



NAMA : Jud Amal Mukhtar
NIM : 2041720168
KELAS : 2C / TI
MATERI : Inheritance

Percobaan 1

Pertanyaan :

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

Jawab :

```
2 public class ClassB extends ClassA{
3     public int z;
4     public void getNilaiZ(){
5         System.out.println("nilai Z : " + z);
6     }
7     public void getJumlah(){
8         System.out.println("jumlah : "+(x+y+z));
9     }
10 }
11
```

Run:

```
nilai x : 20
nilai y : 30
nilai Z : 5
jumlah : 55
BUILD SUCCESSFUL (total time: 1 second)
```

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

Jawab :

tidak ada kode extends yang digunakan agar classB dapat memakai atribut classA secara langsung

Percobaan 2

Pertanyaan :

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

Jawab :

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

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

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

Jawab :

atribut x dan y bersifat private, yaitu hanya bisa diakses di dalam class-nya sendiri,



NAMA : Jud Amal Mukhtar
NIM : 2041720168
KELAS : 2C / TI
MATERI : Inheritance

oleh karena itu diperlukan getter untuk memakainya

Percobaan 3

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 :

Yaitu, untuk mengeset atribut r dan phi dari superclass-nya

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 :

untuk mengalikan atribut r dan phi dari superclass dengan atribut t dari class ini yang merupakan subclass

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

Jawab :

karena class Tabung merupakan subclass dari class Bangun dan dalam class Tabung atribut phi dan r sudah diakses dengan menggunakan super

Percobaan 4

Pertanyaan :

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

Jawab :

superclass = ClassA dan ClassB

subclass = ClassB dan ClassC

karena,

ClassA merupakan superclass dari ClassB dan ClassC,

ClassB merupakan superclass dari ClassC,

ClassB merupakan subclass dari ClassA, dan

ClassC merupakan subclass dari ClassB dan ClassA



NAMA : Jud Amal Mukhtar
NIM : 2041720168
KELAS : 2C / TI
MATERI : Inheritance

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 :

```
2 public class ClassC extends ClassB{  
3     ClassC(){  
4         super();  
5         System.out.println("konstruktor C dijalankan");  
6     }  
7 }
```

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!

Jawab :

konstruktor ClassA dijalankan pertama, kemudian ClassB, dan terakhir ClassC, dengan konstruktor ClassC memanggil konstruktor ClassB terlebih dahulu kemudian konstruktor ClassB memanggil konstruktor ClassA terlebih dahulu.

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

Jawab :

menjalankan atau memanggil konstruktor superclass-nya



NAMA : Jud Amal Mukhtar
NIM : 2041720168
KELAS : 2C / TI
MATERI : Inheritance

Tugas

Class Pegawai

```
1 package gajiPegawai;  
2 public class Pegawai10 {  
3     private String nip,nama,alamat;  
4  
5     public Pegawai10(String nip, String nama, String alamat) {  
6         this.nip = nip;  
7         this.nama = nama;  
8         this.alamat = alamat;  
9     }  
10  
11     public String getNama() {  
12         return nama;  
13     }  
14  
15     public int getGaji(){  
16         int gaji=nama.charAt(0)-'0';  
17         return gaji*100000;  
18     }  
19 }
```

Class Dosen

```
1 package gajiPegawai;  
2 public class Dosen10 extends Pegawai10{  
3     private int jumlahSKS,TARIF_SKS=500000;  
4     public Dosen10(String nip, String nama, String alamat) {  
5         super(nip, nama, alamat);  
6     }  
7  
8     public void setSKS(int jumlahSKS) {  
9         this.jumlahSKS = jumlahSKS;  
10    }  
11    @Override  
12    public int getGaji() {  
13        int gaji=jumlahSKS*TARIF_SKS;  
14        return gaji;  
15    }  
16 }
```



NAMA : Jud Amal Mukhtar
NIM : 2041720168
KELAS : 2C / TI
MATERI : Inheritance

Class DaftarGaji

```
1 package gajiPegawai;  
2 public class DaftarGaji10 {  
3     private Pegawai10 pegawai[];  
4  
5     public DaftarGaji10(int pegawai) {  
6         this.pegawai=new Pegawai10[pegawai];  
7         this.initPegawai();  
8     }  
9     private void initPegawai(){  
10        for(int i=0;i<this.pegawai.length;i++){  
11            this.pegawai[i]=new Pegawai10(null,null,null);  
12        }  
13    }  
14  
15    public void addPegawai(Pegawai10 pegawai) {  
16        for(int i=0;i<this.pegawai.length;i++){  
17            if(this.pegawai[i].getNama()==null){  
18                this.pegawai[i]=pegawai;  
19                break;  
20            }  
21        }  
22    }  
23  
24    public void printSemuaGaji(){  
25        for (Pegawai10 pegawai1 : this.pegawai) {  
26            if (pegawai1.getNama() != null) {  
27                System.out.println("Nama Pegawai : " + pegawai1.getNama());  
28                System.out.println("Gaji : " + pegawai1.getGaji() + "\n");  
29            }  
30        }  
31    }  
32 }
```

Class Main

```
1 package gajiPegawai;  
2 public class Main10 {  
3     public static void main(String[] args) {  
4         DaftarGaji10 gaji=new DaftarGaji10(10);  
5         Pegawai10 pgw1,pgw2;  
6         pgw1=new Pegawai10("30143","Jud","Jl. Merpati");  
7         pgw2=new Pegawai10("30250","Amal","Jl. Merak");  
8         Dosen10 dsn=new Dosen10("30233","Mukhtar","Jl. Pinguin");  
9         dsn.setSKS(7);  
10        gaji.addPegawai(pgw1);  
11        gaji.addPegawai(pgw2);  
12        gaji.addPegawai(dsn);  
13        gaji.printSemuaGaji();  
14    }  
15 }
```

Output

```
run: D:\K  
Nama Pegawai : Jud  
Gaji : 2600000  
  
Nama Pegawai : Amal  
Gaji : 1700000  
  
Nama Pegawai : Mukhtar  
Gaji : 3500000  
  
BUILD SUCCESSFUL (total time: 2 seconds)
```