

LAPORAN PRAKTIKUM

OOP

JOBSHEET 9



Disusun oleh:

Pascalis Dewangga S. L. 2241720140

TI - 2D

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023

1. Percobaan 1

a. Kode Program

Class Karyawan

```
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;
    }
}

```

Class Staff

```

public class Staff extends Karyawan {
    private int lembur;
    private double gajiLembur;

    public void setLembur(int lembur) {
        this.lembur = lembur;
    }

    public void setGajiLembur(double gajiLembur) {
        this.gajiLembur = gajiLembur;
    }

    public int getLembur() {
        return lembur;
    }

    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());
        System.out.println("-----");
    }
}

```

Class Manager

```

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

    public void setTunjangan(double tunjangan) {
        this.tunjangan = tunjangan;
    }

    public void setBagian(String bagian) {
        this.bagian = bagian;
    }

    public void setStaff(Staff st[]) {
        this.st = st;
    }

    public double getTunjangan() {
        return tunjangan;
    }

    public String getBagian() {
        return bagian;
    }

    public void viewStaff() {
        int i;
        for (i = 0; i < st.length; i++) {
            st[i].lihatInfo();
        }
    }

    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());
        System.out.println("-----");
    }
}

```

```

        this.viewStaff();
    }

    public double getGaji() {
        return super.getGaji() + tunjangan;
    }
}

```

Class Utama

```

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];

        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("Usman");
        staff1[1].setNip("0003");
        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);

        man[0].lihatInfo();
        man[1].lihatInfo();
    }
}

```

b. Hasil

```

Program Testing Class Manager & Staff
Manager :Administrasi
NIP      :101
Nama     :Tedjo
Golongan :1
Tunjangan :5000000
Gaji     :1000000
Bagian   :Administrasi
-----
NIP      :0003
Nama     :Usman
Golongan :2
Jml Lembur :10
Gaji Lembur :10000
Gaji     :3100000
-----
NIP      :0005
Nama     :Anugerah
Golongan :2
Jml Lembur :10
Gaji Lembur :55000
Gaji     :3550000
-----
Manager :Pemasaran
NIP      :102
Nama     :Atika
Golongan :1
Tunjangan :2500000
Gaji     :7500000
Bagian   :Pemasaran
-----
NIP      :0004
Nama     :Hendra
Golongan :3
Jml Lembur :15
Gaji Lembur :5500
Gaji     :2002500
-----
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
-----

```

2. Latihan

2.1 Dari source coding diatas terletak dimanakah overloading?

Jawab: Method bernama **perkalianku()** adalah contoh overloading

2.2 Jika terdapat overloading ada berapa jumlah parameter yang berbeda?

Jawab: ada satu parameter yang berbeda, dimana pada method **perkalianku()** yang kedua terdapat tambahan parameter dengan inisial **c**

2.3 Dari source coding diatas terletak dimanakah overloading?

Jawab: Method bernama **perkalian()**

2.4 Jika terdapat overloading ada berapa tipe parameter yang berbeda?

Jawab: ada satu parameter yang berbeda, dimana pada method **perkalian()** yang kedua parameternya bertipe data **double**.

2.5 Dari source coding diatas terletak dimanakah overriding?

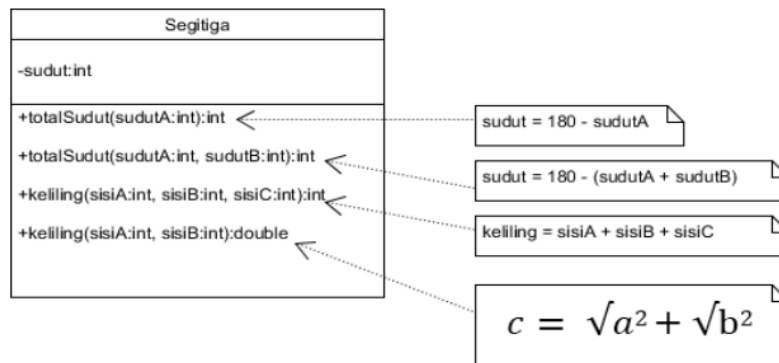
Jawab: Pada method **swim()** yang ada pada class **Ikan** dan **Piranha**

2.6 Jabarkanlah apabila sourcoding diatas jika terdapat overriding?

Jawab: dasarnya method yang ada pada class **Ikan** dan **Piranha** adalah method yang sama, hanya saja pada class **Piranha**, method tersebut memberikan output yang sedikit berbeda.

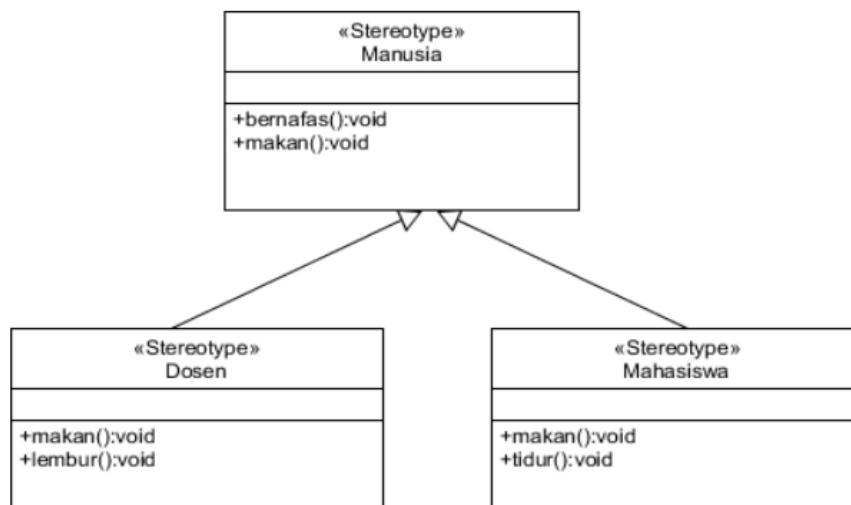
3. Tugas

3.1 Overloading



Terdapat dua macam method **totalSudut()** yang mana method tersebut merupakan contoh overloading. Keduanya dibedakan dengan parameter yang mana method **totalSudut()** yang kedua mendapat tambahan parameter berinisial **sudutB** dengan tipe data yang sama. Kemudian overloading juga dapat kita temui pada method **keliling()** dimana terdapat dua macam method yang mirip, dengan perbedaan tipe data dari method tersebut. Satu method bertipe data integer, dan yang lainnya menggunakan tipe data double.

3.2 Overriding



Teknik Dynamic Method Dispatch adalah mekanisme di mana pemanggilan metode yang akan dieksekusi ditentukan saat runtime. Dalam hal ini, maka ketika pemanggilan method **makan()** pada class **Dosen** dan **Mahasiswa**, akan seperti ini:

```
public class Main {
    public static void main(String[] args) {
        Manusia manusia1 = new Dosen();
        Manusia manusia2 = new Mahasiswa();

        manusia1.makan();
        manusia2.makan();
    }
}
```