

PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

“Inheritance”



DOSEN PEMBIMBING

Septian Enggar Sukmana, S.Pd., M.T.

Disusun oleh :

Muhammad Reza Khatami

(2041720076)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG 2021

Jobsheet 6

Pertanyaan!

🔧 Percobaan 1

1. Pada percobaan 1 diatas program yang dijalankan terjadi error, kemudian perbaiki sehingga program tersebut bisa dijalankan dan tidak error!

Jawab :

- ClassB.java

```
1 package jobsheet_6_pbo;
2
3 public class ClassB extends ClassA{
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 }
```

Pada ClassB ditambah kata kunci extends

- Output

```
run:
nilai x: 20
nilai y: 30
nilai z: 5
jumlah : 55
BUILD SUCCESSFUL (total time: 2 seconds)
```

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

Jawab : Program tersebut eror karena tidak ada kata kunci **extends** yang diikuti nama parent class-nya pada subclass.

🔧 Percobaan 2

1. Pada percobaan 2 diatas program yang dijalankan terjadi error, kemudian perbaiki sehingga program tersebut bisa dijalankan dan tidak error!

Jawab :

- ClassA.java

```

1  package Percobaan2;
2
3  public class ClassA {
4      private int x;
5      private int y;
6
7      public void setX(int x) {
8          this.x = x;
9      }
10
11     public void setY(int y) {
12         this.y = y;
13     }
14
15     public int getX() {
16         return x;
17     }
18
19     public int getY() {
20         return y;
21     }
22
23     public void getNilai() {
24         System.out.println("nilai x: " + x);
25         System.out.println("nilai y: " + y);
26     }
27 }

```

Pada ClassA ditambah method getX() dan getY()

- ClassB.java

```

1  package Percobaan2;
2
3  public class ClassB extends ClassA {
4      private int z;
5
6      public void setZ(int z) {
7          this.z = z;
8      }
9
10     public void getNilaiZ() {
11         System.out.println("nilai z: " + z);
12     }
13
14     public void getJumlah() {
15         System.out.println("jumlah: " + (getX()+getY()+z));
16     }
17 }

```

Pada ClassB ditambah kata kunci extends dan pada method get jumlah pemanggilannya diubah menjadi (getX()+getY()+z)

- Output

```

run:
nilai x: 20
nilai y: 30
nilai z: 5
jumlah: 55
BUILD SUCCESSFUL (total time: 0 seconds)

```

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

Jawab : Program eror karena pada ClassA tidak ada method getX() dan getY() kemudian pada ClassB tidak ada kata kunci extends dan pemanggilan pada method getJumlah belum diubah.

✚ Percobaan 3

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 : Kata kunci **super** dipakai untuk merujuk pada member dari parent class yaitu atribut phi dan 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));  
}
```

Jawab : : Kata kunci **super** dipakai untuk merujuk pada member dari parent class yaitu atribut phi dan r sedangkan **this** dipakai untuk merujuk pada member dari class itu sendiri yaitu t.

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

Jawab : Karena class tabung merupakan child class dari class bangun dengan ditandai oleh kata kunci **extends** dan class tabung bisa mengakses atribut dari class bangun menggunakan kata kunci **super**.

✚ Percobaan 4

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

Jawab : Yang merupakan **superclass** adalah **ClassA** karena mewariskan ke **ClassB** dan **ClassB** karena mewariskan ke **ClassC** sedangkan yang merupakan **subclass** adalah **ClassB** karena ada kata kunci **extends ClassA** dan **ClassC** karena ada kata kunci **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 :

- ClassC.java

```
1 package Percobaan4;
2
3 public class ClassC extends ClassB {
4
5     public ClassC() {
6         super();
7         System.out.println("konstruktor C dijalankan");
8     }
9 }
10
11 }
```

- Output

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

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)
```

Jawab : Karena pada konstruktor ClassC ada kata kunci **super** yang digunakan untuk mengakses member dari parent class

- Jelaskan bagaimana urutan proses jalannya konstruktor saat objek test dibuat!

Jawab : Pertama konstruktor ClassA dijalankan kemudian dilanjutkan oleh konstruktor ClassB kemudian pada ClassC konstruktor ClassA dan B dipanggil menggunakan kata kunci `super()` dan selanjutnya print dari konstruktor ClassC.

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

```

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

```

Jawab : Fungsi dari super() adalah untuk mengakses member dari parent class yaitu ClassA dan ClassB

Tugas

- Pegawai_15.java

```

1 package Tugas_15;
2
3 public class Pegawai_15 {
4     private String nip;
5     private String nama;
6     private String alamat;
7
8     public Pegawai_15(String nip, String nama, String alamat) {
9         this.nip = nip;
10        this.nama = nama;
11        this.alamat = alamat;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public int getGaji(){
19        return 5000000;
20    }
21 }

```

- Dosen_15.java

```

1 package Tugas_15;
2
3 public class Dosen_15 extends Pegawai_15{
4
5     private int jumlahSKS;
6     private int TARIF_SKS;
7
8     public Dosen_15(String nip, String nama, String alamat) {
9         super(nip, nama, alamat);
10    }
11
12    public void setSKS(int jumlahSKS) {
13        this.jumlahSKS = jumlahSKS;
14    }
15
16    public int getGaji(){
17        TARIF_SKS = 3000000;
18        int gaji_SKS = jumlahSKS * TARIF_SKS;
19        return super.getGaji() + gaji_SKS;
20    }
21 }

```

- DaftarGaji_15.java

```

1 package Tugas_15;
2
3 public class DaftarGaji_15 {
4     private Pegawai_15[] listPegawai;
5
6     public DaftarGaji_15(int listPegawai) {
7         this.listPegawai = new Pegawai_15[listPegawai];
8         this.initlistPegawai();
9     }
10
11     private void initlistPegawai(){
12         for (int i = 0; i < this.listPegawai.length; i++) {
13             this.listPegawai[i] = new Pegawai_15(null, null, null);
14         }
15     }
16
17     public void addPegawai(Pegawai_15 listPegawai){
18         for (int i = 0; i < this.listPegawai.length; i++) {
19             if (this.listPegawai[i].getNama() == null){
20                 this.listPegawai[i] = listPegawai;
21                 break;
22             }
23         }
24     }
25
26     public void printSemuaGaji(){
27         for (int i = 0; i < this.listPegawai.length; i++) {
28             if (this.listPegawai[i].getNama() != null){
29                 System.out.println("Nama : " + listPegawai[i].getNama());
30                 System.out.println("Gaji : " + listPegawai[i].getGaji() + "\n");
31             }
32         }
33     }
34 }

```

- MainTugas_15.java

```

1 package Tugas_15;
2
3 public class MainTugas_15 {
4
5     public static void main(String[] args) {
6         Pegawai_15 pg1 = new Pegawai_15("111", "Reza", "Malang");
7         Dosen_15 ds1 = new Dosen_15("222", "Khatami", "Malang");
8         ds1.setSKS(3);
9         DaftarGaji_15 gaji = new DaftarGaji_15(2);
10        gaji.addPegawai(pg1);
11        gaji.addPegawai(ds1);
12        gaji.printSemuaGaji();
13    }
14 }

```

- Output

```

run:
Nama      : Reza
Gaji      : 5000000

Nama      : Khatami
Gaji      : 5900000

BUILD SUCCESSFUL (total time: 0 seconds)

```