

OOP

Jobsheet 6



Name: Steven Christian Susanto

NIM/Number: 2241720003/24

Class: 2I

Major: Information Technology

Study Program: D-IV Informatics Engineering

Percobaan 1

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

a. Code

```
package Experiment1;

public class ClassB extends ClassA{

    public int z;

    public ClassB(){

        super();

    }

    public void getNilaiZ(){

        System.out.println("nilai Z:"+z);

    }

    public void getJumlah(){

        System.out.println("Jumlah:" + (x+y+z));

    }

}
```

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

a. Kode error karena ClassB belum terhubung sebagai subclass ClassA.

Percobaan 2

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

a. ClassA

```
package Experiment2;

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);
    }
}
```

b. ClassB

```
package Experiment2;

public class ClassB extends ClassA{

    private int z, x, y;

    public ClassB() {

        x = super.getX();

        y = super.getY();

    }

    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));

    }

}
```

2. Jelaskan apa penyebab program pada percobaan 2 ketika dijalankan terdapat error!
 - a. Karena ClassB belum menjadi subclass ClassA, dan untuk mengakses nilai x dan y perlu ada getter di ClassA.

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;  
}
```

- a. Super.phi dan super.r digunakan untuk mengambil nilai dari atribut yang berada di dalam parent class.
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));  
}
```

- a. Super merujuk pada atribut yang ada di parent class, sedangkan this merujuk pada atribut yang ada pada kelas itu sendiri.
3. Jelaskan mengapa pada class Tabung tidak dideklarasikan atribut “phi” dan “r” tetapi class tersebut dapat mengakses atribut tersebut!
 - a. Karena phi dan r merupakan atribut turunan dari parent class dari class Tabung.

Percobaan 4

1. Pada percobaan 4 sebutkan mana class yang termasuk superclass dan subclass, kemudian jelaskan alasannya!
 - a. Superclass: ClassA dan ClassB karena nilai kedua class ini diturunkan kepada class selanjutnya.
 - b. Subclass: ClassB dan ClassC karena kedua class ini mendapat nilai turunan dari class di atasnya.
2. Ubahlah isi konstruktor default ClassC seperti berikut

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

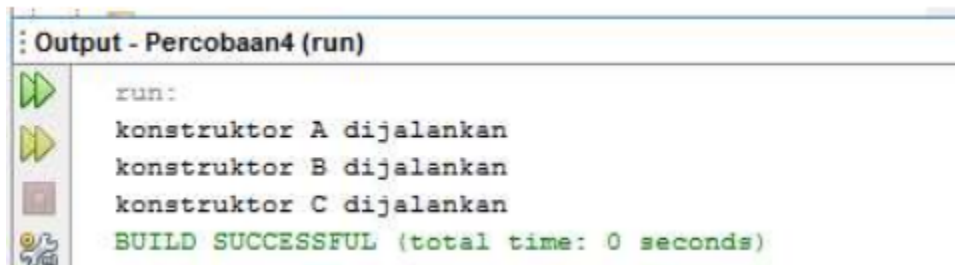
Tambahkan kata super() di baris pertama dalam konstruktor defaultnya. Coba jalankan kembali class Percobaan4 dan terlihat tidak ada perbedaan dari hasil outputnya!

- a. Tidak ada perbedaan.
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() 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

- a. Konstruktor ClassC dijalankan.
 - b. Konstruktor ClassC secara eksplisit memanggil konstruktor ClassB menggunakan `super()`;
 - c. Konstruktor ClassB dijalankan, yang pada gilirannya memanggil konstruktor ClassA.
 - d. Konstruktor ClassA dijalankan.
 - e. Kontrol kembali ke konstruktor ClassB setelah konstruktor ClassA selesai, dan pesan "konstruktor B dijalankan" dicetak.
 - f. Kontrol kembali ke konstruktor ClassC setelah konstruktor ClassB selesai, dan pesan "konstruktor C dijalankan" dicetak.
4. Apakah fungsi `super()` pada potongan program di bawah ini di ClassC!
- a. Fungsinya adalah untuk memanggil konstruktor dari superclass.

Tugas

```
class Pegawai {  
  
    private String nip;  
  
    private String nama;  
  
    private String alamat;  
  
  
    public Pegawai(String nip, String nama, String alamat) {  
  
        this.nip = nip;  
  
        this.nama = nama;  
  
        this.alamat = alamat;  
  
    }  
  
  
    public String getName() {  
  
        return nama;  
  
    }  
  
  
    public double getGaji() {  
  
        return 0.0;  
  
    }  
  
}
```

```
class Dosen extends Pegawai {  
  
    private int jumlahSKS;  
  
    private static final double TARIF_SKS = 50000.0;  
  
  
    public Dosen(String nip, String nama, String alamat) {
```

```

        super(nip, nama, alamat);

        this.jumlahSKS = 0;
    }

    public void setSKS(int jumlahSKS) {

        this.jumlahSKS = jumlahSKS;
    }

    @Override

    public double getGaji() {

        return jumlahSKS * TARIF_SKS;
    }
}

class DaftarGaji {

    private Pegawai[] listPegawai;

    public DaftarGaji(int jumlahPegawai) {

        listPegawai = new Pegawai[jumlahPegawai];
    }

    public void addPegawai(int index, Pegawai pegawai) {

        listPegawai[index] = pegawai;
    }

    public void printSemuaGaji() {

```

```
        for (Pegawai pegawai : listPegawai) {

            System.out.println("Nama Pegawai: " + pegawai.getNama());

            System.out.println("Gaji: " + pegawai.getGaji());

            System.out.println();

        }

    }

}
```

```
public class Main {

    public static void main(String[] args) {

        DaftarGaji daftarGaji = new DaftarGaji(2);

        Dosen dosen = new Dosen("123", "John Doe", "Jl. ABC");

        Pegawai pegawai = new Pegawai("456", "Jane Doe", "Jl. XYZ");

        dosen.setSKS(8);

        daftarGaji.addPegawai(0, dosen);

        daftarGaji.addPegawai(1, pegawai);

        daftarGaji.printSemuaGaji();

    }

}
```