

# JOBSHEET 6

## INHERITANCE (PEWARISAN)

### 1. KOMPETENSI

1. Memahami konsep dasar inheritance atau pewarisan.
2. Mampu membuat suatu subclass dari suatu superclass tertentu.

### 2. PENDAHULUAN

**Inheritance** atau **pewarisan sifat** merupakan suatu cara untuk menurunkan suatu class yang lebih umum menjadi suatu class yang lebih spesifik. Inheritance adalah salah satu ciri utama suatu bahasa program yang berorientasi pada objek. Inti dari pewarisan adalah sifat reusable dari konsep object oriented. Setiap **subclass** akan “mewarisi” sifat dari **superclass** selama bersifat protected ataupun public.

Dalam inheritance terdapat dua istilah yang sering digunakan. Kelas yang menurunkan disebut kelas dasar (**base class/super class**), sedangkan kelas yang diturunkan disebut kelas turunan (**derived class/sub class/child class**) . Di dalam Java untuk mendeklarasikan suatu class sebagai subclass dilakukan dengan cara menambahkan kata kunci **extends** setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class. Berikut adalah contoh deklarasi inheritance.

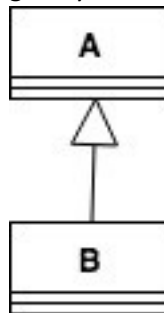
```
public class B extends A {  
    ...  
}
```

Contoh diatas memberitahukan kompiler Java bahwa kita ingin meng-extend class A ke class B. Dengan kata lain, class B adalah subclass (class turunan) dari class A, sedangkan class A adalah parent class dari class B.

Karakteristik pada super class akan dimiliki juga oleh subclassnya. Terdapat 3 bentuk pewarisan: single inheritance, multilevel inheritance, dan multiple inheritance. Namun yang akan dibahas pada jobsheet ini adalah single inheritance dan multilevel inheritance.

#### 1. Single Inheritance

Single inheritance adalah Suatu class yang hanya mempunyai satu parent class. Contoh:



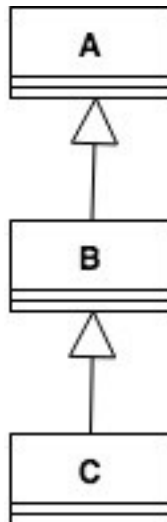
Gambar 1. Contoh Single Inheritance

Berdasarkan Gambar 1 dapat diketahui bahwa class B merupakan subclass yang mempunyai satu parent yaitu class A sehingga disebut single inheritance.

## 2. Multilevel Inheritance

Multilevel inheritance adalah Suatu subclass bisa menjadi superclass bagi class yang lain.

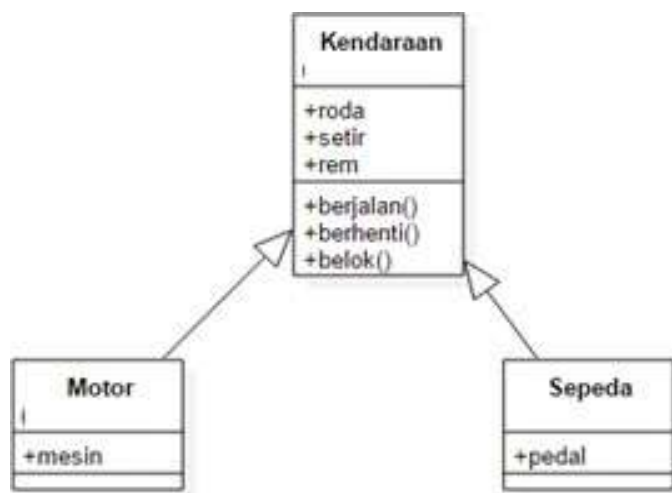
Contoh:



Gambar 2. Contoh Multilevel Inheritance

Berdasarkan Gambar 2 diatas dapat dilihat bahwa class B merupakan subclass dari class A, sehingga dalam hal ini class A adalah superclass dan class B adalah subclass. Kemudian class B yang awalnya merupakan subclass mempunyai subclass lagi yaitu class C sehingga class B menjadi superclass dari class C, begitu juga seterusnya jika class C memiliki subclass lagi.

Pada class diagram, pewarisan digambarkan dengan sebuah garis tegas, dengan segitiga di ujungnya. Class yang dekat pada segitiga merupakan superclass, sedangkan class yang jauh dari segitiga merupakan subclass. Untuk membentuk sebuah subclass, keyword “extends” digunakan (lihat contoh pada sesi “Implementasi Pewarisan”). Berikut ini adalah contoh class diagram dari pewarisan:



Gambar 3 Contoh class diagram dalam inheritance

Suatu parent class dapat tidak mewariskan sebagian member-nya kepada subclass-nya. Sejauh mana suatu member dapat diwariskan ke class lain, ataupun suatu member dapat diakses dari class lain, sangat berhubungan dengan access control (kontrol pengaksesan). Di dalam java, kontrol pengaksesan dapat digambarkan dalam tabel berikut ini:

Modifier	class yang sama	package yang sama	subclass	class manapun
private	√			
default	√	√		
protected	√	√	√	
public	√	√	√	√

Kata kunci super dipakai untuk merujuk pada member dari parent class. Sebagaimana kata kunci this yang dipakai untuk merujuk pada member dari class itu sendiri. Format penulisannya adalah sebagai berikut:

- **super.namaAtribut**  
Merujuk/mengakses atribut dari parent class /superclass
- **super.namaMethod()**  
Merujuk/memanggil method dari parent class /superclass
- **super()**  
Merujuk / memanggil konstruktor parent class /superclass  
Hanya bisa digunakan dibaris pertama dalam konstruktor.
- **super(parameter1, parameter2,dst)**  
Merujuk / memanggil konstruktor berparamter dari superklas

Ketika mmebuat objek dari subclass, pada saat itu juga objek pada superclass juga akan terbentuk. Dengan katalain, ketika konstruktor subclass dijalankan untuk membuat objek, saat itu juga konstruktor superclass akan berjalan. Jadi di setiap konstruktor subclass, pada baris pertama konstruktor subclass tersebut akan dipanggil konstruktor superclass. Sebelum subclass menjalankan konstruktornya sendiri, subclass akan menjalankan konstruktor superclass terlebih dahulu.

### 3. PERCOBAAN 1 (extends)

#### A. TAHAPAN PERCOBAAN

1. Buatlah sebuah class parent/superclass dengan nama ClassA.java

```

0  |  *
1  |  * @author WINDOWS 10
2  |  */
3  |  public class ClassA {
4  |      public int x;
5  |      public int y;
6  |
7  |      public void getNilai(){
8  |          System.out.println("nilai x:" + x );
9  |          System.out.println("nilai y:" + y);
10 |      }
11 |  }
12 |

```

2. Buatlah sebuah class anak/subclass dengan nama ClassB.java

```

    * @author WINDOWS 10
    */
    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));
        }
    }

```

3. Buatlah class Percobaan1.java untuk menjalankan program diatas!

```

    * @author WINDOWS 10
    */
    public class Percobaan1 {

        /**
         * @param args the command line arguments
         */
        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();
        }
    }

```

4. Jalankan program diatas, kemudian amati apa yang terjadi!

## B. PERTANYAAN

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!

.....

.....

.....

## 4. PERCOBAAN 2 (Hak Akses)

### A. TAHAPAN PERCOBAAN

1. Buatlah sebuah class parent/superclass dengan nama ClassA.java

```
10 | *  
11 |  * @author WINDOWS 10  
12 |  */  
13 |  
14 | public class ClassA {  
15 |     private int x;  
16 |     private int y;  
17 |  
18 |     public void setX(int x){  
19 |         this.x = x;  
20 |     }  
21 |     public void setY(int y){  
22 |         this.y = y;  
23 |     }  
24 |     public void getNilai(){  
25 |         System.out.println("nilai x:" + x );  
26 |         System.out.println("nilai y:" + y);  
27 |     }  
28 | }
```

2. Buatlah sebuah class anak/subclass dengan nama ClassB.java

```
1 |  * @author WINDOWS 10  
2 |  */  
3 |  
4 | public class ClassB {  
5 |     private int z;  
6 |  
7 |     public void setZ(int z){  
8 |         this.z = z;  
9 |     }  
10 |  
11 |     public void getNilaiZ(){  
12 |         System.out.println("nilai Z:" + z);  
13 |     }  
14 |  
15 |     public void getJumlah(){  
16 |         System.out.println("jumlah:" + (x+y+z));  
17 |     }  
18 | }
```

3. Buatlah class Percobaan2.java untuk menjalankan program diatas!

```

    * @author WINDOWS 10
    */
    public class Percobaan2 {

        /**
         * @param args the command line arguments
         */
        public static void main(String[] args) {

            ClassB hitung = new ClassB();
            hitung.setX(20);
            hitung.setY(30);
            hitung.setZ(5);
            hitung.getNilai();
            hitung.getNilaiZ();
            hitung.getJumlah();

        }

    }

```

4. Jalankan program diatas, kemudian amati apa yang terjadi!

## B. 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!

.....

.....

.....

## 5. PERCOBAAN 3 (Super)

### A. TAHAPAN PERCOBAAN

1. Buatlah sebuah class parent/superclass dengan nama Bangun.java

```

10      * @author WINDOWS 10
11      */
12      public class Bangun {
13          protected double phi;
14          protected int r;
15      }
16
17

```

2. Buatlah sebuah class anak/subclass dengan nama Tabung.java

```

11  L  */
12  public class Tabung extends Bangun{
13      protected int t;
14      public void setSuperPhi(double phi){
15          super.phi = phi;
16      }
17      public void setSuperR(int r){
18          super.r = r;
19      }
20      public void setT(int t){
21          this.t = t;
22      }
23
24      public void volume(){
25          System.out.println("Volume Tabung adalah: "+(super.phi*super.r*super.r*this.t));
26      }
27  }
28
29

```

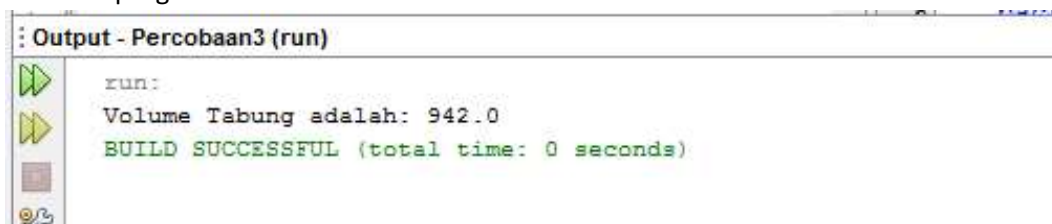
3. Buatlah class Percobaan3.java untuk menjalankan program diatas!

```

11  L  */
12  public class Percobaan3 {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          Tabung tabung=new Tabung();
20          tabung.setSuperPhi(3.14);
21          tabung.setSuperR(10);
22          tabung.setT(3);
23          tabung.volume();
24      }
25  }
26
27

```

4. Jalankan program diatas!



```

Output - Percobaan3 (run)
run:
Volume Tabung adalah: 942.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

## B. 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;
}

```

- .....
- .....
- .....
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));  
}
```

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

## 6. PERCOBAAN 4 (super constructor)

### A. TAHAPAN PERCOBAAN

1. Buatlah tiga file dengan nama ClassA.java , ClassB.java , dan ClassC.java, seperti pada kode program dibawah ini!

ClassA.java

```
10 | * @author WINDOWS 10  
11 | */  
12 |  
13 | public class ClassA {  
14 |     ClassA() {  
15 |         System.out.println("konstruktor A dijalankan");  
16 |     }  
17 | }
```

ClassB.java

```
9 | *  
10 | * @author WINDOWS 10  
11 | */  
12 |  
13 | public class ClassB extends ClassA {  
14 |     ClassB() {  
15 |         System.out.println("konstruktor B dijalankan");  
16 |     }  
17 | }
```

ClassC.java



```

9      *
10     * @author WINDOWS 10
11     */
12     public class ClassC extends ClassB{
13     -   ClassC(){
14     -       System.out.println("konstruktor C dijalankan");
15     -   }
16     }
17

```

2. Buatlah class Percobaan4.java untuk menjalankan program diatas!

```

10     * @author WINDOWS 10
11     */
12     public class Percobaan4 {
13
14     -   /**
15     -       * @param args the command line arguments
16     -       */
17     -   public static void main(String[] args) {
18     -       // TODO code application logic here
19     -       ClassC test= new ClassC();
20     -   }
21
22     }
23

```

3. Jalankan program kemudian amati apa yang terjadi!

```

Output - Percobaan4 (run)

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

```

## B. PERTANYAAN

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

.....

.....

.....

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!

3. Ubah 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() dibaris kedua dalam konstruktor defaultnya dan terlihat ada error. Kemudian kembalikan super() kebaris 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

```

: 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!

.....

.....

.....

4. Apakah fungsi super() pada potongan program dibawah ini di ClassC!

```

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

```

.....

.....

.....

## 9. 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.

