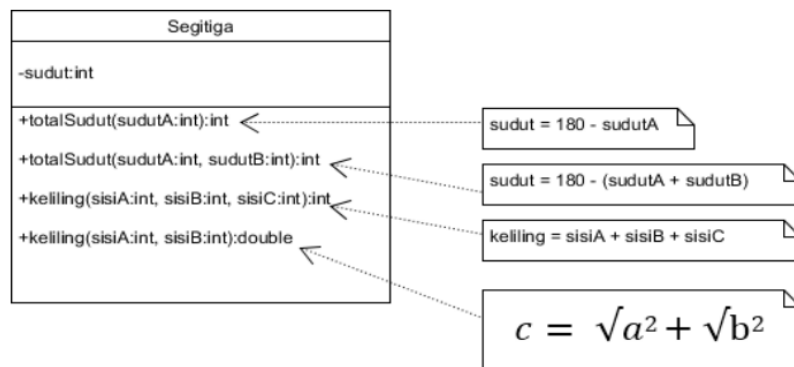
2. Jabarkanlah apabila sourcoding diatas jika terdapat overriding?

Pada source code di atas, overriding terjadi ketika kelas Piranha menimpa (override) implementasi dari method swim yang diwarisi dari kelas Ikan. Dengan demikian, ketika objek dari kelas Piranha dipanggil untuk menjalankan method swim, implementasi yang dipanggil adalah versi yang ada di kelas Piranha (bukan dari kelas Ikan).

tugas

5.1 Overloading

Implementasikan konsep overloading pada class diagram dibawah ini :





```
package jobsheet9;

public class Segitiga {
    private int sudut;

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

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

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

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

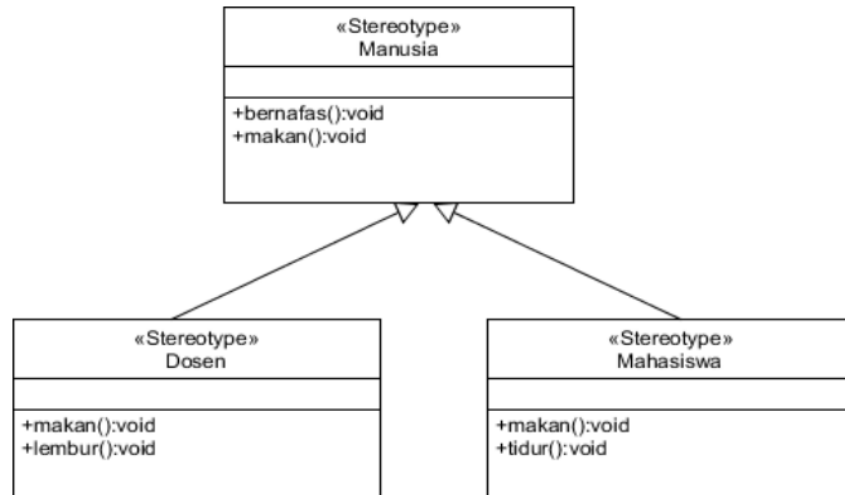
    public static void main(String[] args) {
        Segitiga segitiga = new Segitiga();
        int totalSudut1 = segitiga.totalSudut(60);
        System.out.println("Total Sudut 1: " + totalSudut1);
        int totalSudut2 = segitiga.totalSudut(60, 30);
        System.out.println("Total Sudut 2: " + totalSudut2);
        int kelilingABC = segitiga.keliling(3, 4, 5);
        System.out.println("Keliling ABC: " + kelilingABC);
        double kelilingAB = segitiga.keliling(3, 4);
        System.out.println("Keliling AB: " + kelilingAB);
    }
}
```

```
PS E:\SEMESTER 3\OOP\OOP> & 'C:\Program Files\Java\jdk1.8.0_121\bin\
\Users\WINDOWS 11\AppData\Roaming\Code\User\workspaceStorage\22ef0bb
36e70\redhat.java\jdt_ws\OOP_71cf1347\bin' 'jobsheet9.Segitiga'
Total Sudut 1: 120
Total Sudut 2: 90
Keliling ABC: 12
Keliling AB: 5.0
PS E:\SEMESTER 3\OOP\OOP>
```



5.2 Overriding

Implementasikan class diagram dibawah ini dengan menggunakan teknik dynamic method dispatch :



```
package jobsheet9;

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

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

class Dosen extends Manusia {
    public void makan() {
        System.out.println("Dosen sedang makan");
    }

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

class Mahasiswa extends Manusia {
    public void makan() {
        System.out.println("Mahasiswa sedang makan");
    }

    public void tidur() {
```



```
        System.out.println("Mahasiswa sedang tidur");  
    }  
}
```

```
package jobsheet9;  
  
public class Main {  
    public static void main(String[] args) {  
        Manusia manusia1 = new Dosen();  
        Manusia manusia2 = new Mahasiswa();  
        manusia1.bernafas();  
        manusia1.makan();  
        manusia2.bernafas();  
        manusia2.makan();  
    }  
}
```

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: Main + - [ ] [X]  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWin  
s  
  
PS E:\SEMESTER 3\OOP\OOP> & 'C:\Program Files\Java\jdk1.8.0_121\bin\java.exe' '-cp'  
  \Users\WINDOWS 11\AppData\Roaming\Code\User\workspaceStorage\22ef0bb066b2e04bb7720e1  
  36e70\redhat.java\jdt_ws\OOP_71cf1347\bin' 'jobsheet9.Main'  
Manusia sedang bernafas  
Dosen sedang makan  
Manusia sedang bernafas  
Mahasiswa sedang makan  
PS E:\SEMESTER 3\OOP\OOP>  
  
⊗ 0 △ 3 🔊 0 ➡ Ln 5, Col 25 Spaces: 4 UTF-8 CRLF {} Java 🔗 Go Live
```