



OBJEK ORIENTED PROGRAM

JOBSHEET 6



AYU JOVITA WIDYADHARI

2241720219/04/21

1ST Practicum

Practicum

```
5 package com.mycompany.jobsheet6;
6 public class ClassA {
7     public int x;
8     public int y;
9
10    public void getValue() {
11        System.out.println("Value x: " + x);
12        System.out.println("Value y: " + y);
13    }
14 }
15
```

```
5 package com.mycompany.jobsheet6;
6 public class ClassB {
7     public int z;
8
9     public void getValueZ() {
10        System.out.println("nilai Z: " + z);
11    }
12    public void getTotal() {
13        System.out.println("Jumlah: " + (x + y + z));
14    }
15 }
```

```
11 public class Percobaan1 {
12     public static void main(String[] args) {
13         ClassB hitung = new ClassB();
14         hitung.x=20;
15         hitung.y=30;
16         hitung.z=5;
17         hitung.getValue();
18         hitung.getValueZ();
19         hitung.getTotal();
20     }
21 }
```

Questions

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

Jawab:

```

5 package com.mycompany.jobsheet6;
6 public class ClassB extends ClassA{
7     public int z;
8
9     public void getValueZ(){
10         System.out.println("nilai Z: " + z);
11     }
12     public void getTotal(){
13         System.out.println("Jumlah: " + (x + y + z));
14     }
15 }

```

```

--- exec:3.1.0:exec (default-
Value x: 20
Value y: 30
nilai Z: 5
Jumlah: 55
-----
BUILD SUCCESS

```

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

Jawab:

Pada percobaan1 terjadi error karena classA tidak terhubung dengan classB begitu juga pada class percobaan1 meskipun berada di satu proyek yang sama. Hal tersebut dikarenakan tidak ada variable atau method dalam classB yang menggunakan atau mengacu pada anggota classA.

Jadi solusinya jika kita ingin mengakses classB ke class A kita dapat menambahkan referensi dari classB ke objek classA atau bisa juga menggunakan pewarisan (inheritance).

2nd Practicum

Practicum

```

public class ClassA {
7     //percobaan 2
8     private int x;
9     private int y;
10
11     public void setX(int x){
12         this.x = x;
13     }
14     public void setY(int y){
15         this.y = y;
16     }
17     public void getValue(){
18         System.out.println("nilai x: " + x);
19         System.out.println("nilai y: " + y);
20     }
21 }

```

```

6 public class ClassB extends ClassA{
7     //percobaan 2
8     private int z;
9
10    public void setZ(int z){
11        this.z = z;
12    }
13    public void getValueZ(){
14        System.out.println("nilai z: " + z);
15    }
16    public void getTotal(){
17        System.out.println("Jumlah: " + (x + y + z));
18    }
19 }
20

```

```

11 public class Percobaan2 {
12     public static void main(String[] args) {
13         ClassB hitung = new ClassB();
14         hitung.setX=20;
15         hitung.setY=30;
16         hitung.setZ=5;
17         hitung.getValue();
18         hitung.getValueZ();
19         hitung.getTotal();
20     }
21 }
22

```

Questions

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

```

5 package com.mycompany.jobsheet6;
6 public class ClassA {
7     private int x;
8     private int y;
9
10    public void setX(int x) {
11        this.x = x;
12    }
13
14    public void setY(int y) {
15        this.y = y;
16    }
17
18    public int getX() {
19        return x;
20    }
21
22    public int getY() {
23        return y;
24    }
25 }

```

```

1  package com.mycompany.jobsheet6;
2  import com.mycompany.jobsheet6.ClassA;
3
4  public class ClassB extends ClassA {
5      private int z;
6
7      public void setZ(int z) {
8          this.z = z;
9      }
10
11     public int getZ() {
12         return z;
13     }
14
15     public void getTotal() {
16         // Memanggil metode getValue() dari ClassA untuk mengakses variabel x dan y
17         System.out.println("nilai x: " + super.getX());
18         System.out.println("nilai y: " + super.getY());
19         System.out.println("nilai z: " + z);
20         System.out.println("Jumlah: " + (super.getX() + super.getY() + z));
21     }
22 }
23

```

```

1  package com.mycompany.jobsheet6;
2
3  public class Percobaan2 {
4      public static void main(String[] args) {
5          ClassB hitung = new ClassB();
6          hitung.setX(x: 20);
7          hitung.setY(y: 30);
8          hitung.setZ(z: 5);
9          hitung.getTotal();
10     }
11 }
12
13

```

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

.....

.....

.....

3rd Practicum

Practicum

```

12     protected double phi;
13     protected int r;
14 }
15

```

```

10
11 public class Tabung extends Bangun{
12     protected int t;
13     public void setSuperPhi(double phi){
14         super.phi=phi;
15     }
16     public void setSuperR(int r){
17         super.r = r;
18     }
19     public void setT(int t){
20         this.t = t;
21     }
22     public void volume(){
23         System.out.println("Volume Tabung adalah: " +(super.phi*super.r*super.r*this.t));
24     }
25 }
26

```

```

11 public class Percobaan3 {
12     public static void main(String[] args){
13         Tabung tabung = new Tabung();
14         tabung.setSuperPhi(phi:3.14);
15         tabung.setSuperR(r: 10);
16         tabung.setT(t: 3);
17         tabung.volume();
18     }
19 }
20

```

```

Volume Tabung adalah: 942.0
-----
BUILD SUCCESS

```

Questions

1. Jelaskan fungsi “super” pada potongan program berikut di class Tabung!

Jawab:

Pada class terdapat “super” yang digunakan untuk merujuk kepada variabel seperti phi dan r yang ada di class Bangunan dari kelas Tabung.

2. Jelaskan fungsi “super” dan “this” pada potongan program berikut di class Tabung!

Jawab:

Kata kunci khusus seperti “super” dan “this” digunakan untuk merujuk kepada objek yang berbeda dalam konteks yang berbeda

Class Tabung, super digunakan untuk merujuk kepada variabel-variabel dari superclass (phi dan r dari Bangun), sedangkan this digunakan untuk merujuk kepada variabel t dari class Tabung itu sendiri.

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

Jawab:

Karena atribut phi dan r sudah di protected di class Bangun. Sehingga atribut tersebut tidak perlu dideklarasikan lagi di class Tabung.

4th Practicum

Practicum

```
5 package com.mycompany.jobsheet6d;
6 public class ClassA {
7     ClassA() {
8         System.out.println(x: "konstruktor A dijalankan");
9     }
10 }
11
```

```
5 package com.mycompany.jobsheet6d;
6 public class ClassB extends ClassA {
7     ClassB() {
8         System.out.println(x: "konstruktor B dijalankan");
9     }
10 }
11
```

```
5 package com.mycompany.jobsheet6d;
6 public class ClassC extends ClassB {
7     ClassC() {
8         System.out.println(x: "konstruktor C diajalankan");
9     }
10 }
11
```

```
5 package com.mycompany.jobsheet6d;
6 public class Percobaan4 {
7     public static void main(String[] args) {
8         ClassC test = new ClassC();
9     }
10 }
11
```

```
konstruktor A dijalankan
konstruktor B dijalankan
konstruktor C diajalankan
```

```
-----
BUILD SUCCESS
-----
```

Question

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

Jawab:

- **Superclass:** ClassA merupakan superclass dari ClassB.
- **Subclass:** ClassB merupakan subclass dari ClassA, begitu juga ClassC merupakan subclass dari ClassB, dan ClassC merupakan subclass dari kedua ClassA dan ClassB.
- **Alasan:**

- a. Sebuah kelas yang di-extend oleh kelas lain menjadi superclass dari kelas yang meng-extend-nya. Maka dari itu ClassA extend ClassB, yang berarti ClassA merupakan superclass dari ClassB.
 - b. Begitu juga pada ClassB adalah superclass dari ClassC, karena ClassC meng-extend ClassB. Secara tidak langsung Class A merupakan superclass dari ClassC.
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:

```
5 package com.mycompany.jobsheet6d;  
6 public class ClassC extends ClassB {  
7     ClassC() {  
8         super();  
9         System.out.println("konstruktor C dijalankan");  
10    }  
11 }  
12
```

Output:

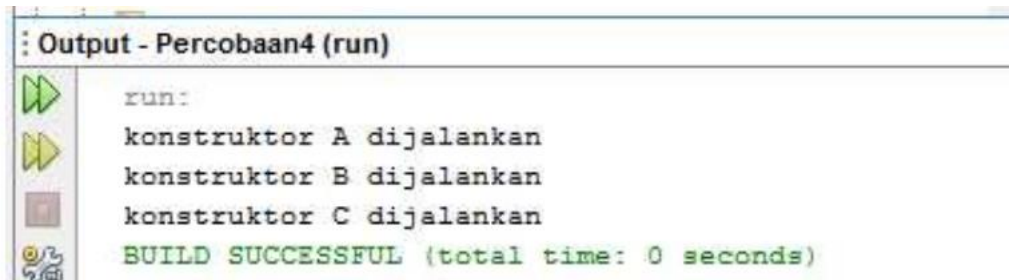
```
konstruktor A dijalankan  
konstruktor B dijalankan  
konstruktor C dijalankan  
-----  
BUILD SUCCESS  
-----
```

Bisa kita lihat pada output diatas, setelah penambahan konstruktor super() di ClassC dan setelah itu dijalankan programnya, hasil dari output tidak berubah dari sebelumnya. Namun, dengan penambahan konstruktor tersebut, kita bisa memastikan bahwa konstruktor default dari ClassB yang mana menjadi superclass dari ClassC, dijalankan secara eksplisit sebelum konstruktor ClassC dijalankan.

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



```
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!

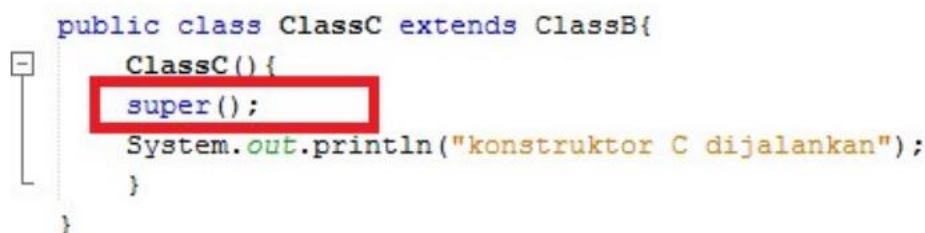
Jawab:

Urutan proses jalannya konstruktor:

- 1) Pertama-tama konstruktor dari ClassA dijalankan karena ClassC adalah subclass dari ClassB, dan ClassB adalah subclass dari ClassA.
- 2) Kemudian konstruktor dari ClassB dijalankan karena ClassC adalah subclass dari ClassB.
- 3) konstruktor dari ClassC dijalankan setelah konstruktor dari kedua superclassnya selesai dijalankan.

Perubahan pada posisi `super()` dalam konstruktor subclass sangat penting untuk memastikan urutan yang benar dalam inisialisasi objek dan mencegah kesalahan runtime atau hasil yang tidak diinginkan dalam program java.

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



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

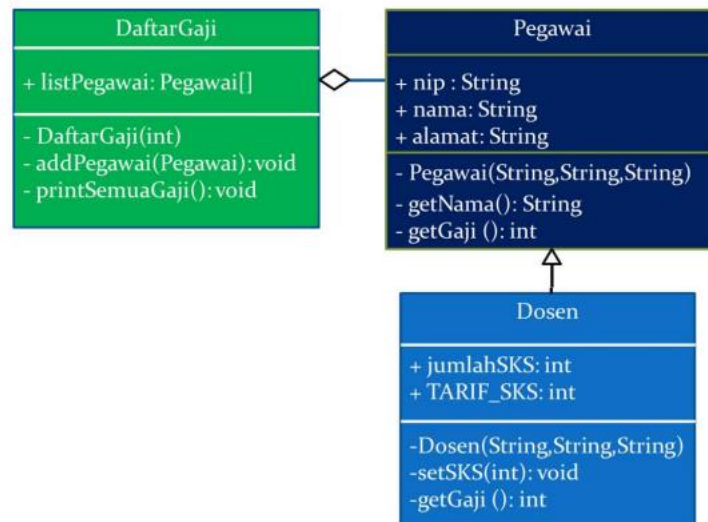
Jawab:

`Super()` pada ClassC digunakan untuk memanggil konstruktor dari superclass yakni di ClassB sebelum melakukan inisialisasi pada objek ClassC. Dengan menggunakan konstruktor tersebut, kita bisa memastikan bahwa inisialisasi pada superclass dilakukan dengan benar sebelum melakukan inisialisasi kelas ClassC.

Atau dengan kata lain konstrukto ini memastikan bahwa konstruktor dari superclass-nya dijalankan sebelum baris-baris kode dalam konstruktor subclass dieksekusi, dan memastikan subclass diinisialisasi dengan benar.

Assignment

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.



```
15 public class Pegawai {
16     private String nip;
17     private String nama;
18     private String alamat;
19
20     public Pegawai(String nip, String nama, String alamat){
21         this.nip = nip;
22         this.nama = nama;
23         this.alamat = alamat;
24     }
25     public String getNama(){
26         return nama;
27     }
28     public int getGaji(){
29         return 0;
30     }
31 }
```

```

11 public class DaftarGaji {
12     private Pegawai[] listPegawai;
13     private int jumlahPegawai;
14
15     public DaftarGaji(int maxPegawai) {
16         listPegawai = new Pegawai[maxPegawai];
17         jumlahPegawai = 0;
18     }
19
20     public void addPegawai(Pegawai pegawai) {
21         if (jumlahPegawai < listPegawai.length) {
22             listPegawai[jumlahPegawai] = pegawai;
23             jumlahPegawai++;
24         } else {
25             System.out.println(x: "Maaf, daftar pegawai sudah penuh.");
26         }
27     }
28
29     public void printSemuaGaji() {
30         System.out.println(x: "Daftar Gaji Pegawai:");
31         for (int i = 0; i < jumlahPegawai; i++) {
32             System.out.println("Nama: " + listPegawai[i].getNama());
33             System.out.println("Gaji: " + listPegawai[i].getGaji());
34             System.out.println(x: "-----");
35         }
36     }
37 }

```

```

11 public class Dosen extends Pegawai{
12     private int jumlah_SKS;
13     private int TARIF_SKS;
14
15     public Dosen(String nip, String nama, String alamat){
16         super(nip, nama, alamat);
17     }
18     public void setSKS(int jumlahSKS){
19         this.jumlah_SKS = jumlahSKS;
20     }
21     @Override
22     public int getGaji(){
23         return jumlah_SKS * TARIF_SKS;
24     }
25 }
26

```

```

11 public class Main {
12     public static void main(String[] args) {
13         DaftarGaji daftarGaji = new DaftarGaji(maxPegawai:3); // Maksimal 3 pegawai dalam daftar
14
15         Pegawai pegawai1 = new Pegawai(nip:"224", nama:"Ayu Jovita", alamat: "Jl. Kesumba No. 34");
16         Dosen dosen1 = new Dosen(nip:"172", nama:"WIdya Chandra", alamat: "Jl. Mt.Haryono No. 5");
17         dosen1.setSKS(jumlahSKS: 10); // Set jumlah SKS yang diajar oleh dosen1
18
19         Pegawai pegawai2 = new Pegawai(nip:"789", nama:"Gusti Ayu", alamat: "Jl. Sobontoro No.9");
20
21         // Menambahkan pegawai ke daftar gaji
22         daftarGaji.addPegawai(pegawai:pegawai1);
23         daftarGaji.addPegawai(pegawai:dosen1);
24         daftarGaji.addPegawai(pegawai:pegawai2);
25
26         // Menampilkan gaji semua pegawai
27         daftarGaji.printSemuaGaji();
28     }
29 }

```

Daftar Gaji Pegawai:

Nama: Ayu Jovita

Gaji: 0

Nama: WIdya Chandra

Gaji: 0

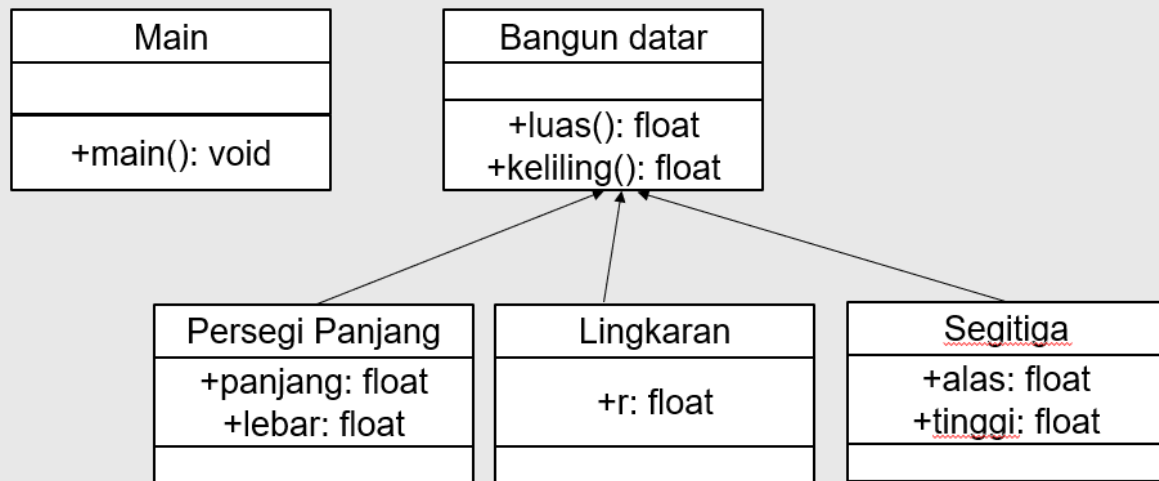
Nama: Gusti Ayu

Gaji: 0

BUILD SUCCESS

PPT Assignment

- 1. Tuliskan program yang berfungsi untuk menghitung luas dan keliling bangun datar.



```
13 public class BangunDatar {
14     public float luas() {
15         return 0;
16     }
17     public float keliling() {
18         return 0;
19     }
20 }
```

```

public class PersegiPanjang extends BangunDatar {
    private float panjang;
    private float lebar;

    public PersegiPanjang(float panjang, float lebar) {
        this.panjang = panjang;
        this.lebar = lebar;
    }

    @Override
    public float luas() {
        return panjang * lebar;
    }

    @Override
    public float keliling() {
        return 2 * (panjang + lebar);
    }
}

```

```

public class Lingkaran extends BangunDatar {
    private float r;

    public Lingkaran(float r) {
        this.r = r;
    }

    @Override
    public float luas() {
        return (float) (Math.PI * r * r);
    }

    @Override
    public float keliling() {
        return (float) (2 * Math.PI * r);
    }
}

```

```

public class Segitiga extends BangunDatar {
    private float alas;
    private float tinggi;

    public Segitiga(float alas, float tinggi) {
        this.alas = alas;
        this.tinggi = tinggi;
    }

    @Override
    public float luas() {
        return 0.5f * alas * tinggi;
    }

    @Override
    public float keliling() {
        // anggap segitiga sebagai segitiga sembarang
        return -1;
    }
}

```

```

public class Main {
    public static void main(String[] args) {
        PersegiPanjang persegiPanjang = new PersegiPanjang(panjang:5, lebar:3);
        Lingkaran lingkaran = new Lingkaran(r:4);
        Segitiga segitiga = new Segitiga(alas:6, tinggi:8);

        System.out.println("Luas Persegi Panjang: " + persegiPanjang.luas());
        System.out.println("Keliling Persegi Panjang: " + persegiPanjang.keliling());

        System.out.println("Luas Lingkaran: " + lingkaran.luas());
        System.out.println("Keliling Lingkaran: " + lingkaran.keliling());

        System.out.println("Luas Segitiga: " + segitiga.luas());
        System.out.println("Keliling Segitiga: " + segitiga.keliling());
    }
}

```

Output - tugasPpt (run)

```

run:
Luas Persegi Panjang: 15.0
Keliling Persegi Panjang: 16.0
Luas Lingkaran: 50.265484
Keliling Lingkaran: 25.132742
Luas Segitiga: 24.0
Keliling Segitiga: -1.0
BUILD SUCCESSFUL (total time: 0 seconds)

```