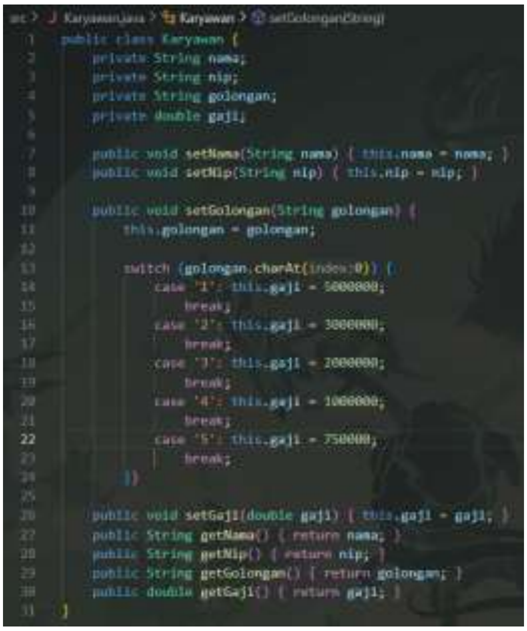
Zid’Avwa Al Bari’i

244107020083 / TI – 2I / 26

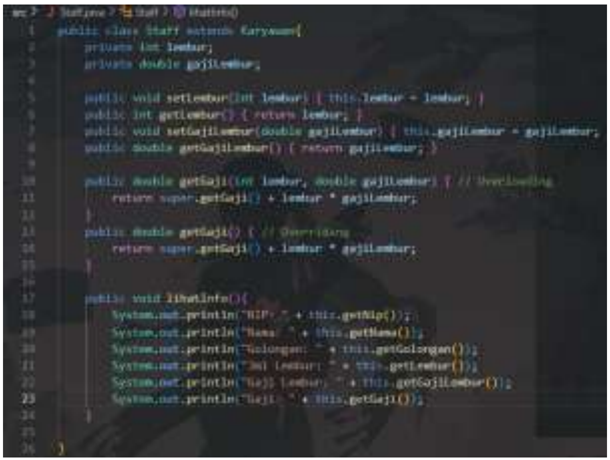
Jobsheet 7

Exercise 1

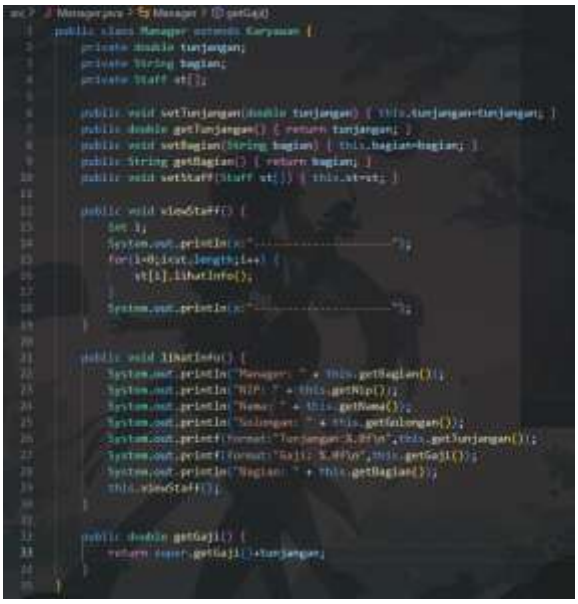
Codes


src > Karyawan.java > Karyawan > setGolongan(String)

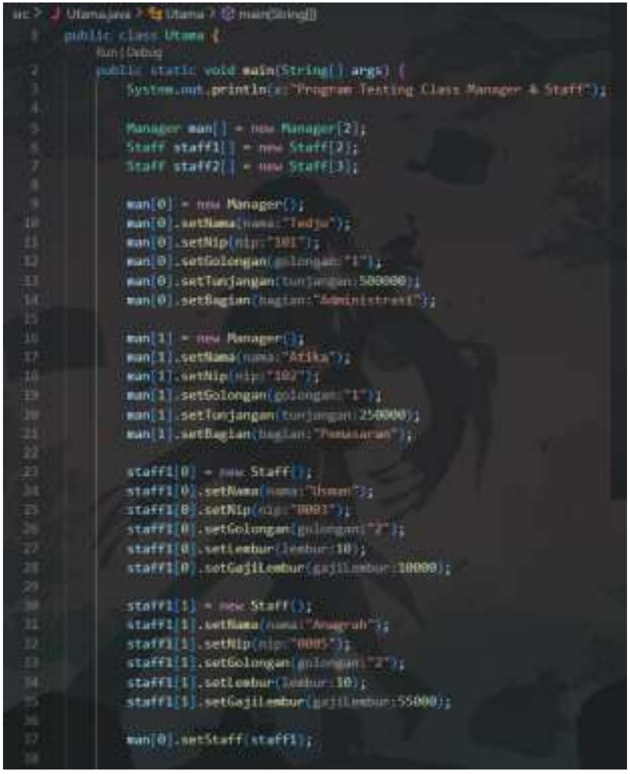
```java
public class Karyawan {
    private String nama;
    private String nip;
    private String golongan;
    private double gaji;

    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(index: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; }
}
```


src > Staff.java > Staff > lihatInfo()

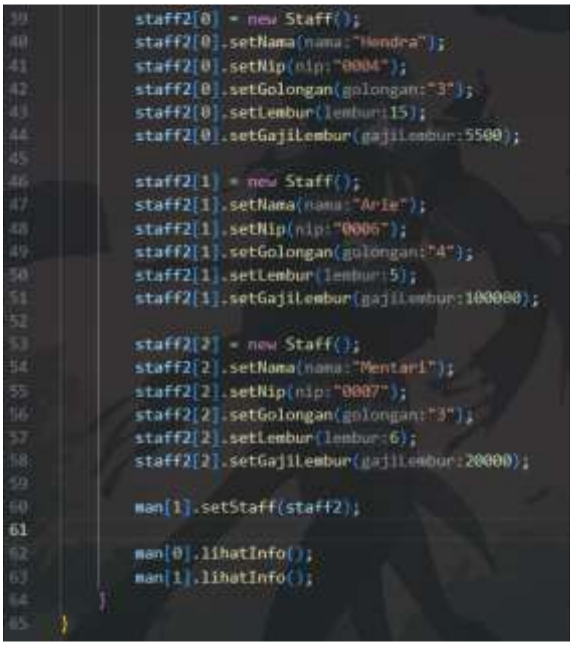
```java
public class Staff extends Karyawan{
    private int lembur;
    private double gajiLembur;

    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) { // Overloading
        return super.getGaji() + lembur * gajiLembur;
    }
    public double getGaji() { // Overriding
        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.println("Gaji Lembur: " + this.getGajiLembur());
        System.out.println("Gaji: " + this.getGaji());
    }

}
```


src > Manager.java > Manager > getGaji()

```java
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(x:"-------------------------------");
        for(i=0;i<st.length;i++) {
            st[i].lihatInfo();
        }
        System.out.println(x:"-------------------------------");
    }

    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(format:"Tunjangan:%.0f\n",this.getTunjangan());
        System.out.printf(format:"Gaji: %.0f\n",this.getGaji());
        System.out.println("Bagian: " + this.getBagian());
        this.viewStaff();
    }

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


src > Utama.java > Utama > main(String[])

```java
public class Utama {
    Run | Debug
    public static void main(String[] args) {
        System.out.println(x:"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(nama:"Tedjo");
        man[0].setNip(nip:"101");
        man[0].setGolongan(golongan:"1");
        man[0].setTunjangan(tunjangan:500000);
        man[0].setBagian(bagian:"Administrasi");

        man[1] = new Manager();
        man[1].setNama(nama:"Atika");
        man[1].setNip(nip:"102");
        man[1].setGolongan(golongan:"1");
        man[1].setTunjangan(tunjangan:250000);
        man[1].setBagian(bagian:"Pemasaran");

        staff1[0] = new Staff();
        staff1[0].setNama(nama:"Usman");
        staff1[0].setNip(nip:"0003");
        staff1[0].setGolongan(golongan:"2");
        staff1[0].setLembur(lembur:10);
        staff1[0].setGajiLembur(gajiLembur:10000);

        staff1[1] = new Staff();
        staff1[1].setNama(nama:"Anugrah");
        staff1[1].setNip(nip:"0005");
        staff1[1].setGolongan(golongan:"2");
        staff1[1].setLembur(lembur:10);
        staff1[1].setGajiLembur(gajiLembur:55000);

        man[0].setStaff(staff1);

```



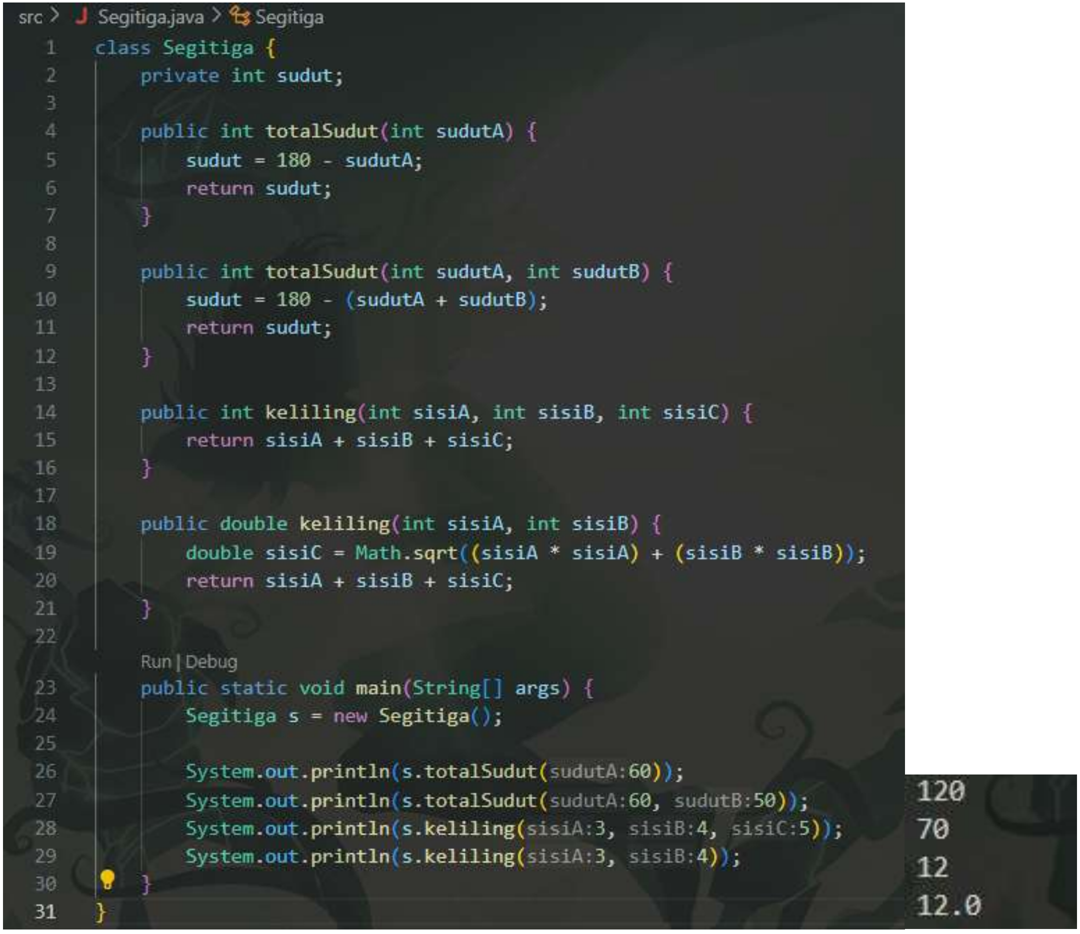
```
        staff2[0] = new Staff();
        staff2[0].setNama(nama:"Hendra");
        staff2[0].setNip(nip:"0004");
        staff2[0].setGolongan(golongan:"3");
        staff2[0].setLembur(lembur:15);
        staff2[0].setGajiLembur(gajiLembur:5500);

        staff2[1] = new Staff();
        staff2[1].setNama(nama:"Arie");
        staff2[1].setNip(nip:"0006");
        staff2[1].setGolongan(golongan:"4");
        staff2[1].setLembur(lembur:5);
        staff2[1].setGajiLembur(gajiLembur:100000);

        staff2[2] = new Staff();
        staff2[2].setNama(nama:"Mentari");
        staff2[2].setNip(nip:"0007");
        staff2[2].setGolongan(golongan:"3");
        staff2[2].setLembur(lembur:6);
        staff2[2].setGajiLembur(gajiLembur:20000);

        man[1].setStaff(staff2);

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

Questions & Answers:

1. Where is the overloading?
   The overloading occurs in the two perkalian() methods, where one has two parameters (int a, int b) and the other has three parameters (int a, int b, int c).
2. How many different parameters are there?
   There are 2 different parameter variations:
   - perkalian(int, int)
   - perkalian(int, int, int)
3. Where is the overloading?
   The overloading occurs in the perkalian() methods where one uses integer parameters (int a, int b) and the other uses double parameters (double a, double b), but with the same method name.
4. How many different types of parameters are there?
   There are 2 different data types:
   - int, int
   - double, double
5. Where is the overriding?
   The overriding happens in the Piranha class, where the method: public void swim() overrides the same swim() method from the Ikan class.
6. Describe the source coding if there is overriding.
   Overriding means the subclass (Piranha) provides a new implementation of a method that already exists in its superclass (Ikan) using the same method name and parameters. When the method is called through an object of the subclass, the subclass version is executed (polymorphism).

TASKS

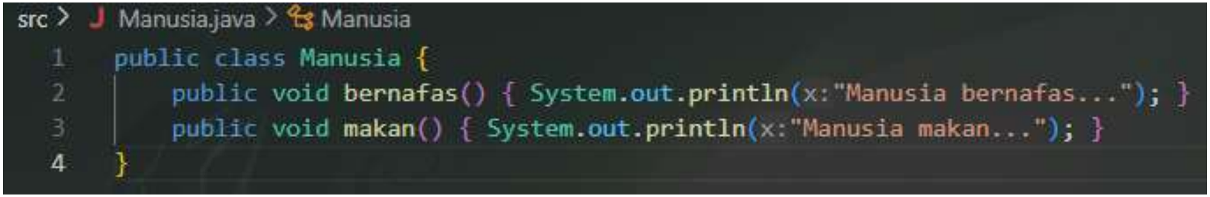
1. Overloading



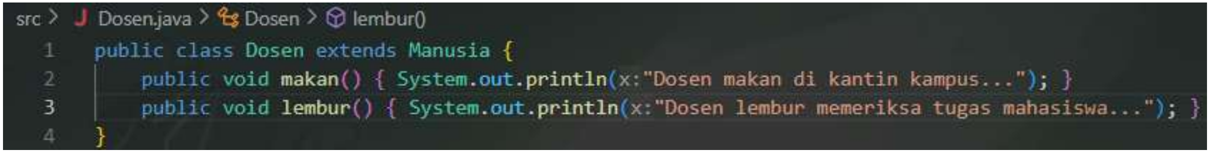
```java
class Segitiga {
    private int sudut;

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

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

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

    public double keliling(int sisiA, int sisiB) {
        double sisiC = Math.sqrt((sisiA * sisiA) + (sisiB * sisiB));
        return sisiA + sisiB + sisiC;
    }

    public static void main(String[] args) {
        Segitiga s = new Segitiga();

        System.out.println(s.totalSudut(sudutA:60));
        System.out.println(s.totalSudut(sudutA:60, sudutB:50));
        System.out.println(s.keliling(sisiA:3, sisiB:4, sisiC:5));
        System.out.println(s.keliling(sisiA:3, sisiB:4));
    }
}
```

```
120
70
12
12.0
```

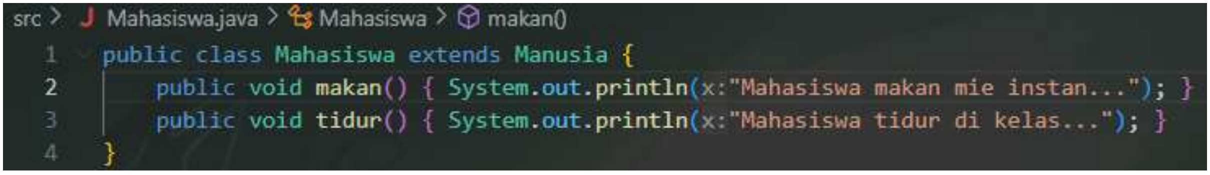
2. Overriding



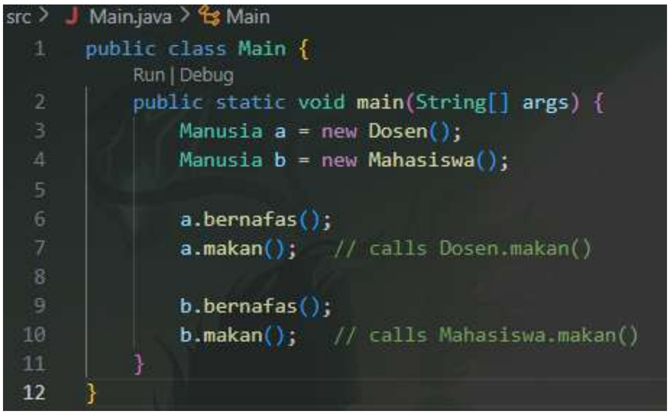
```java
public class Manusia {
    public void bernafas() { System.out.println(x:"Manusia bernafas..."); }
    public void makan() { System.out.println(x:"Manusia makan..."); }
}
```



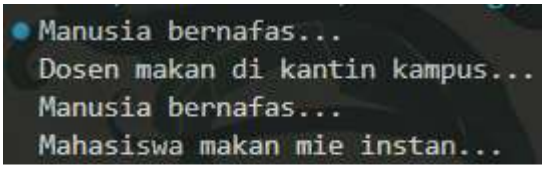
```java
public class Dosen extends Manusia {
    public void makan() { System.out.println(x:"Dosen makan di kantin kampus..."); }
    public void lembur() { System.out.println(x:"Dosen lembur memeriksa tugas mahasiswa..."); }
}
```



```java
public class Mahasiswa extends Manusia {
    public void makan() { System.out.println(x:"Mahasiswa makan mie instan..."); }
    public void tidur() { System.out.println(x:"Mahasiswa tidur di kelas..."); }
}
```



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

        a.bernafas();
        a.makan();    // calls Dosen.makan()

        b.bernafas();
        b.makan();    // calls Mahasiswa.makan()
    }
}
```



```
Manusia bernafas...
Dosen makan di kantin kampus...
Manusia bernafas...
Mahasiswa makan mie instan...
```