JOBSHEET 6

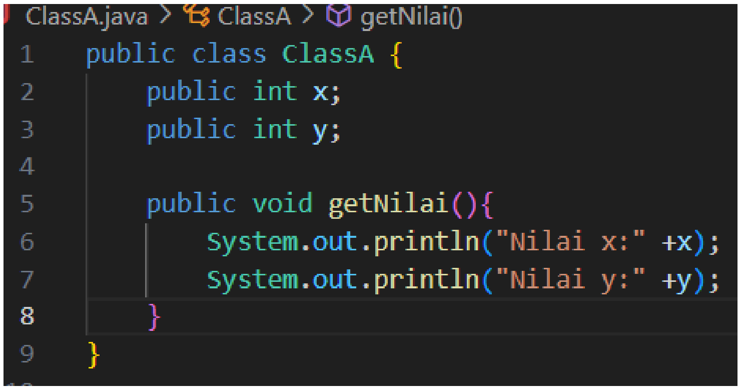
INHERITANCE

Erwan Majid/08/2i

Link Github: https://github.com/Majid5654/Semester-3/tree/Main/JAVA%20OOP/Week6

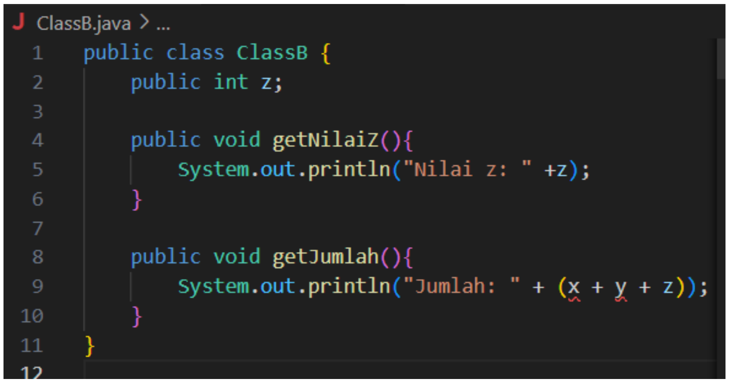
- **TRIAL 1 (extends)**

ClassA:



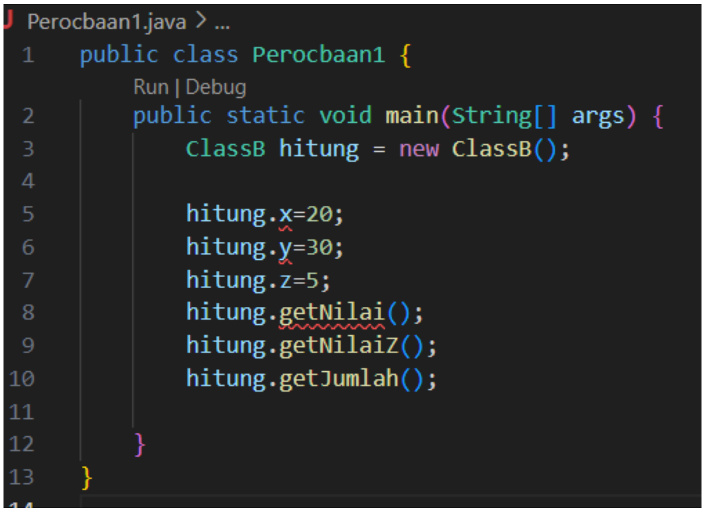
```java
public class ClassA {
    public int x;
    public int y;

    public void getNilai(){
        System.out.println("Nilai x:" +x);
        System.out.println("Nilai y:" +y);
    }
}
```

ClassB:



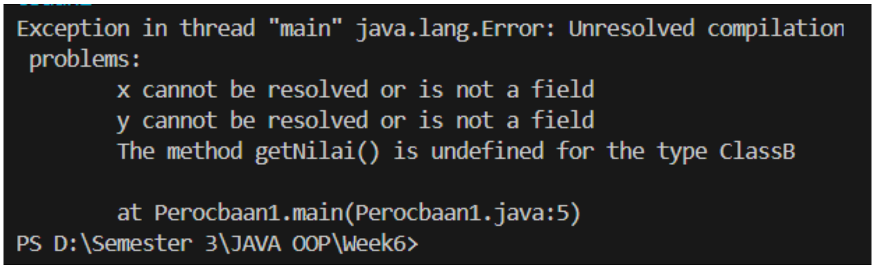
```java
public class ClassB {
    public int z;

    public void getNilaiZ(){
        System.out.println("Nilai z: " +z);
    }

    public void getJumlah(){
        System.out.println("Jumlah: " + (x + y + z));
    }
}
```

Percobaan1:



```java
public class Perocbaan1 {
    public static void main(String[] args) {
        ClassB hitung = new ClassB();

        hitung.x=20;
        hitung.y=30;
        hitung.z=5;
        hitung.getNilai();
        hitung.getNilaiZ();
        hitung.getJumlah();

    }
}
```

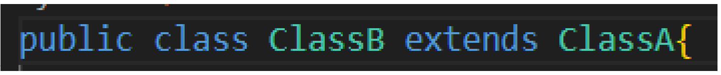
Output:



```
Exception in thread "main" java.lang.Error: Unresolved compilation
 problems:
        x cannot be resolved or is not a field
        y cannot be resolved or is not a field
        The method getNilai() is undefined for the type ClassB

        at Perocbaan1.main(Perocbaan1.java:5)
PS D:\Semester 3\JAVA OOP\Week6>
```

B. QUESTIONS

1. In Experiment 1 above the program that was running error occurred, then fix so that the program can be run and not error!



```java
public class ClassB extends ClassA{
```

-

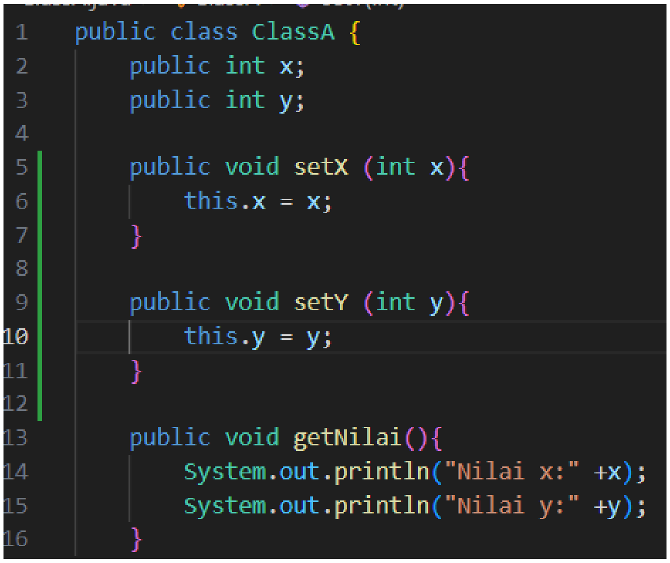
add code extends Class A.

2. Explain what caused the program in experiment 1 when it ran an error!

-It because ClassB access the variables x and y from ClassA, but there is no relationship between the two classes. To fix this, it need to make ClassB extend ClassA, so that ClassB can inherit the fields x and y from ClassA

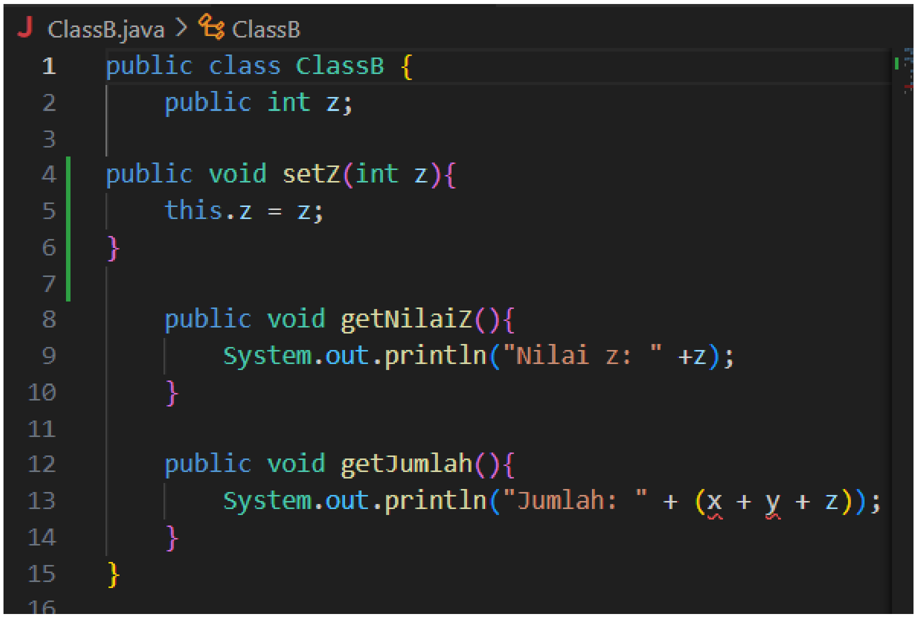
- **TRIAL 2 (Access Control)**

ClassA:



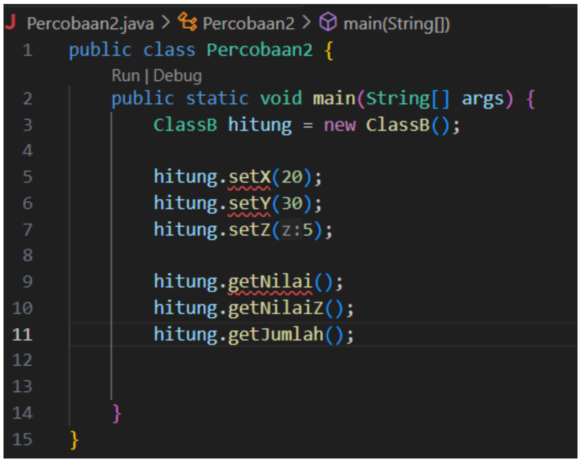
```java
public class ClassA {
    public int x;
    public int y;

    public void setX (int x){
        this.x = x;
    }

    public void setY (int y){
        this.y = y;
    }

    public void getNilai(){
        System.out.println("Nilai x:" +x);
        System.out.println("Nilai y:" +y);
    }
```

ClassB:



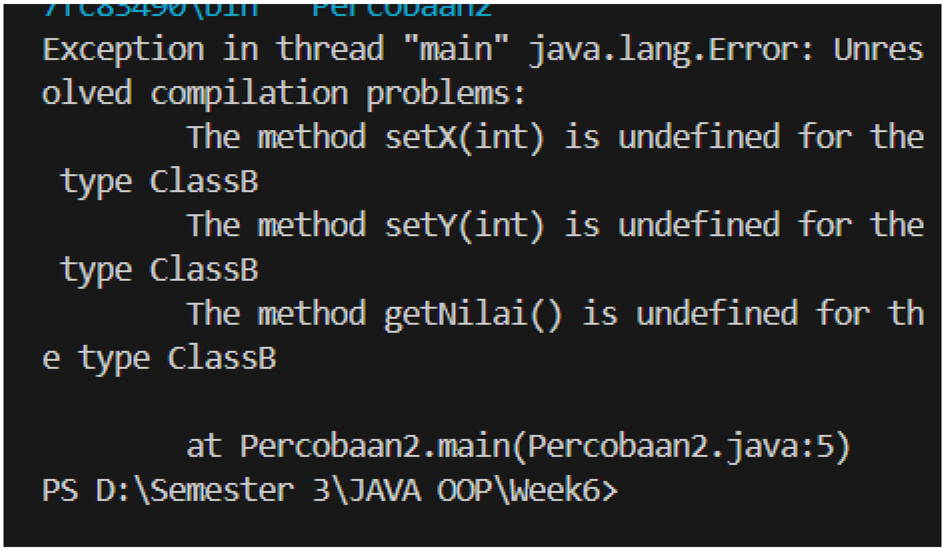
```java
public class ClassB {
    public int z;

public void setZ(int z){
    this.z = z;
}

    public void getNilaiZ(){
        System.out.println("Nilai z: " +z);
    }

    public void getJumlah(){
        System.out.println("Jumlah: " + (x + y + z));
    }
}
```

Main:



```java
public class Percobaan2 {
    public static void main(String[] args) {
        ClassB hitung = new ClassB();

        hitung.setX(20);
        hitung.setY(30);
        hitung.setZ(z:5);

        hitung.getNilai();
        hitung.getNilaiZ();
        hitung.getJumlah();


    }
}
```

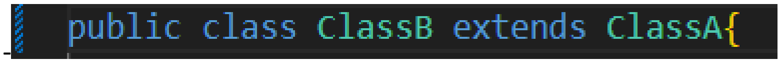
Output:



```
7fc83490\bin' 'Percobaan2
Exception in thread "main" java.lang.Error: Unres
olved compilation problems:
        The method setX(int) is undefined for the
 type ClassB
        The method setY(int) is undefined for the
 type ClassB
        The method getNilai() is undefined for th
e type ClassB

        at Percobaan2.main(Percobaan2.java:5)
PS D:\Semester 3\JAVA OOP\Week6>
```

B. QUESTIONS

1. In Experiment 2 above, the program that runs an error occurs, then fix it so that the program can be run and not error!



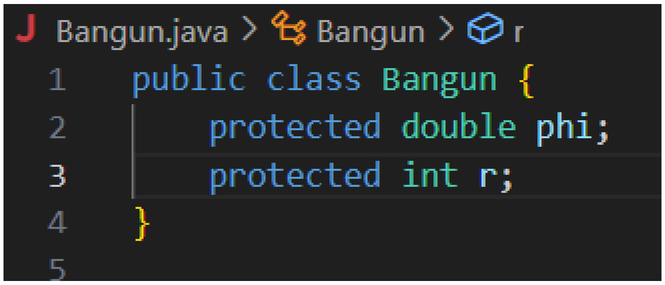
```java
public class ClassB extends ClassA{
```

2. Explain what caused the program in experiment 1 when it ran an error!

-It because ClassB access the variables x and y from ClassA, but there is no relationship between the two classes. To fix this, it need to make ClassB extend ClassA, so that ClassB can inherit the fields x and y from ClassA

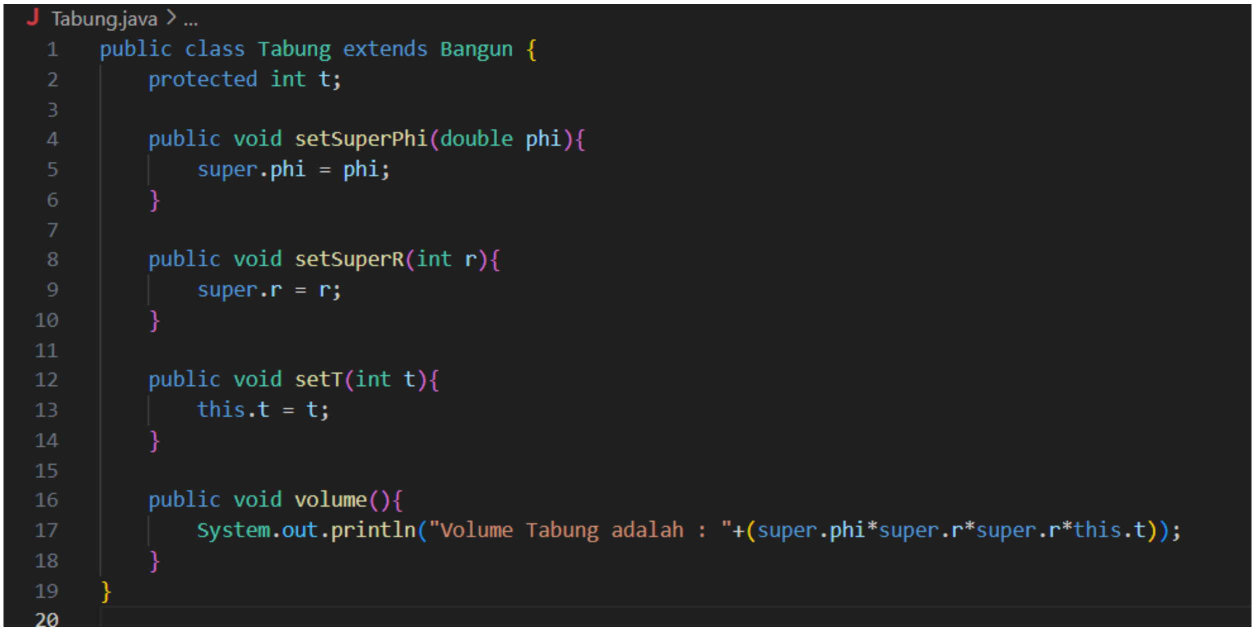
- **TRIAL 3 (Super)**

Bangun:



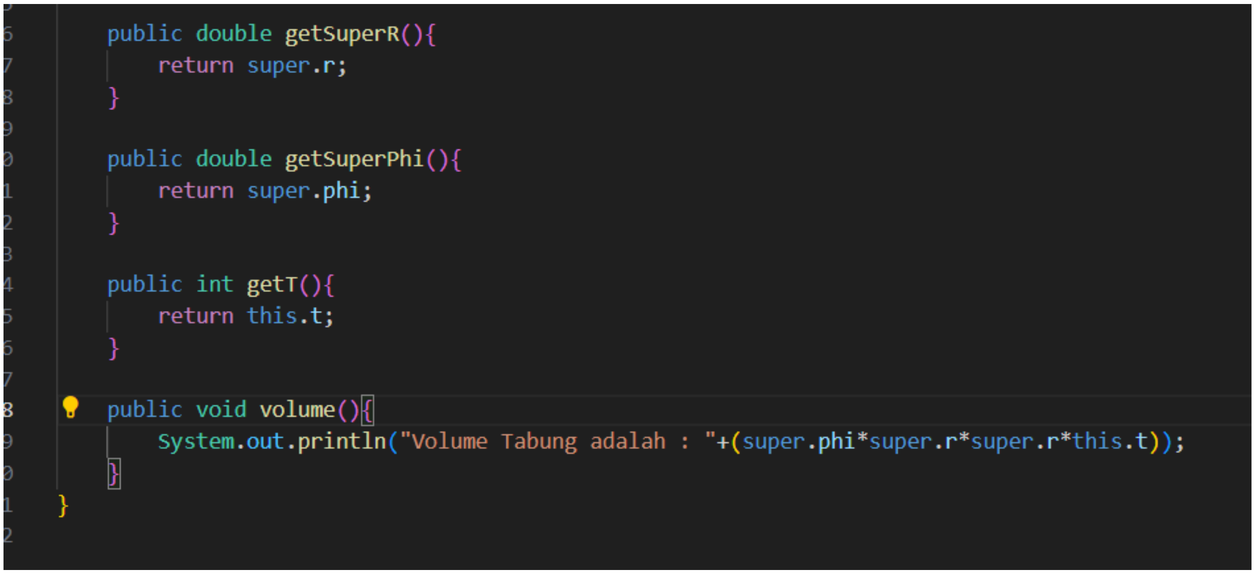
```java
public class Bangun {
    protected double phi;
    protected int r;
}
```

Tabung:



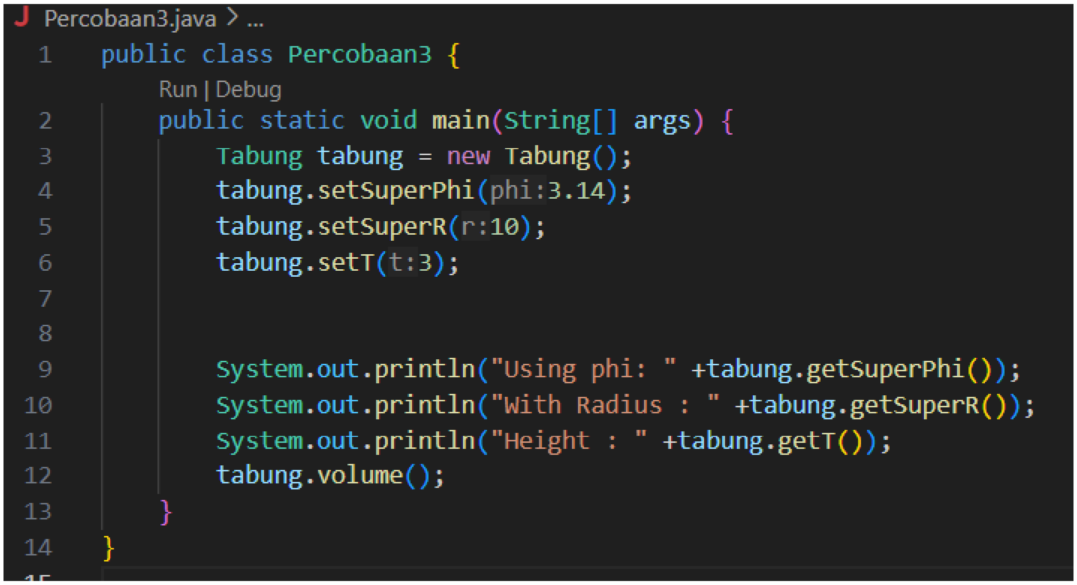
Tabung.java > ...

```java
public class Tabung extends Bangun {
    protected int t;

    public void setSuperPhi(double phi){
        super.phi = phi;
    }

    public void setSuperR(int r){
        super.r = r;
    }

    public void setT(int t){
        this.t = t;
    }

    public void volume(){
        System.out.println("Volume Tabung adalah : "+(super.phi*super.r*super.r*this.t));
    }
}
```



```java
    public double getSuperR(){
        return super.r;
    }

    public double getSuperPhi(){
        return super.phi;
    }

    public int getT(){
        return this.t;
    }

    public void volume(){
        System.out.println("Volume Tabung adalah : "+(super.phi*super.r*super.r*this.t));
    }
}
```

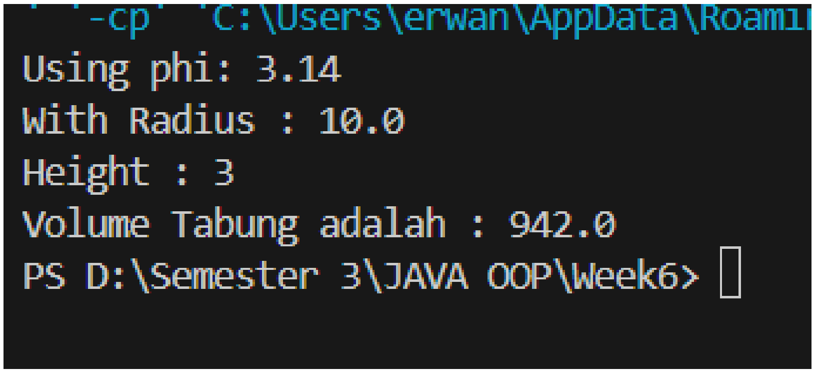
Main:



Percobaan3.java > ...

```java
public class Percobaan3 {
    Run | Debug
    public static void main(String[] args) {
        Tabung tabung = new Tabung();
        tabung.setSuperPhi(phi:3.14);
        tabung.setSuperR(r:10);
        tabung.setT(t:3);


        System.out.println("Using phi: " +tabung.getSuperPhi());
        System.out.println("With Radius : " +tabung.getSuperR());
        System.out.println("Height : " +tabung.getT());
        tabung.volume();
    }
}
```

Output:



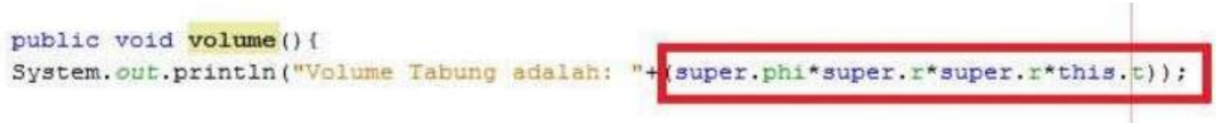
```
' '-cp' 'C:\Users\erwan\AppData\Roami
Using phi: 3.14
With Radius : 10.0
Height : 3
Volume Tabung adalah : 942.0
PS D:\Semester 3\JAVA OOP\Week6> 
```

QUESTIONS

1. Explain the "super" function in the following program snippet in the Tube class!

-super refers to the superclass (parent class) of the current class. Tabung is a subclass that extends Bangun. The fields phi and r defined in the Bangun class. When it call super.phi and super.r belong to the Bangun class and are being inherited into the Tabung class

2. Explain the "super" and "this" functions in the following program snippet in the Tube class



```
public void volume(){
System.out.println("Volume Tabung adalah: "+(super.phi*super.r*super.r*this.t));
```

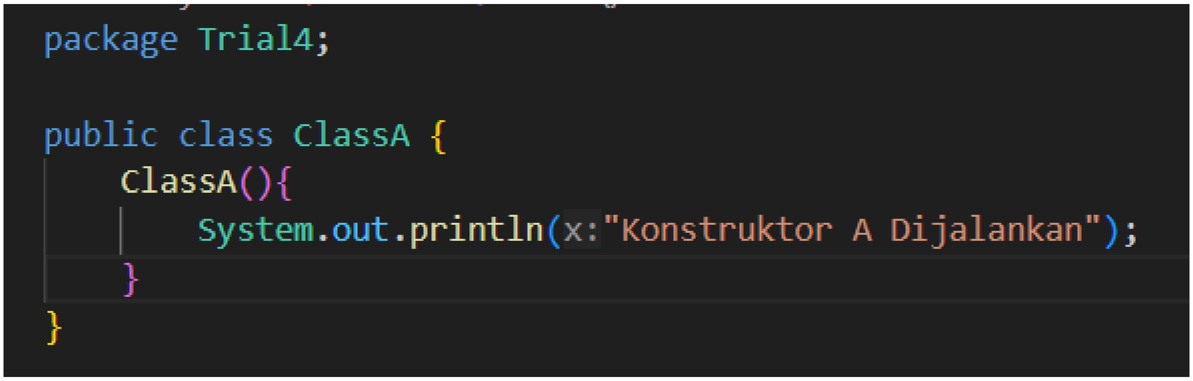
-if ‘super’ its reference to the parent class and are being inherited into tabung class,then ‘this’,is Initializes fields that belong to the current class

3. Explain why the Tube class does not declare the "phi" and "r" attributes, but the class can access these attributes!

-because these attributes are inherited from its superclass (Bangun) and in the tube class extends Bangun. Inheritance is designed to promote code reuse

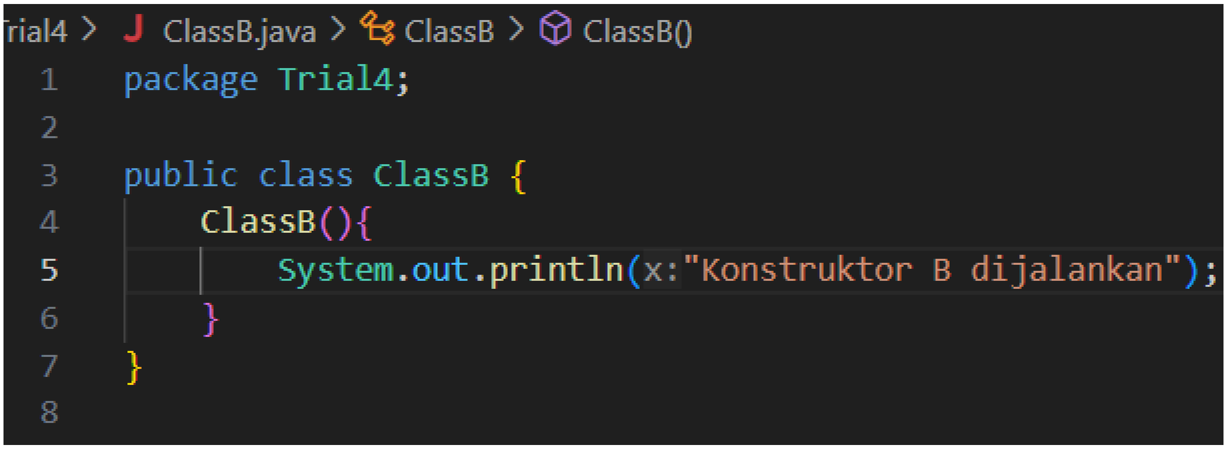
- **TRIAL 4 (super contructor)**

ClassA:



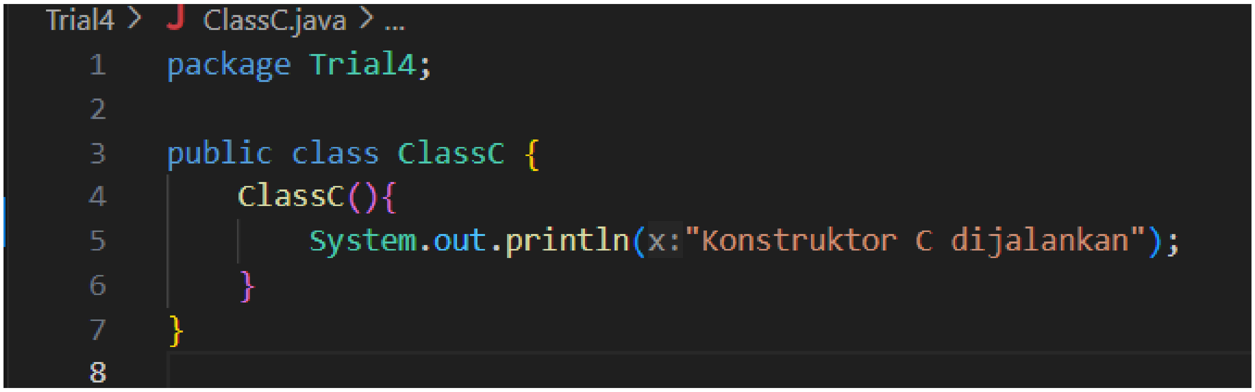
```
package Trial4;

public class ClassA {
    ClassA(){
        System.out.println(x:"Konstruktor A Dijalankan");
    }
}
```

ClassB:



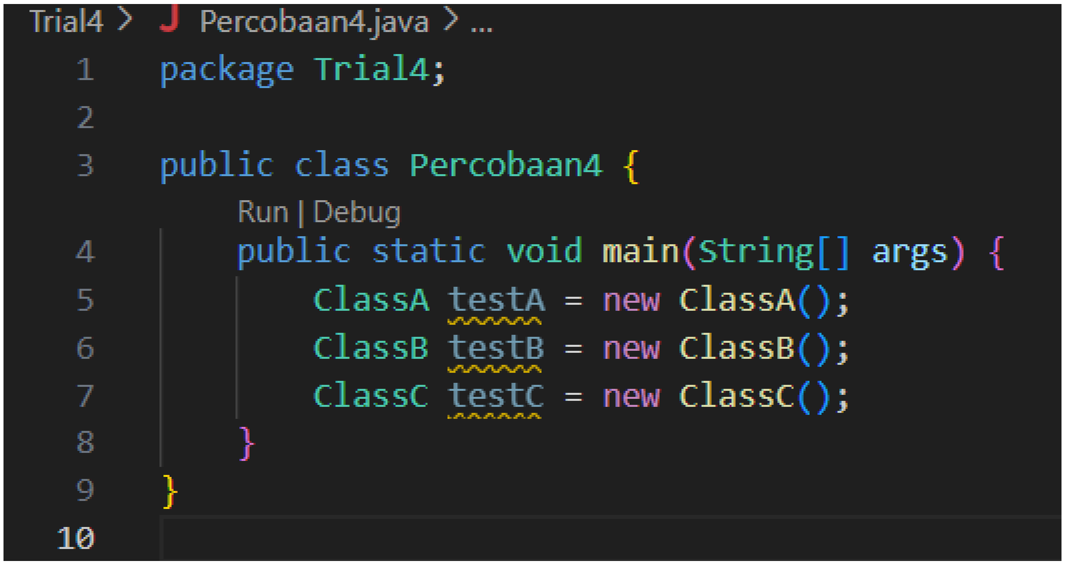
```
package Trial4;

public class ClassB {
    ClassB(){
        System.out.println(x:"Konstruktor B dijalankan");
    }
}

```

ClassC:



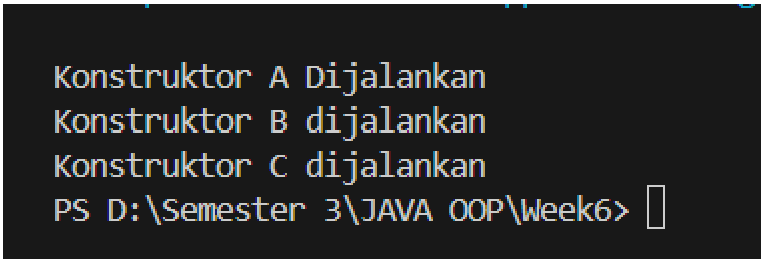
```java
package Trial4;

public class ClassC {
    ClassC(){
        System.out.println(x:"Konstruktor C dijalankan");
    }
}

```

Main:



```java
package Trial4;

public class Percobaan4 {
    public static void main(String[] args) {
        ClassA testA = new ClassA();
        ClassB testB = new ClassB();
        ClassC testC = new ClassC();
    }
}

```

Output:



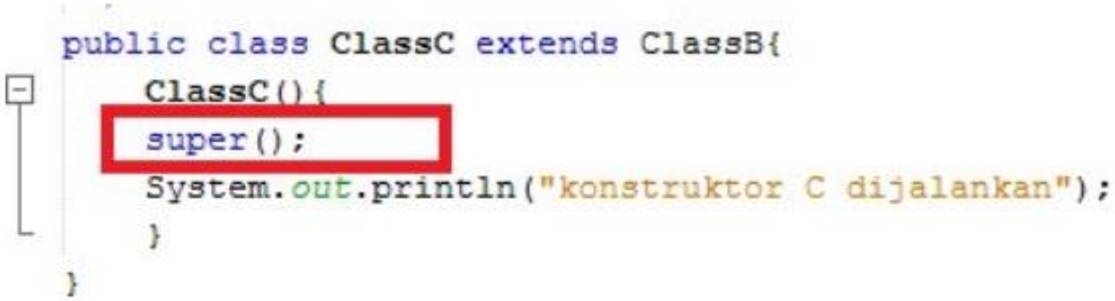
```
Konstruktor A Dijalankan
Konstruktor B dijalankan
Konstruktor C dijalankan
PS D:\Semester 3\JAVA OOP\Week6>
```

QUESTIONS

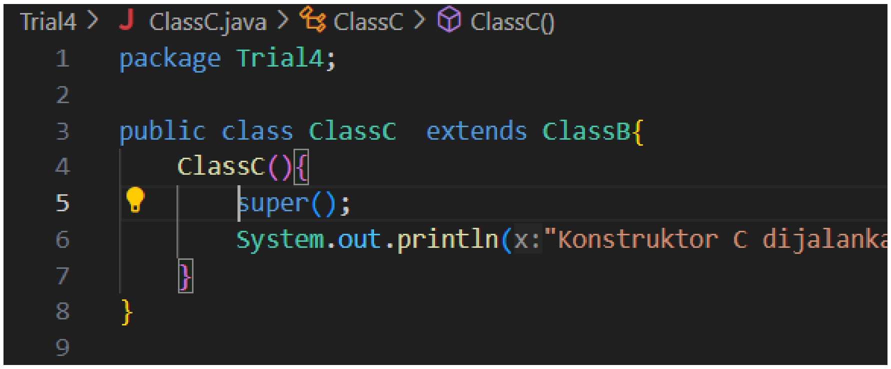
1. In experiment 4 state which class includes the superclass and subclass, then explain the reason!

-none of the classes explicitly include the concepts of superclass or subclass. All classes are independent and do not have an inheritance relationship with each other

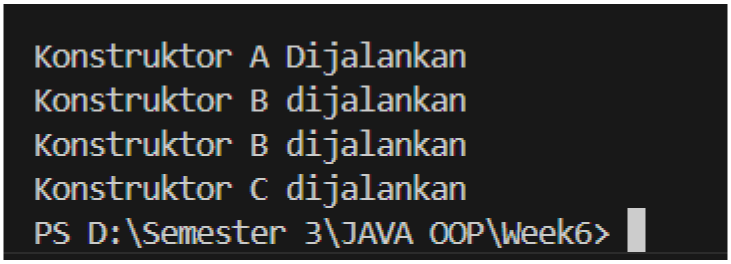
2. Change the contents of the ClassC default constructor as follows:



```
public class ClassC extends ClassB{
    ClassC(){
    super();
    System.out.println("konstruktor C dijalankan");
    }
}
```



```
package Trial4;

public class ClassC  extends ClassB{
    ClassC(){
        super();
        System.out.println(x:"Konstruktor C dijalanka
    }
}

```



```
Konstruktor A Dijalankan
Konstruktor B dijalankan
Konstruktor B dijalankan
Konstruktor C dijalankan
PS D:\Semester 3\JAVA OOP\Week6>
```

Add the word super () in the First row in the default constructor. Try running the Experiment 4 class

again and it looks like there is no difference from the output!

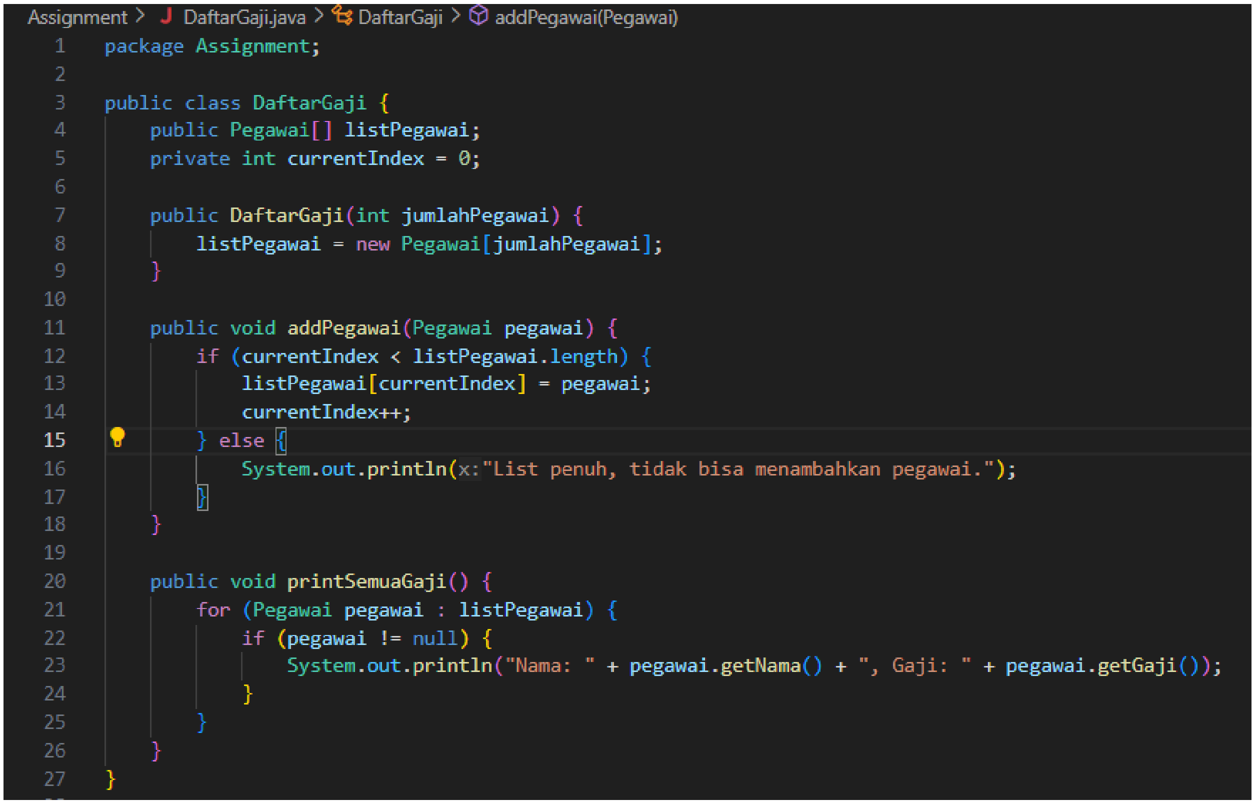
Explain how the order of the constructor goes when the test object is created!
-The constructor call order ensures that the superclass (in this case, ClassB) is initialized before the subclass (ClassC). This order is important to ensure that any inherited properties or behaviors in the subclass are properly set up by the superclass before the subclass-specific code is executed.

4. What is the super () function in the following program snippet in ClassC

-The super() function in this program snippet from ClassC explicitly invokes the constructor of the superclass (ClassB). It ensures that ClassB is initialized properly before the ClassC constructor continues its execution, maintaining the proper order of initialization in the inheritance hierarchy.

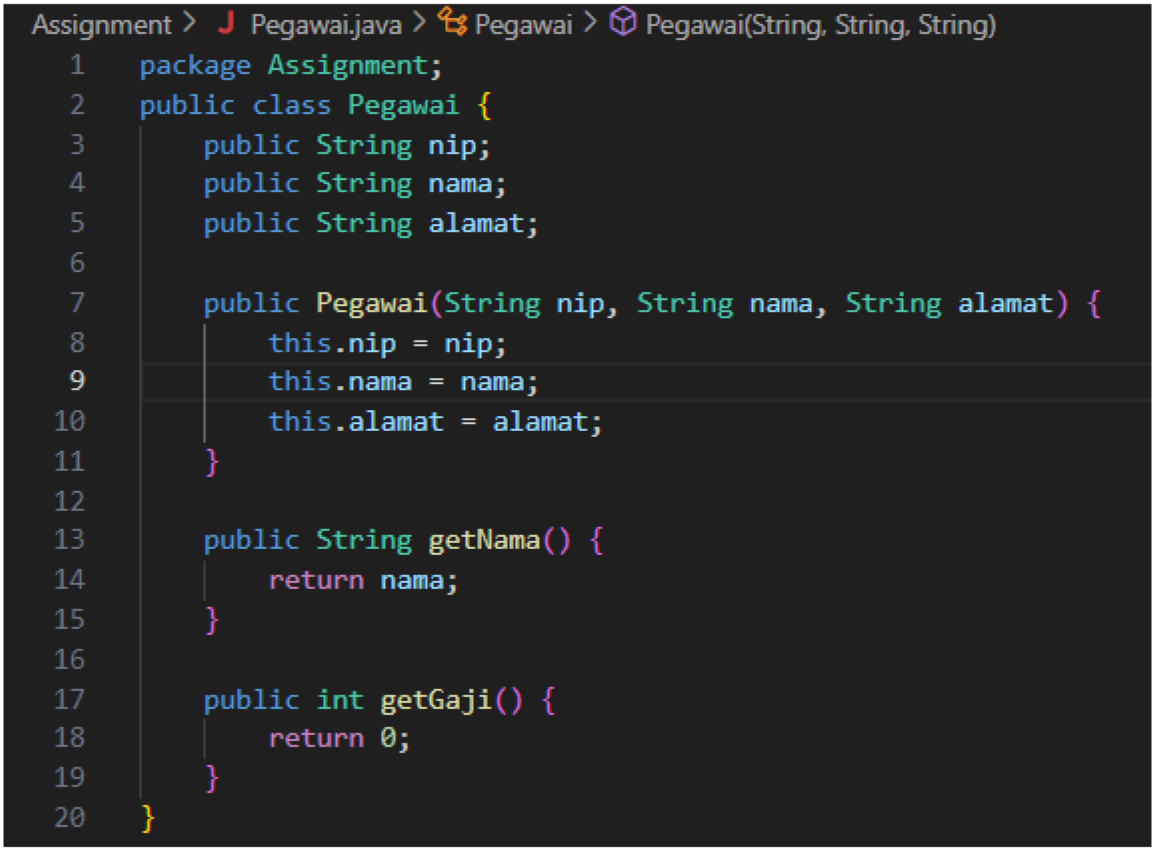
ASSIGNMENTS

DaftarGaji:


Assignment > DaftarGaji.java > DaftarGaji > addPegawai(Pegawai)

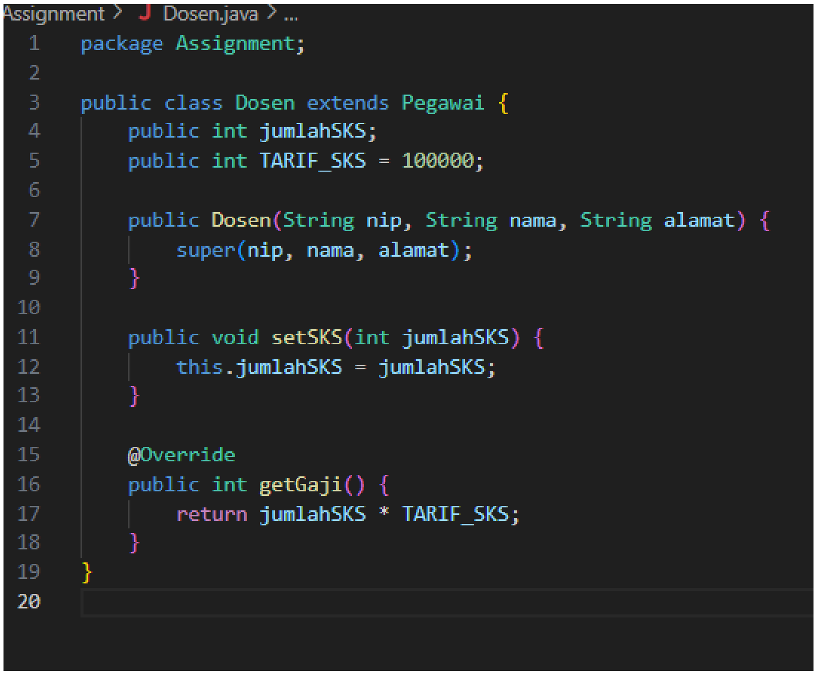
```java
package Assignment;

public class DaftarGaji {
    public Pegawai[] listPegawai;
    private int currentIndex = 0;

    public DaftarGaji(int jumlahPegawai) {
        listPegawai = new Pegawai[jumlahPegawai];
    }

    public void addPegawai(Pegawai pegawai) {
        if (currentIndex < listPegawai.length) {
            listPegawai[currentIndex] = pegawai;
            currentIndex++;
        } else {
            System.out.println(x:"List penuh, tidak bisa menambahkan pegawai.");
        }
    }

    public void printSemuaGaji() {
        for (Pegawai pegawai : listPegawai) {
            if (pegawai != null) {
                System.out.println("Nama: " + pegawai.getNama() + ", Gaji: " + pegawai.getGaji());
            }
        }
    }
}
```

Pegawai:


Assignment > Pegawai.java > Pegawai > Pegawai(String, String, String)

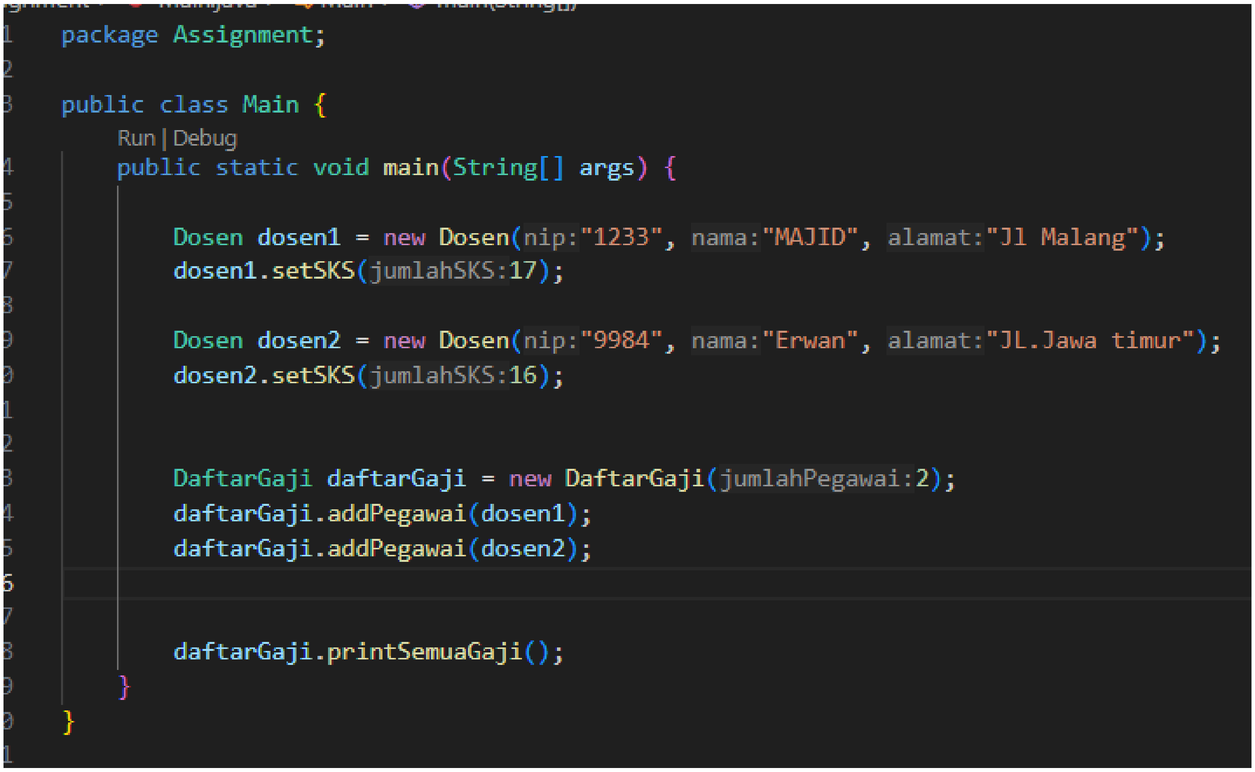
```java
package Assignment;
public class Pegawai {
    public String nip;
    public String nama;
    public String alamat;

    public Pegawai(String nip, String nama, String alamat) {
        this.nip = nip;
        this.nama = nama;
        this.alamat = alamat;
    }

    public String getNama() {
        return nama;
    }

    public int getGaji() {
        return 0;
    }
}
```

Dosen:



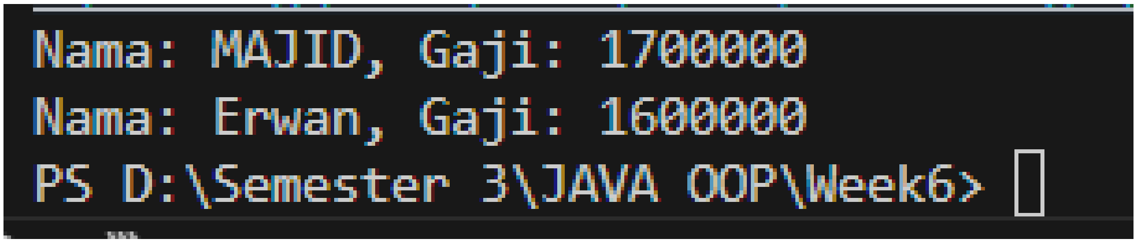
```java
package Assignment;

public class Dosen extends Pegawai {
    public int jumlahSKS;
    public int TARIF_SKS = 100000;

    public Dosen(String nip, String nama, String alamat) {
        super(nip, nama, alamat);
    }

    public void setSKS(int jumlahSKS) {
        this.jumlahSKS = jumlahSKS;
    }

    @Override
    public int getGaji() {
        return jumlahSKS * TARIF_SKS;
    }
}

```

Main:



```
package Assignment;

public class Main {
    Run | Debug
    public static void main(String[] args) {

        Dosen dosen1 = new Dosen(nip:"1233", nama:"MAJID", alamat:"Jl Malang");
        dosen1.setSKS(jumlahSKS:17);

        Dosen dosen2 = new Dosen(nip:"9984", nama:"Erwan", alamat:"JL.Jawa timur");
        dosen2.setSKS(jumlahSKS:16);


        DaftarGaji daftarGaji = new DaftarGaji(jumlahPegawai:2);
        daftarGaji.addPegawai(dosen1);
        daftarGaji.addPegawai(dosen2);


        daftarGaji.printSemuaGaji();
    }
}
```

Output:



```
Nama: MAJID, Gaji: 1700000
Nama: Erwan, Gaji: 1600000
PS D:\Semester 3\JAVA OOP\Week6>
```