

Object Oriented Programming

Jobsheet 6

inheritance



Name: Azahra Salsabila

NIM: 2241720073

Class: 2I

No: 05

Major: Information Technology

Study Program: Informatic Engineering

Experiment 1

Screenshot program code

```
1 package semester3.jobsheet6.praktikum1;
2 public class Percobaan1 {
    Run | Debug
3     public static void main(String[] args) {
4         ClassB hitung = new ClassB();
5         hitung.x = 20;
6         hitung.y = 30;
7         hitung.z = 5;
8         hitung.getNilai();
9         hitung.getNilaiZ();
10        hitung.getJumlah();
11    }
12 }
```

Screenshot run

```
PS D:\terserah\terserah> & 'C:\Program Files\Java\jdk1.8.0_121\bin\java.exe' '-
cp' 'D:\terserah\terserah\bin' 'semester3.jobsheet6.praktikum1.Percobaan1'
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
    x cannot be resolved or is not a field
    y cannot be resolved or is not a field
    The method getNilai() is undefined for the type ClassB

    at semester3.jobsheet6.praktikum1.Percobaan1.main(Percobaan1.java:5)
```

Questions

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

```
nilai x: 20
nilai y: 30
nilai z: 5
jumlah: 55
```

- menambahkan extends saat deklarasi class B.

```
public class ClassB extends ClassA {
```

2. Jelaskan apa penyebab program pada percobaan 1 ketika dijalankan terdapat error!
 - Pada percobaan 1, program mengalami error karena pada class ClassB mencoba menggunakan method getNilai() yang sebenarnya ada di class ClassA, tetapi class ClassB tidak memiliki akses ke method tersebut karena tidak ada hubungan pewarisan (inheritance) antara kedua class tersebut. Oleh karena itu, program menghasilkan kesalahan "The method getNilai() is undefined for the type ClassB".

Experiment 2

Screenshot program code

```

1  package semester3.jobsheet6.praktikum2;
2  public class ClassA {
3      private int x;
4      private int y;
5
6      public void setX(int x) {
7          this.x = x;
8      }
9
10     public void setY(int y) {
11         this.y = y;
12     }
13
14     public void getNilai() {
15         System.out.println("nilai x: " + x);
16         System.out.println("nilai y: " + y);
17     }
18 }

```

```

1  package semester3.jobsheet6.praktikum2;
2  public class ClassB {
3      private int z;
4      public void setZ(int z) {
5          this.z = z;
6      }
7
8      public void getNilaiZ() {
9          System.out.println("nilai Z:" + z);
10     }
11
12     public void getJumlah() {
13         System.out.println("jumlah:" + (x + y + z));
14     }
15 }

```

```

1  package semester3.jobsheet6.praktikum2;
2  public class Percobaan2 {
3      Run | Debug
4      public static void main(String[] args) {
5          ClassB hitung = new ClassB();
6          hitung.x = 20;
7          hitung.y = 30;
8          hitung.z = 5;
9          hitung.getNilai();
10         hitung.getNilaiZ();
11         hitung.getJumlah();
12     }

```

Screenshot run

```
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
  x cannot be resolved or is not a field
  y cannot be resolved or is not a field
  The field ClassB.z is not visible
  The method getNilai() is undefined for the type ClassB

  at semester3.jobsheet6.praktikum2.Percobaan2.main(Percobaan2.java:5)
```

Questions

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

```
➤ nilai x: 20
   nilai y: 30
   nilai z: 5
   jumlah: 55
```

Menambahkan 'extends' saat deklarasi Class B agar variable pada Class A dapat diakses. Dan mengubah akses modifier variabel x dan y dalam class ClassA menjadi protected agar bisa diakses oleh class turunan seperti ClassB

```
protected int x;
protected int y;
```

2. Jelaskan apa penyebab program pada percobaan 1 ketika dijalankan terdapat error!
 - Pada percobaan 2, program mengalami error karena class ClassB mencoba mengakses variabel x dan y, yang sebenarnya terdapat dalam class ClassA, tetapi class ClassB tidak memiliki akses ke variabel tersebut karena mereka adalah private (hanya dapat diakses di dalam class yang sama). Oleh karena itu, program menghasilkan kesalahan "The field x is not visible."

Experiment 3

Screenshot program code

```
1 package semester3.jobsheet6.praktikum3;
2 public class Bangun {
3     protected double phi;
4     protected int r;
5 }
```

```

1 package semester3.jobsheet6.praktikum3;
2 public class Tabung extends Bangun {
3     protected int t;
4     public void setSuperPhi(double phi) {
5         super.phi = phi;
6     }
7     public void setSuperR (int r) {
8         super.r = r;
9     }
10    public void setT (int t) {
11        this.t = t;
12    }
13    public void volume() {
14        System.out.println("Volume Tabung adalah: " + (super.phi * super.r * super.r * this.t));
15    }
16 }

```

```

1 package semester3.jobsheet6.praktikum3;
2 public class Percobaan3 {
3     Run | Debug
4     public static void main(String[] args) {
5         //TODO code application logic here
6         Tabung tabung = new Tabung();
7         tabung.setSuperPhi(phi:3.14);
8         tabung.setSuperR(r:10);
9         tabung.setT(t:3);
10        tabung.volume();
11    }
12 }

```

Screenshot run

Volume Tabung adalah: 942.0

Questions

1. Jelaskan fungsi “super” pada potongan program berikut di class Tabung!
 - Fungsi "super" pada potongan program `super.phi = phi;` `super.r = r;` digunakan untuk mengakses atribut phi dan r dari class induk (superclass) yang dalam hal ini adalah class Bangun. Ini berguna karena atribut phi dan r dalam class Bangun memiliki akses modifier protected, yang berarti dapat diakses oleh class turunan seperti Tabung.
2. Jelaskan fungsi “super” dan “this” pada potongan program berikut di class Tabung!
 - Fungsi "super" digunakan dalam potongan program `super.phi * super.r * super.r` untuk mengakses atribut phi dan r dari class Bangun. Fungsi "this" dalam potongan program tersebut digunakan untuk mengakses atribut t yang ada dalam class Tabung.
3. Jelaskan mengapa pada class Tabung tidak dideklarasikan atribut “phi” dan “r” tetapi class tersebut dapat mengakses atribut tersebut!
 - Class Tabung tidak perlu mendeklarasikan atribut phi dan r lagi karena class Tabung adalah turunan dari class Bangun. Oleh karena itu, class Tabung sudah mewarisi atribut phi dan r dari class Bangun secara implisit. Ini adalah salah satu prinsip dalam konsep pewarisan (inheritance) di mana class turunan dapat

menggunakan atribut dan method dari class induk tanpa perlu mendefinisikannya ulang, asalkan atribut dan method tersebut memiliki akses modifier yang memungkinkan untuk diakses oleh class turunan.

Experiment 4

Screenshot program code

```
1 package semester3.jobsheet6.praktikum4;
2 public class ClassA {
3     ClassA() {
4         System.out.println(x:"Konstruktor A dijalankan");
5     }
6 }
```

```
1 package semester3.jobsheet6.praktikum4;
2 public class ClassB extends ClassA {
3     ClassB() {
4         System.out.println(x:"Konstruktor B dijalankan");
5     }
6 }
```

```
1 package semester3.jobsheet6.praktikum4;
2 public class ClassC extends ClassB {
3     ClassC() {
4         System.out.println(x:"Konstruktor C dijalankan");
5     }
6 }
```

```
1 package semester3.jobsheet6.praktikum4;
2 public class Percobaan4 {
3     Run | Debug
4     public static void main(String[] args) {
5         ClassC test = new ClassC();
6     }
7 }
```

Screenshot run

```
Konstruktor A dijalankan
Konstruktor B dijalankan
Konstruktor C dijalankan
```

Questions

1. Pada percobaan 4 sebutkan mana class yang termasuk superclass dan subclass, kemudian jelaskan alasannya!
 - Pada percobaan 4, class yang termasuk superclass dan subclass adalah:
 - a) ClassA adalah superclass, karena ClassA adalah kelas dasar yang tidak memiliki superclass.

- b) ClassB adalah subclass dari ClassA dan superclass untuk ClassC. ClassB merupakan subclass dari ClassA, karena ClassB memperluas (mewarisi) fitur-fitur dari ClassA.
 - c) ClassC adalah subclass dari ClassB. ClassC merupakan subclass dari ClassB, karena ClassC memperluas (mewarisi) fitur-fitur dari ClassB, yang pada gilirannya mewarisi fitur-fitur dari ClassA.
- 2. Ubahlah isi konstruktor default ClassC seperti berikut:
 Tambahkan kata `super()` di baris Pertama dalam konstruktor defaultnya. Coba jalankan kembali class Percobaan4 dan terlihat tidak ada perbedaan dari hasil outputnya!
 - Ketika `super()` ditambahkan di baris pertama dalam konstruktor default ClassC seperti yang ditunjukkan, sebenarnya konstruktor kelas superclass (ClassB) dipanggil sebelum menjalankan konstruktor kelas saat ini (ClassC). Ini adalah perilaku default jika konstruktor tidak eksplisit memanggil konstruktor kelas atasnya dengan `super()` atau `super(parameters)`. Dalam hal ini, karena konstruktor ClassB adalah konstruktor default, yang akan dipanggil oleh `super()`, maka hasilnya tidak berbeda dari output sebelumnya.
- 3. Ubahlah 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
 Jelaskan bagaimana urutan proses jalannya konstruktor saat objek test dibuat!
 - Ketika posisi `super()` diubah ke baris kedua dalam konstruktor default ClassC, terjadi kesalahan karena aturan bahasa Java. Konstruktor kelas harus selalu dipanggil pada baris pertama dalam konstruktor. Jika `super()` dipanggil setelah instruksi lain dalam konstruktor, maka Java akan menimbulkan kesalahan.
 - Ketika `super()` dipindahkan ke baris kedua, maka konstruktor ClassC akan memanggil konstruktor ClassB terlebih dahulu dan kemudian menjalankan instruksi di dalam konstruktor ClassC. Ini sebenarnya adalah perilaku yang diharapkan. Meskipun `super()` dipindahkan ke baris kedua, Java tetap memaksa untuk memanggil kelas superclass terlebih dahulu.
 - Ketika `super()` dipindahkan ke baris kedua, maka output yang ditampilkan ketika objek test dari ClassC dibuat akan mencetak "Konstruktor B dijalankan" terlebih dahulu, dan kemudian "Konstruktor C dijalankan."
- 4. Apakah fungsi `super()` pada potongan program dibawah ini di ClassC!
 - Fungsi `super()` dalam potongan program `super()`; di ClassC digunakan untuk memanggil konstruktor kelas superclass (ClassB) sebelum menjalankan konstruktor ClassC. Itu berarti, sebelum menjalankan instruksi dalam konstruktor ClassC, Java akan menjalankan konstruktor ClassB terlebih dahulu. Ini adalah cara untuk memastikan bahwa konstruktor kelas atas (superclass) dijalankan sebelum konstruktor kelas saat ini.

Assignment

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

```
Daftar Gaji Pegawai:  
Nama: Cello  
Gaji: 100000  
  
Nama: Helga  
Gaji: 200000  
  
Nama: Milan  
Gaji: 300000  
  
Nama: Hilmy  
Gaji: 400000
```



```
1  package semester3.jobsheet6.assignment;  
2  public class Pegawai {  
3      private String nip;  
4      private String nama;  
5      private String alamat;  
6  
7      public Pegawai(String nip, String nama, String alamat) {  
8          this.nip = nip;  
9          this.nama = nama;  
10         this.alamat = alamat;  
11     }  
12  
13     public String getNama() {  
14         return nama;  
15     }  
16  
17     public int getGaji() {  
18         return 0;  
19     }  
20 }
```



```

1  package semester3.jobsheet6.assignment;
2  public class Dosen extends Pegawai {
3      private int jumlahSKS;
4      private static final int TARIF_SKS = 50000;
5
6      public Dosen(String nip, String nama, String alamat) {
7          super(nip, nama, alamat);
8          this.jumlahSKS = 0;
9      }
10
11     public void setSKS(int jumlahSKS) {
12         this.jumlahSKS = jumlahSKS;
13     }
14
15     public int getGaji() {
16         return jumlahSKS * TARIF_SKS;
17     }
18 }

```

```

1  package semester3.jobsheet6.assignment;
2  public class DaftarGaji {
3      private Pegawai[] listPegawai;
4      private int numPegawai;
5
6      public DaftarGaji(int maxPegawai) {
7          listPegawai = new Pegawai[maxPegawai];
8          numPegawai = 0;
9      }
10
11     public void addPegawai(Pegawai pegawai) {
12         if (numPegawai < listPegawai.length) {
13             listPegawai[numPegawai] = pegawai;
14             numPegawai++;
15         } else {
16             System.out.println(x:"Daftar gaji sudah penuh");
17         }
18     }
19
20     public void printSemuaGaji() {
21         System.out.println(x:"Daftar Gaji Pegawai: ");
22         for (int i = 0; i < numPegawai; i++) {
23             System.out.println("Nama: " + listPegawai[i].getNama());
24             System.out.println("Gaji: " + listPegawai[i].getGaji());
25             System.out.println();
26         }
27     }
28 }

```

```
1 package semester3.jobsheet6.assignment;
2 public class Main {
3     Run | Debug
4     public static void main(String[] args) {
5         DaftarGaji daftarGaji = new DaftarGaji(maxPegawai:8);
6
7         Dosen dosen1 = new Dosen(nip:"112233", nama:"Cello", alamat:"jl. neverland no. 1");
8         dosen1.setSKS(jumlahSKS:2);
9
10        Dosen dosen2 = new Dosen(nip:"445566", nama:"Helga", alamat:"jl. neverland no. 2");
11        dosen2.setSKS(jumlahSKS:4);
12
13        Dosen dosen3 = new Dosen(nip:"778899", nama:"Milan", alamat:"jl. neverland no. 3");
14        dosen3.setSKS(jumlahSKS:6);
15
16        Dosen dosen4 = new Dosen(nip:"90123", nama:"Hilmy", alamat:"jl. neverland no. 4");
17        dosen4.setSKS(jumlahSKS:8);
18
19        daftarGaji.addPegawai(dosen1);
20        daftarGaji.addPegawai(dosen2);
21        daftarGaji.addPegawai(dosen3);
22        daftarGaji.addPegawai(dosen4);
23
24        daftarGaji.printSemuaGaji();
25    }
```