

Jobsheet 6 - Inheritance
Object Oriented Programming



Arranged by :
Shofwah Kanaka Ebsa Anargya
2241720254 / 22
2I

INFORMATION TECHNOLOGY
D-IV INFORMATICS ENGINEERING
MALANG STATE POLYTECHNIC
2023

PERCOBAAN 1 (extends)

A. TAHAPAN PERCOBAAN

1. Buatlah sebuah class parent/superclass dengan nama ClassA.java

```
package js6.Percobaan1;

public class ClassA {
    public int x;
    public int y;
    public void getNilai () {
        System.out.println("nilai x:" + x);
        System.out.println("nilai y:" + y);
    }
}
```

2. Buatlah sebuah class anak/subclass dengan nama ClassB.java

```
package js6.Percobaan1;

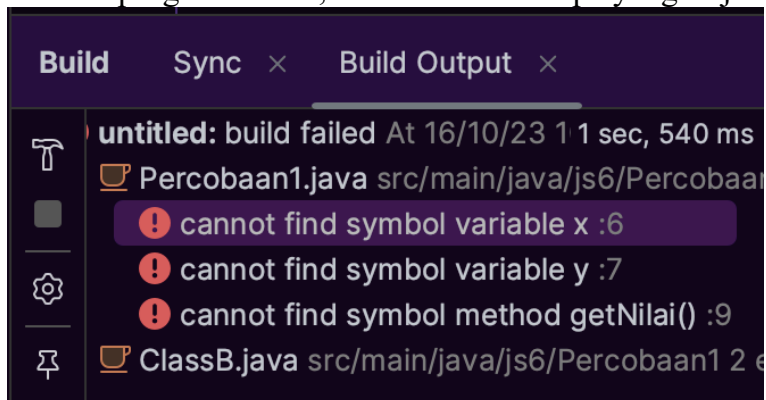
public class ClassB {
    public int z;
    public void getNilaiZ () {
        System.out.println("nilai Z:" + z);
    }
    public void getJumlah () {
        System.out.println("jumlah: " + (x+y+z));
    }
}
```

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

```
package js6.Percobaan1;

public class Percobaan1 {
    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();
    }
}
```

5. Jalankan program diatas, kemudian amati apa yang terjadi!



a. PERTANYAAN

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

```
public class ClassB extends ClassA{
    public int z;
    public void getNilaiZ () {
        System.out.println("nilai Z:"+ z);
    }
    public void getJumlah () {
        System.out.println("jumlah: " + (x+y+z));
    }
}
```

Run Percobaan1 x

/Library/Java/JavaVirtualMachines/jdk-19

↑ nilai x:20
↓ nilai y:30
⇌ nilai Z:5
⇓ jumlah: 55

Process finished with exit code 0

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

ClassB tidak memiliki akses ke variabel x dan y yang ada diClassA`, kecuali jika hubungan turunan antara keduanya telah ditetapkan dengan benar.

PERCOBAAN 2 (Hak Akses)

A. TAHAPAN PERCOBAAN

1. Buatlah sebuah class parent/superclass dengan nama ClassA.java

```
package js6.Percobaan2;

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 void getNilai () {
        System.out.println("nilai x: " + x);
        System.out.println("nilai y:" + y);
    }
}
```

2. Buatlah sebuah class anak/subclass dengan nama ClassB.java

```
package js6.Percobaan2;

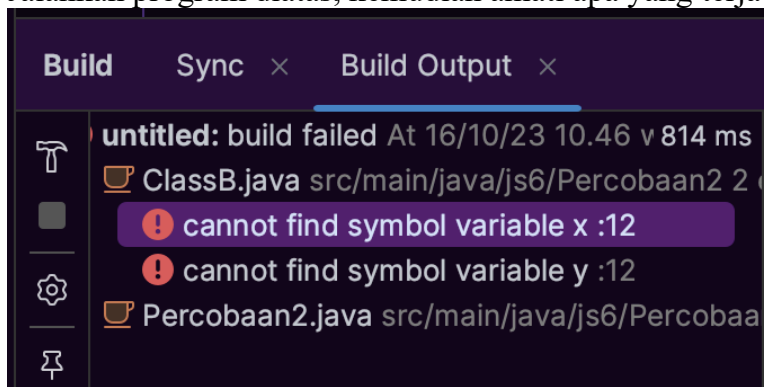
public class ClassB {
    private int z;
    public void setZ (int z) {
        this.z = z;
    }
    public void getNilaiZ () {
        System.out.println("nilai Z:"+ z);
    }
    public void getJumlah () {
        System.out.println("jumlah: " + (x + y + z));
    }
}
```

3. Buatlah class Percobaan2.java untuk menjalankan program diatas!

```
package js6.Percobaan2;

public class Percobaan2 {
    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!

```
public class ClassB extends ClassA{  
    public class ClassA {  
        2 usages  
        public int x;  
        2 usages  
        public int y;  
    }  
}
```

↑
↓
⇌
⇓

```
/Library/Java/JavaVirtualMachines/j  
nilai x: 20  
nilai y:30  
nilai Z:5  
jumlah: 55
```

2. Jelaskan apa penyebab program pada percobaan 1 ketika dijalankan terdapat error! Karena belum ada extends untuk kata kunci yang digunakan dalam pemrograman berorientasi objek untuk menggambarkan hubungan inheritance antara kelas. Dan variable x dan y dalam ClassA bersifat private yang seharusnya Public / Protected

PERCOBAAN 3 (Super)

A. TAHAPAN PERCOBAAN

1. Buatlah sebuah class parent/superclass dengan nama Bangun.java

```
package js6.Percobaan3;

public class Bangun {
    protected double phi;
    protected int r;
}
```

2. Buatlah sebuah class anak/subclass dengan nama Tabung.java

```
package js6.Percobaan3;

public class Tabung extends Bangun{
    protected int t;
    public void setSuperPhi (double phi) {
        super.phi = phi;
    }
    public void setSuperR (int r) {
        super.r = r;
    }
    public void setT (int t) {
        this.t = t;
    }
    public void volume () {
        System.out.println("Volume Tabung adalah: " + (super.phi *
        super.r * super.r * this.t));
    }
}
```

3. Buatlah class Percobaan3.java untuk menjalankan program diatas!

```
package js6.Percobaan3;

public class Percobaan3 {
    public static void main(String[] args) {
        Tabung tabung=new Tabung ();
        tabung.setSuperPhi (3.14);
        tabung.setSuperR (10);
        tabung.setT (3);
        tabung.volume();
    }
}
```

4. Jalankan program diatas!

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/
Volume Tabung adalah: 942.0

Process finished with exit code 0
```

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;
}
```

Super digunakan dalam class Tabung untuk mengakses atribut phi dan r yang merupakan atribut dari superclass Bangun. Kedua method tersebut digunakan untuk menginisialisasi atribut phi dan r dari superclass Bangun dengan nilai yang diberikan pada method tersebut.

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));  
}
```

super.phi merujuk ke atribut phi dalam superclass, super.r merujuk ke atribut r dalam superclass, dan this.t merujuk ke atribut t dalam subclass.

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

Pada class Tabung, atribut phi dan r sebenarnya dideklarasikan sebagai protected pada class parent/superclass Bangun.

Atribut phi dan r adalah bagian dari superclass Bangun yang menjadi superclass dari class Tabung.

Penggunaan kata kunci protected memungkinkan atribut tersebut diakses dari subclass (dalam hal ini, class Tabung) tanpa harus dideklarasikan kembali.

Penggunaan akses modifier protected memungkinkan implementasi dari konsep warisan (inheritance) dalam pemrograman berorientasi objek, di mana subclass dapat mewarisi atribut dan metode dari superclass-nya. Itulah mengapa atribut phi dan r dapat diakses oleh class Tabung.

PERCOBAAN 4 (super constructor)

A. TAHAPAN PERCOBAAN

1. Buatlah tiga file dengan nama ClassA.java , ClassB.java , dan ClassC.java, seperti pada kodeprogram dibawah ini!

ClassA.java

```
package js6.Percobaan4;

public class ClassA {
    ClassA() {
        System.out.printf("konstruktor A dijalankan\n");
    }
}
```

ClassB.java

```
package js6.Percobaan4;

public class ClassB extends ClassA{
    ClassB(){
        System.out.printf("konstruktor B dijalankan\n");
    }
}
```

ClassC.java

```
package js6.Percobaan4;

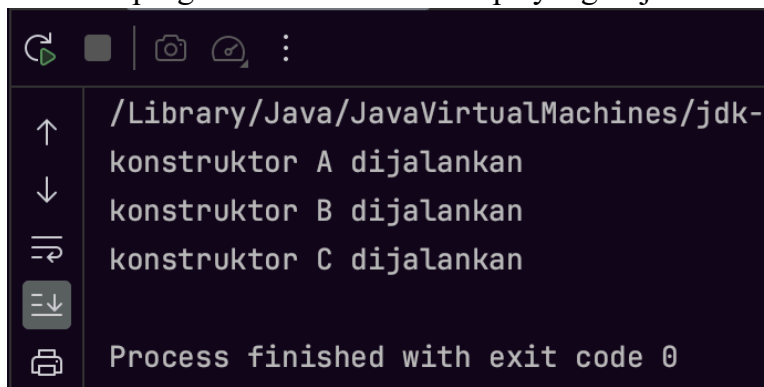
public class ClassC extends ClassB{
    ClassC(){
        System.out.printf("konstruktor C dijalankan\n");
    }
}
```

2. Buatlah class Percobaan4.java untuk menjalankan program diatas!

```
package js6.Percobaan4;

public class Percobaan4 {
    public static void main(String[] args) {
        ClassC test = new ClassC();
    }
}
```

3. Jalankan program kemudian amati apa yang terjadi!



```
↑ /Library/Java/JavaVirtualMachines/jdk-
↓ konstruktor A dijalankan
⇌ konstruktor B dijalankan
⇌ konstruktor C dijalankan
⇓
📄 Process finished with exit code 0
```

B. PERTANYAAN

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

ClassA: Merupakan superclass dalam hierarki class. ClassA adalah class paling atas atau class induk (parent class) yang tidak memiliki superclass lainnya. Tidak ada class lain yang menjadi turunan langsung dari ClassA.

ClassB: Merupakan subclass dari ClassA. ClassB mewarisi atribut dan metode yang dimiliki oleh ClassA. ClassB adalah turunan langsung dari ClassA, sehingga ClassA adalah superclass dari ClassB.

ClassC: Merupakan subclass dari ClassB. ClassC mewarisi atribut dan metode dari ClassA dan ClassB. ClassB adalah superclass dari ClassC, dan secara tidak langsung, ClassA juga merupakan superclass dari ClassC melalui ClassB.

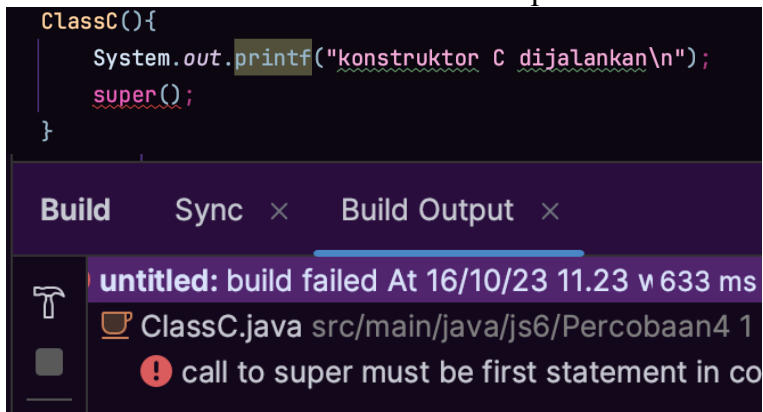
2. Ubahlah isi konstruktor default ClassC seperti berikut:

```
ClassC(){  
    super();  
    System.out.printf("konstruktor C dijalankan\n");  
}
```

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

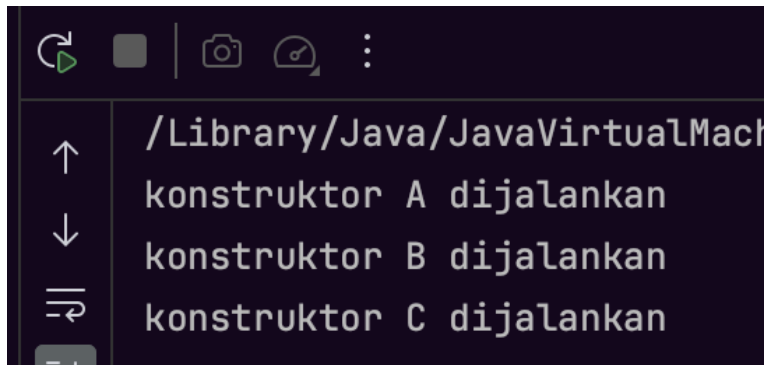
Tambahnya `super()` pada konstruktor ClassC tidak mengubah hasil program ini, karena ClassB dan ClassA tidak menerima argumen di konstruktor defaultnya. Jadi, memanggil `super()` dengan atau tanpa argumen akan mengarah ke konstruktor default superclass.

3. Ublah isi konstruktor default ClassC seperti berikut:



Ketika mengubah posisi `super()` di baris kedua dalam konstruktor defaultnya dan terlihat ada error. Kemudian kembalikan `super()` ke baris 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



```
/Library/Java/JavaVirtualMach
konstruktor A dijalankan
konstruktor B dijalankan
konstruktor C dijalankan
```

Jelaskan bagaimana urutan proses jalannya konstruktor saat objek test dibuat!

Ketika objek test dari class ClassC dibuat, urutan proses jalannya konstruktor adalah sebagai berikut:

Konstruktor ClassC dijalankan.

Konstruktor ClassB dijalankan dengan memanggil super().

Konstruktor ClassA dijalankan dengan memanggil super() dari konstruktor ClassB.

Mengapa bisa tampilan output seperti itu? Karena saat konstruktor ClassC dijalankan, konstruktor ClassB dipanggil terlebih dahulu melalui super(), yang kemudian memanggil konstruktor ClassA dengan super(). Oleh karena itu, outputnya adalah:

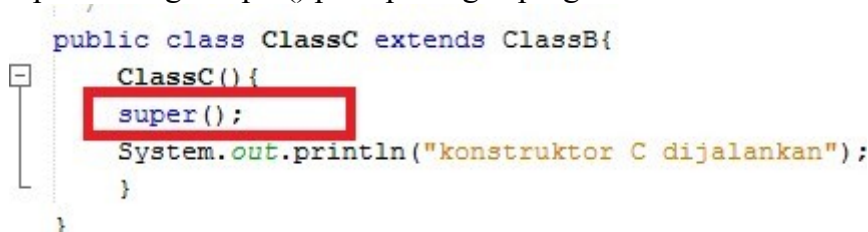
konstruktor A dijalankan

konstruktor B dijalankan

konstruktor C dijalankan

Jadi, konstruktor ClassA dijalankan sebelum konstruktor ClassB dan ClassC karena urutan pemanggilan super() dalam konstruktor ClassC adalah super() pada baris pertama.

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



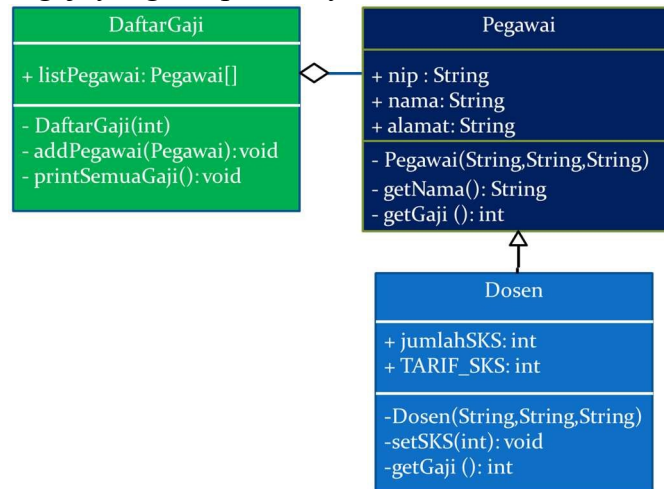
```
public class ClassC extends ClassB{
    ClassC(){
        super();
        System.out.println("konstruktor C dijalankan");
    }
}
```

Fungsi dari super() di sini adalah memanggil konstruktor superclass dari ClassC, yaitu konstruktor ClassB. Dengan kata lain, super() digunakan untuk memastikan bahwa konstruktor ClassB (superclass) dijalankan sebelum konstruktor ClassC (subclass).

Jadi, dalam urutan eksekusi, pertama-tama konstruktor ClassB dijalankan karena super() dipanggil, dan kemudian konstruktor ClassC dijalankan. Ini memastikan bahwa inisialisasi yang benar terjadi dalam hierarki pewarisan kelas.

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



```
package js6.Tugas;

public class DaftarGaji {
    private Pegawai[] listPegawai;
    private int jumlahPegawai;

    public DaftarGaji(int maxJumlahPegawai) {
        listPegawai = new Pegawai[maxJumlahPegawai];
        jumlahPegawai = 0;
    }

    public void addPegawai(Pegawai pegawai) {
        if (jumlahPegawai < listPegawai.length) {
            listPegawai[jumlahPegawai] = pegawai;
            jumlahPegawai++;
        } else {
            System.out.println("Maksimum jumlah pegawai tercapai.");
        }
    }

    public void printSemuaGaji() {
        for (int i = 0; i < jumlahPegawai; i++) {
            System.out.println("Nama Pegawai: " +
                listPegawai[i].getName());
            System.out.println("Gaji Pegawai: " +
                listPegawai[i].getGaji());
            System.out.println();
        }
    }
}
```

```
package js6.Tugas;

public 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 int getGaji() {
        return 0;
    }
}
```

```

package js6.Tugas;

public class Dosen extends Pegawai {
    private int jumlahSKS;
    private static final int TARIF_SKS = 100000;

    public Dosen(String nip, String nama, String alamat) {
        super(nip, nama, alamat);
        jumlahSKS = 0;
    }

    public void setSKS(int jumlahSKS) {
        this.jumlahSKS = jumlahSKS;
    }

    public int getGaji() {
        return jumlahSKS * TARIF_SKS;
    }
}

```

```

package js6.Tugas;

public class Main {
    public static void main(String[] args) {
        DaftarGaji daftarGaji = new DaftarGaji(10);

        Pegawai pegawai1 = new Pegawai("12345", "Shofwah", "Malang");
        Dosen dosen1 = new Dosen("67890", "Kanaka", "Ngawi");

        dosen1.setSKS(10);

        daftarGaji.addPegawai(pegawai1);
        daftarGaji.addPegawai(dosen1);

        daftarGaji.printSemuaGaji();
    }
}

```

Run

Main

↑

↓

↔

⇅

🖨

🗑

/Library/Java/JavaVirtualMachines/jdk-19.

Nama Pegawai: Shofwah

Gaji Pegawai: 0

Nama Pegawai: Kanaka

Gaji Pegawai: 1000000

