

# JOBSHEET 6

## INHERITANCE (PEWARISAN)

Nama : Reza Kurniawan  
Kelas : SIB 2E  
NIM : 2241760059

### 1. 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

```
0  |  *
1  |  * @author WINDOWS 10
2  |  */
3  |  public class ClassB {
4  |      public int z;
5  |
6  |      public void getNilaiZ(){
7  |          System.out.println("nilai Z:" + z);
8  |      }
9  |
10 |      public void getJumlah(){
11 |          System.out.println("jumlah:" + (x+y+z));
12 |      }
13 |  }
14 |
```

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

```

L  */
  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!

Jawaban :

Modifikasi

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

Setelah dilakukan modifikasi berupa penambahan extends Class Amaka code akan bisa dijalankan.

2. Jelaskan apa penyebab program pada percobaan 1 ketika dijalankan terdapat error!

Jawaban : fungsi extends digunakan untuk membangun hubungan pewarisan antar kelas. Ketika sebuah subclass tidak menambahkan extends terhadap superclassnya, maka Subclass tidak memiliki akses langsung ke atribut dan metode yang didefinisikan dalam superclass. Hasilnya jika subclass mencoba mengakses atribut atau metode superclass, compiler akan menghasilkan 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  public class ClassB {
4      private int z;
5
6      public void setZ(int z){
7          this.z = z;
8      }
9      public void getNilaiZ(){
10         System.out.println("nilai Z: "+ z);
11     }
12     public void getJumlah(){
13         System.out.println("jumlah: "+ (x+y+z));
14     }
15 }
```

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!

Jawaban :

Modifikasi ClassB dengan menambahkan extends ClassA agar dapat mewarisi attribute x dan y yang terdapat di ClassA.

```

public class ClassB extends ClassA {
    private int z;
}

```

Akan tetapi, meskipun ClassB sudah mewarisi ClassA, variabel x dan y dideklarasikan sebagai private, yang berarti tidak dapat diakses secara langsung di kelas turunan seperti ClassB. Sehingga perlu adanya perubahan pada modifier pada attribute x dan y pada ClassA menjadi protected atau public

```

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

```

2. Jelaskan apa penyebab program pada percobaan 1 ketika dijalankan terdapat error!  
Jawaban : Karena tidak adanya extends yang berfungsi untuk membangun hubungan pewarisan antar ClassB dan ClassA. Tidak hanya itu, attribute pada ClassA di set private sehingga pewarisnya atau ClassB tidak memiliki hak akses terhadap attribute tersebut.

## 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;
}

```

Jawaban : Penggunaan Kata kunci super dipakai untuk merujuk pada member dari parent class. Jadi, pada "super.phi = phi" adalah kita merujuk pada Class Parentnya adalah Class Bangun dengan variabel yang kita target adalah phi. Lalu, pada "super.r = r" juga sama kita merujuk pada Class Parentnya adalah Class Bangun dengan variabel yang kita target adalah 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));  
}
```

Jawaban : Fungsi "super" untuk merujuk pada member dari parent class. Sedangkan "this" merujuk pada member dari class itu sendiri. Jadi, pada "super.phi" dan "super.r" kita merujuk pada parent class yaitu class Bangun dengan member phi dan r. Lalu, pada "this.t" kita merujuk pada class itu sendiri yaitu class Tabung dengan atribut t.

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

Jawaban : Karena class Tabung merupakan hasil pewarisan dari class Bangun. Yang dimana class Tabung bisa mengakses atribut atau method yang ada pada class Bangun tanpa perlu mendeklarasikan pada class Tabung sendiri. Jadi, hanya tinggal memanggil saja atribut atau method yang dibutuhkan dari class Bangun.

## 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!

Jawaban :

a. ClassA

- ClassA adalah superclass dalam hierarki ini.
- ClassA tidak meng-extend (mewarisi) dari kelas lain.
- ClassA adalah kelas dasar atau paling atas dalam hierarki ini.
- Tidak ada kelas yang meng-extend dari ClassA.

b. ClassB

- ClassB adalah subclass dari ClassA.
- ClassB meng-extend (mewarisi) dari ClassA.
- Ini berarti ClassB mewarisi sifat dan perilaku dari ClassA.
- ClassB adalah anak dari ClassA dalam hierarki ini.



c. ClassC

- ClassC adalah subclass dari ClassB.
- ClassC meng-extend (mewarisi) dari ClassB.
- Dengan demikian, ClassC secara tidak langsung juga merupakan subclass dari ClassA melalui rantai pewarisan (inheritance) dari ClassB.
- Ini berarti ClassC mewarisi sifat dan perilaku dari ClassA melalui ClassB.
- ClassC adalah cucu dari ClassA dalam hierarki ini.

Dari penjelasan diatas didapat sebuah rangkuman berikut

- ClassA adalah superclass (kelas induk).
- ClassB adalah subclass dari ClassA.
- ClassC adalah subclass dari ClassB dan secara tidak langsung juga merupakan subclass dari ClassA melalui pewarisan (inheritance) dari 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!

3. Ublah 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!

Jawaban : Jadi urutan prosesnya seperti berikut

- Objek test dibuat menggunakan konstruktor ClassC. Oleh karena itu, konstruktor ClassC akan dijalankan terlebih dahulu.
- Dalam konstruktor ClassC, terdapat pemanggilan super(), yang mengakibatkan pemanggilan konstruktor dari kelas induk, yaitu ClassB. Oleh karena itu, konstruktor ClassB dijalankan sebelum



kembali ke konstruktor ClassC.

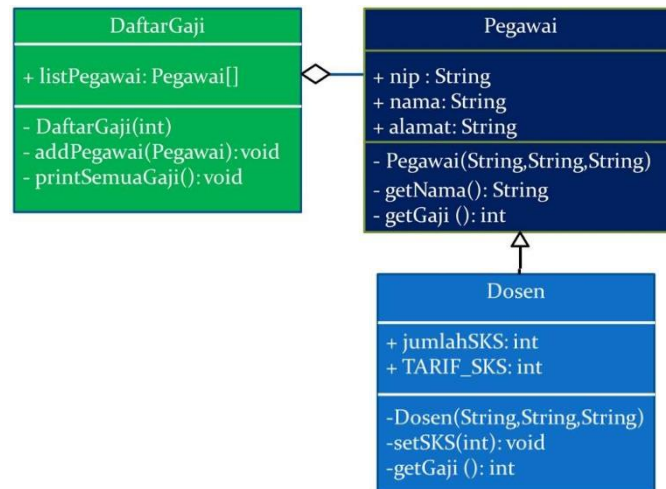
- c. Konstruktor ClassB memanggil super(), yang kembali memanggil konstruktor dari kelas induk teratas, yaitu ClassA. Oleh karena itu, konstruktor ClassA dijalankan terlebih dahulu.
  - d. Konstruktor ClassA mencetak pesan "Konstruktor A dijalankan."
  - e. Setelah konstruktor ClassA selesai dijalankan, program kembali ke konstruktor ClassB, yang kemudian mencetak pesan "Konstruktor B dijalankan."
  - f. Kemudian, program melanjutkan ke konstruktor ClassC dan mencetak pesan "Konstruktor C dijalankan."
4. Apakah fungsi super() pada potongan program dibawah ini di ClassC!

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

Jawaban : Kata kunci super() digunakan untuk memanggil konstruktor dari kelas induk (superclass) dalam kelas turunan (subclass). Dengan kata lain, super() memastikan bahwa konstruktor kelas induk dijalankan sebelum konstruktor kelas turunan dalam rantai pewarisan.

## 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.



Jawaban :

Class Main.java

```
1 package Tugas;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner reza = new Scanner(System.in);
7         DaftarGaji daftarGaji = new DaftarGaji();
8
9         while (true) {
10             System.out.println();
11             System.out.println("*****");
12             System.out.println("Menu:");
13             System.out.println("1. Tambah Dosen");
14             System.out.println("2. Tampilkan Gaji Pegawai");
15             System.out.println("3. Keluar");
16             System.out.println("*****");
17             System.out.print("Pilihan: ");
18             int choice = reza.nextInt();
19
20             switch (choice) {
21                 case 1:
22                     System.out.println("*****");
23                     System.out.println("Masukkan data Dosen:");
24                     System.out.print("NIP: ");
25                     String nip = reza.next();
26                     System.out.print("Nama: ");
27                     String nama = reza.next();
28                     reza.nextLine(); // Consume the newline character
29                     System.out.print("Alamat: ");
30                     String alamat = reza.nextLine();
31                     Dosen dosen = new Dosen(nip, nama, alamat);
32
33                     System.out.print("Jumlah SKS: ");
34                     int sks = reza.nextInt();
35                     dosen.setSKS(sks);
36                     System.out.print("Tarif SKS: ");
37                     int tarif = reza.nextInt();
38                     dosen.setTarif(tarif);
39
40                     daftarGaji.addPegawai(dosen);
41                     System.out.println("Dosen ditambahkan!");
42                     break;
43
44                 case 2:
45                     System.out.println();
46                     System.out.println();
47                     System.out.println("Daftar Gaji Pegawai:");
48                     daftarGaji.printSemuaGaji();
49                     System.out.println();
50                     break;
51
52                 case 3:
53                     System.out.println("Terima kasih!");
54                     System.exit(0);
55
56                 default:
57                     System.out.println("*****");
58                     System.out.println("Pilihan tidak valid. Silakan coba lagi.");
59                     System.out.println("*****");
60             }
61         }
62     }
63 }
64 }
65 }
```


## Class DaftarGaji.java

```
1  package Tugas;
2
3  import java.util.ArrayList;
4
5  public class DaftarGaji {
6      private ArrayList<Pegawai> listPegawai = new ArrayList<Pegawai>();
7
8      public DaftarGaji() {
9
10     }
11
12     public void addPegawai(Pegawai pegawai) {
13         listPegawai.add(pegawai);
14     }
15
16     public void printSemuaGaji() {
17         System.out.println("-----");
18         System.out.printf("| %-20s | %-10s |\n", "Nama Pegawai", "Gaji");
19         System.out.println("-----");
20         for (Pegawai pegawai : listPegawai) {
21             System.out.printf("| %-20s | %-10s |\n", pegawai.getNama(), pegawai.getGaji());
22         }
23         System.out.println("-----");
24     }
25 }
26
27
```

## Class Dosen.java

```
1  package Tugas;
2
3  public class Dosen extends Pegawai {
4      private int jumlahSKS, TARIF_SKS;
5
6      public Dosen(String nip, String nama, String alamat) {
7          super(nip, nama, alamat);
8      }
9
10     public void setSKS(int sks) {
11         jumlahSKS = sks;
12     }
13
14     public void setTarif(int tarif) {
15         TARIF_SKS = tarif;
16     }
17
18     @Override
19     public int getGaji() {
20         return jumlahSKS * TARIF_SKS;
21     }
22 }
23
```

## Class Pegawai.java



```
1  package Tugas;
2
3  public class Pegawai {
4      public String nip, nama, alamat;
5
6      public Pegawai(String nip, String nama, String alamat) {
7          this.nip = nip;
8          this.nama = nama;
9          this.alamat = alamat;
10     }
11
12     public String getNama() {
13         return nama;
14     }
15
16     public int getGaji() {
17         return 0;
18     }
19 }
```