

# **Object Oriented Programming**

## **Jobsheet 6 – Inheritance 1**



**Hawa Esanda**

**2241720079\_11\_TI-2I**

## PERCOBAAN 1 (extends)

```
package Percobaan1;

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

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

```
package Percobaan1;

public class ClassB {
    public int z;

    public void getNilaiZ(){
        System.out.println("nilai z:" +z);
    }
    // Menambahkan parameter untuk menerima objek ClassA
    public void getJumlah(ClassA objA) {
        // Menggunakan objek ClassA untuk mengakses x dan y
        System.out.println("jumlah:" + (objA.x + objA.y + z));
    }
}
```

```

package Percobaan1;

public class Percobaan1 {
    public static void main(String[] args) {
        ClassA objA = new ClassA();
        ClassB hitung = new ClassB();

        objA.x = 10;
        objA.y = 30;
        hitung.z = 5;

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

```

```

nilai x:10
nilai y:30
nilai z:5
jumlah:45

```

### Petanyaan

1. Pada percobaan 1 diatas program yang dijalankan terjadi error, kemudian perbaiki sehingga program tersebut bisa dijalankan dan tidak error!
2. Jelaskan apa penyebab program pada percobaan 1 ketika dijalankan terdapat error!

**Jawab :** karena mencoba mengakses variabel **x** dan **y** dari objek **hitung**, yang sebenarnya merupakan objek dari kelas **ClassB**. Namun, variabel **x** dan **y** hanya didefinisikan di dalam kelas **ClassA**. Dengan perubahan ini, **ClassB** sekarang merupakan turunan dari **ClassA**, sehingga variabel **x** dan **y** dapat diakses langsung oleh objek **hitung** dari **ClassB**.

## PERCOBAAN 2 (Hak Akses)

```
package Percobaan2;

public class ClassA {
    private int x;
    private int y;

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

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

    public int getY() {
        return y;
    }
    public void getNilai(){
        System.out.println("nilai x:" + x);
        System.out.println("nilai y:" + y);
    }
}
```

```
package Percobaan2;

public class ClassB {
    private int z;

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

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

    public void getJumlah(ClassA objA){
        // Menggunakan objek ClassA untuk mengakses x dan y
        System.out.println("jumlah:" + (objA.getX() + objA.getY() + z));
    }
}
```

```

package Percobaan2;

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

        objA.setX(20);
        objA.setY(30);
        hitung.setZ(5);

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

```

```

nilai x:20
nilai y:30
nilai Z:5
jumlah:55

```

### Pertanyaan

1. Pada percobaan 2 diatas program yang dijalankan terjadi error, kemudian perbaiki sehingga program tersebut bisa dijalankan dan tidak error!
2. Jelaskan apa penyebab program pada percobaan 1 ketika dijalankan terdapat error!

**Jawab :** penambahan metode **getX** dan **getY** di **ClassA** untuk mengakses nilai **x** dan **y**. Kemudian, di **ClassB**, kita menggunakan objek **ClassA** sebagai parameter untuk mengakses nilai **x** dan **y** dalam metode **getJumlah**.

## PERCOBAAN 3 (Super)

```
package Percobaan3;

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

```
package Percobaan3;

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));
    }
}
```

```

package Percobaan3;

public class Percobaan3 {
    public static void main(String[] args) {
        Tabung tabung = new Tabung();
        tabung.setSuperPhi(3.14);
        tabung.setSuperR(10);
        tabung.setT(3);
        tabung.volume();
    }
}

```

Volume Tabung adalah: 942.0

### Pertanyaan

1. Jelaskan fungsi “super” pada potongan program berikut di class Tabung!

```

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

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

```

**Jawab :** **super** digunakan untuk merujuk pada superclass atau kelas induk. Penggunaan **super**, dapat mengakses variabel atau metode dari superclass yang mungkin di-override atau disembunyikan oleh subclass.

2. Jelaskan fungsi “super” dan “this” pada potongan program berikut di class Tabung!

```

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

```

**Jawab :** ‘**super**’ digunakan untuk merujuk ke variabel (**phi** dan **r**) dari superclass (**Bangun**). ‘**this**’ digunakan untuk merujuk ke variabel **t** dari instance kelas **Tabung**.

3. Jelaskan mengapa pada class Tabung tidak dideklarasikan atribut “phi” dan “r” tetapi class tersebut dapat mengakses atribut tersebut!

**Jawab :** Pada class **Tabung**, atribut **phi** dan **r** tidak dideklarasikan karena class tersebut meng-extend class **Bangun**. Dengan pewarisan, **Tabung** secara otomatis mendapatkan akses ke atribut **phi** dan **r** yang telah dideklarasikan di class **Bangun**, tanpa perlu mendeklarasikannya lagi dalam class **Tabung**.

## PERCOBAAN 4 (super constructor)

```
package Percobaan4;

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

```
package Percobaan4;

public class ClassB extends
ClassA{
    ClassB(){
        System.out.println(
            "Konstruktor B dijalankan");
    }
}
```



```
package Percobaan4;

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

```
package Percobaan4;

public class Percobaan4 {
    public static void main(
String[] args) {
        ClassC test = new
ClassC();
    }
}
```

```
Konstruktor A dijalankan
Konstruktor B dijalankan
Konstruktor C dijalankan
```

### Pertanyaan

1. Pada percobaan 4 sebutkan mana class yang termasuk superclass dan subclass, kemudian jelaskan alasannya!

#### Jawab :

- ClassA adalah superclass karena tidak memiliki kelas induk. Karena, tidak ada deklarasi extends yang menunjukkan bahwa ClassA mewarisi dari kelas lain.
- ClassB adalah subclass dari ClassA karena menggunakan extends ClassA.
- ClassC adalah subclass dari ClassB karena menggunakan extends ClassB.

2. Ubahlah isi konstruktor default ClassC seperti berikut:

```

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

```

Tambahkan kata `super()` di baris Pertama dalam konstruktor defaultnya. Coba jalankan kembali class Percobaan4 dan terlihat tidak ada perbedaan dari hasil outputnya!

**Jawab :** Ya tidak ada yang berubah, karena Java secara otomatis memanggil konstruktor default superclass jika tidak ada pemanggilan eksplisit ke konstruktor superclass. Sehingga, penambahan **super()** bersifat opsional dan tidak mempengaruhi hasil output program.

3. Ubahlah isi konstruktor default ClassC seperti berikut:

```

12 public class ClassC extends ClassB{
13     ClassC(){
14         System.out.println("konstruktor C dijalankan");
15         super();
16     }
17 }

```

Ketika mengubah posisi `super()` di baris kedua dalam konstruktor defaultnya dan terlihat ada error. Kemudian kembalikan `super()` ke baris pertama seperti sebelumnya, maka errornya akan hilang. Perhatikan hasil keluaran ketika class Percobaan4 dijalankan. Kenapa bisa tampil output seperti berikut pada saat instansiasi objek test dari class ClassC.

**Jawab :** Setiap konstruktor dijalankan berurutan dari superclass ke subclass. Ini disebabkan oleh pemanggilan **super()** yang terdapat dalam konstruktor **ClassB** dan **ClassC**.

```

: Output - Percobaan4 (run)
run:
konstruktor A dijalankan
konstruktor B dijalankan
konstruktor C dijalankan
BUILD SUCCESSFUL (total time: 0 seconds)

```

Jelaskan bagaimana urutan proses jalannya konstruktor saat objek test dibuat!

**Jawab :**

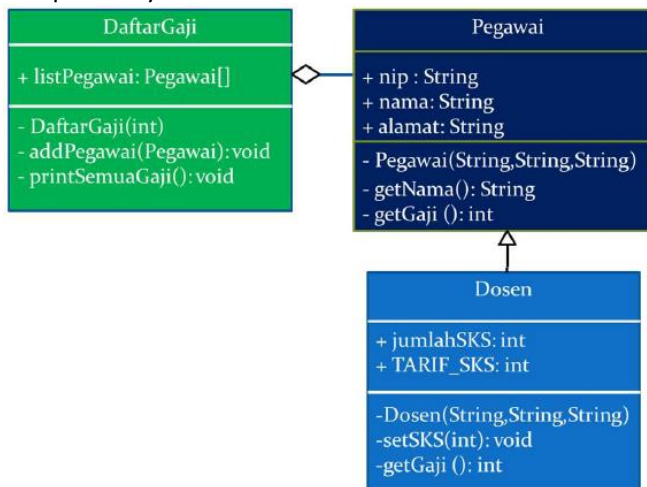
Urutan proses konstruktor saat objek **test** dari **ClassC** dibuat:

1. Objek **test** dari **ClassC** dibuat.
  2. Konstruktor **ClassC** dijalankan.
  3. Konstruktor **ClassC** memanggil konstruktor **ClassB** menggunakan **super()**.
  4. Konstruktor **ClassB** memanggil konstruktor **ClassA** menggunakan **super()**.
  5. Konstruktor **ClassA** dijalankan.
4. Apakah fungsi `super()` pada potongan program dibawah ini di ClassC!

**Jawab :** Fungsi dari **super()** pada konstruktor **ClassC** adalah untuk memanggil konstruktor dari superclass-nya, yaitu **ClassB**. Dengan memanggil **super()**, konstruktor dari **ClassB** akan dijalankan terlebih dahulu sebelum kode dalam konstruktor **ClassC** dijalankan.

## Tugas

1. Buatlah sebuah program dengan konsep pewarisan seperti pada class diagram berikut ini. Kemudian buatlah instansiasi objek untuk menampilkan data nama pegawai dan gaji yang didapatkannya.



Jawab :

```
package tugas;

public class DaftarGaji {
    Pegawai[] listPegawai;

    DaftarGaji(int jumlahPegawai){
        listPegawai = new Pegawai[jumlahPegawai];
    }
    public void addPegawai(Pegawai pegawai){
        for (int i = 0; i < listPegawai.length; i++) {
            if (listPegawai[i] == null) {
                listPegawai[i] = pegawai;
                break;
            }
        }
    }
    public void printSemuaGaji(){
        for (Pegawai pegawai : listPegawai) {
            if (pegawai != null) {
                System.out.println("-----");
                System.out.println("NIP\t: " + pegawai.nip);
                System.out.println("Nama\t: " + pegawai.getNama());
                System.out.println("Alamat\t: " + pegawai.alamat);
                System.out.println("Gaji\t: Rp." + pegawai.getGaji());
                System.out.println("-----");
                System.out.println("");
            }
        }
    }
}
```

```

package tugas;

public class Pegawai {
    String nip;
    String nama;
    String alamat;

    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 5000000;
    }
}

```

```

package tugas;

public class Dosen extends Pegawai{
    int jumlahSKS;
    int TARIF_SKS = 1000000;

    Dosen(String nip, String nama, String alamat){
        super(nip, nama, alamat);
    }
    public void setSKS(int jumlahSKS){
        this.jumlahSKS = jumlahSKS;
    }
    public int getGaji(){
        return super.getGaji() + (jumlahSKS *
TARIF_SKS);
    }
}

```

```
package tugas;

public class tugaMain {
    public static void main(String[] args) {
        DaftarGaji dg = new DaftarGaji(5);
        Pegawai pegawai = new Pegawai("123456", "Lengkara", "Jl. Malang 1");
        Dosen d1 = new Dosen("123789", "Argantara", "Jl. Surabaya 2");

        d1.setSKS(10);
        dg.addPegawai(pegawai);
        dg.addPegawai(d1);
        dg.printSemuaGaji();

    }
}
```

```
-----
NIP      : 123456
Nama     : Lengkara
Alamat   : Jl. Malang 1
Gaji     : Rp.5000000
-----
```

```
-----
NIP      : 123789
Nama     : Argantara
Alamat   : Jl. Surabaya 2
Gaji     : Rp.15000000
-----
```