

JOBSHEET 6
OBJECT ORIENTED PROGRAMMING



Arranged By :

Lenka Melinda Florianka

2241720074

Class 2I

INFORMATION TECHNOLOGY
D-IV INFORMATICS ENGINEERING
MALANG STATE POLYTECHNIC
2023

Practicum 1

Code :

```
1 package josbheet5;
2
3 public class Percobaan1 {
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    }
13 }
14 }
15 }
```

```
1 package josbheet5;
2
3 public class ClassB {
4     public int z;
5
6     public void getNilaiZ(){
7         System.out.println("nilai z:" + z);
8     }
9
10    public void getJumlah(){
11        System.out.println("jumlah:" + (x+y+z));
12    }
13 }
14 }
```

```
1 package josbheet5;
2
3 public class ClassA {
4     public int x;
5     public int y;
6
7     public void getNilai(){
8         System.out.println("nilai x:" + x);
9         System.out.println("nilai x:" + y);
10    }
11 }
12 }
```

Output :

```

  ClassB.java josbheet5  2
    x cannot be resolved to a variable Java(33554515) [Ln 11, Col 41]
    y cannot be resolved to a variable Java(33554515) [Ln 11, Col 43]
  Percobaan1.java josbheet5  3
    x cannot be resolved or is not a field Java(33554502) [Ln 6, Col 16]
    y cannot be resolved or is not a field Java(33554502) [Ln 7, Col 16]
    The method getNilai() is undefined for the type ClassB Java(67108964) [Ln 9, Col 16]

```

Question

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

```

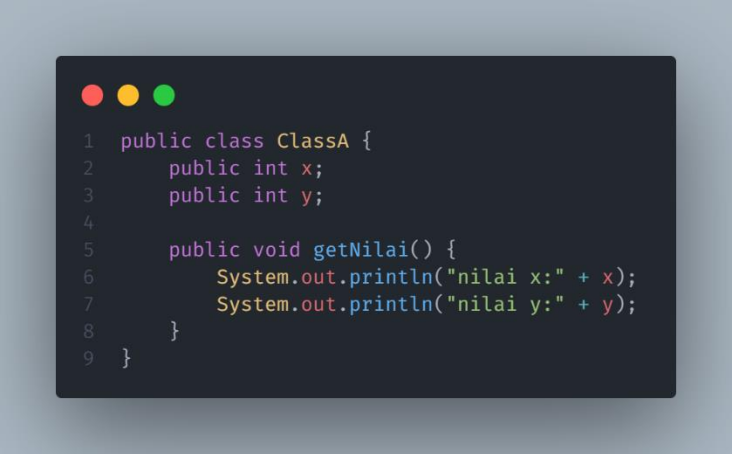
1 public class ClassB {
2     public int z;
3     public int x; // Add x and y in ClassB to access them
4     public ClassA classA;
5
6     public void getNilaiZ() {
7         System.out.println("nilai z:" + z);
8     }
9
10    public void getJumlah(int x, int y) {
11        System.out.println("jumlah:" + (x + y + z));
12    }
13
14    public void getJumlah() {
15    }
16
17    public void getNilai() {
18    }
19 }

```

```

1 public class Percobaan1 {
2     public static void main(String[] args) {
3         ClassA classA = new ClassA(); // Create an instance of ClassA
4         classA.x = 20;
5         classA.y = 30;
6
7         ClassB hitung = new ClassB();
8         hitung.z = 5;
9         hitung.classA = classA; // Set the reference to ClassA
10
11         hitung.getNilai();
12         hitung.getNilaiZ();
13         hitung.getJumlah();
14     }
15 }

```



```
1 public class ClassA {  
2     public int x;  
3     public int y;  
4  
5     public void getNilai() {  
6         System.out.println("nilai x:" + x);  
7         System.out.println("nilai y:" + y);  
8     }  
9 }
```

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

Penyebab error pada program awal adalah:

- Terdapat kesalahan dalam mencetak nilai y pada getNilai() di ClassA, yang seharusnya mencetak "nilai y" bukan "nilai x".
- Pada ClassB, method getJumlah() mencoba mengakses variabel x dan y, tetapi variabel-variabel ini tidak dideklarasikan dalam ClassB. Oleh karena itu, perlu memastikan bahwa objek hitung memiliki nilai x dan y sebelum menghitung jumlahnya.

Practicum 2

Code :

```
1 public class Percobaan2 {
2     public static void main(String[] args) {
3         ClassB hitung = new ClassB();
4
5         hitung.setX(20);
6         hitung.setY(30);
7         hitung.setZ(5);
8         hitung.getNilai();
9         hitung.getNilaiZ();
10        hitung.getJumlah();
11    }
12 }
```

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

```
1 public class ClassA {
2     public int x;
3     public 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 void getNilai() {
14        System.out.println("nilai x:" + x);
15        System.out.println("nilai y:" + y);
16    }
17 }
```

Output :

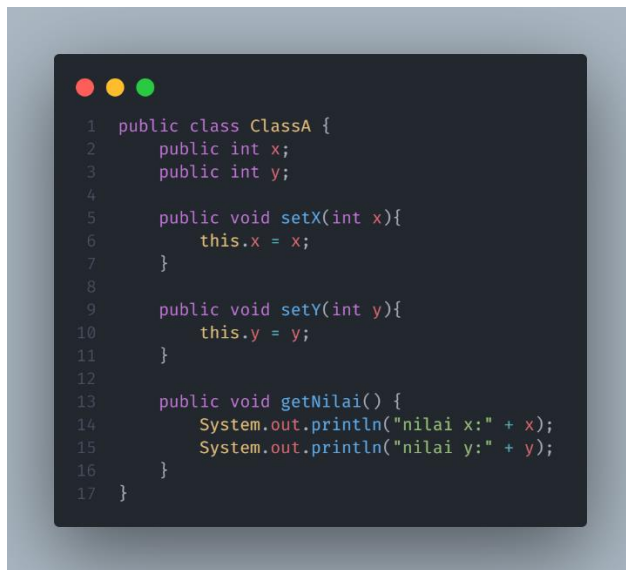
```
▼ Percobaan2.java josbheet5\prac2 4
❌ The method setX(int) is undefined for the type ClassB Java(67108964) [Ln 7, Col 16]
❌ The method setY(int) is undefined for the type ClassB Java(67108964) [Ln 8, Col 16]
❌ The method getNilai() is undefined for the type ClassB Java(67108964) [Ln 10, Col 16]
❌ The method getJumlah(int, int) in the type ClassB is not applicable for the arguments () Java(67108979) [Ln 12, Col 16]
```

Questions

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

```
1 public class Percobaan2 {
2     public static void main(String[] args) {
3         ClassB hitung = new ClassB();
4
5         hitung.setX(20);
6         hitung.setY(30);
7         hitung.setZ(5);
8         hitung.getNilai();
9         hitung.getNilaiZ();
10        hitung.getJumlah();
11    }
12 }
```

```
1 public class ClassB {
2     public int z;
3
4     public void setZ(int z){
5         this.z = z;
6     }
7     public void getNilaiZ() {
8         System.out.println("nilai z:" + z);
9     }
10
11    public void getJumlah(int x, int y) {
12        System.out.println("jumlah:" + (x + y + z));
13    }
14    public void setX(int i) {
15    }
16    public void setY(int i) {
17    }
18    public void getNilai() {
19    }
20    public void getJumlah() {
21    }
22 }
```

A screenshot of a code editor with a dark background and light-colored text. The code is for a Java class named ClassA. It has two public integer variables, x and y. There are two public void methods: setX(int x) and setY(int y), both of which assign the parameter value to the corresponding instance variable. There is also a public void method named getNilai() which prints the values of x and y to the console using System.out.println. The code is numbered from 1 to 17 on the left side of the editor.

```
1 public class ClassA {
2     public int x;
3     public 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 void getNilai() {
14        System.out.println("nilai x:" + x);
15        System.out.println("nilai y:" + y);
16    }
17 }
```

2. Jelaskan apa penyebab program pada percobaan 2 ketika dijalankan terdapat error!
 - Di ClassB, menggunakan metode getJumlah yang memerlukan dua argumen (x dan y), tetapi memanggilnya tanpa argumen di main pada percobaan 2. Kita perlu memperbarui metode getJumlah agar menggunakan nilai x, y, dan z yang ada di ClassB. Perbaikan ini akan memungkinkan program berjalan dengan benar.
 - Juga, pastikan untuk mengimpor kelas yang diperlukan. Dalam kasus ini, kita perlu mengimpor java.lang untuk menggunakan System.out.println.

Practicum 3

Code :

```
1 public class Percobaan3 {
2     public static void main(String[] args) {
3         Tabung tabung = new Tabung();
4         tabung.setSuperPhi(3.14);
5         tabung.setSuperR(10);
6         tabung.setT(3);
7         tabung.volume();
8     }
9 }
```

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

```
1 public class Bangun {
2     protected double phi;
3     protected int r;
4 }
```

Output :

```
7979e\bin - josuneets.pracs.re
Volume Tabung adalah: 942.0
PS C:\Dev\OOP>
```


Questions

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

Dalam potongan program di class Tabung, kata kunci super digunakan untuk merujuk kepada atribut atau metode dari superclass (kelas induk), dalam hal ini, kelas Bangun. Dengan super.phi

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

Dalam potongan program di class Tabung, kata kunci super digunakan untuk merujuk kepada atribut phi dan r yang ada di superclass Bangun, sementara kata kunci this digunakan untuk merujuk kepada atribut t yang ada di kelas Tabung itu sendiri.

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

Pada class Tabung, tidak perlu mendeklarasikan atribut phi dan r karena class Tabung mewarisi atribut phi dan r dari superclass Bangun menggunakan mekanisme inheritance.

Practicum 4

Code :

```
1 public class Percobaan4 {
2     public static void main(String[] args) {
3         ClassB test2 = new ClassB();
4         ClassC test3 = new ClassC();
5     }
6 }
```

```
1 public class ClassC {
2     ClassC(){
3         System.out.println("konstruktor C dijalankan");
4     }
5 }
```

```

1 public class ClassB extends ClassA {
2     ClassB(){
3         System.out.println("konstruktor B dijalankan");
4     }
5 }

```

```

1 public class ClassA {
2     ClassA(){
3         System.out.println("konstruktor A dijalankan");
4     }
5 }

```

Output :

```

konstruktor A dijalankan
konstruktor B dijalankan
konstruktor C dijalankan
PS C:\Dev\OOP>

```

Questions

1. Pada percobaan 4 sebutkan mana class yang termasuk superclass dan subclass, kemudian jelaskan alasannya!
 - `ClassA` adalah superclass karena tidak meng-extend kelas lain.
 - `ClassB` adalah subclass dari `ClassA` karena meng-extend `ClassA`.
 - `ClassC` adalah kelas mandiri yang tidak memiliki hubungan warisan (inheritance) dengan kelas lain.
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!

```

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

```

Setelah melakukan perubahan ini, jika Anda menjalankan kembali Percobaan4, kita masih tidak akan melihat perbedaan dalam hasil outputnya. Hal ini terjadi karena ClassC tidak memiliki superclass

(kelas yang di-extend), sehingga pemanggilan `super()` dalam konstruktor default tidak memiliki efek nyata dalam kasus ini. Itu biasanya digunakan untuk memanggil konstruktor superclass saat ada pewarisan.

3. Ublah isi konstruktor default ClassC seperti berikut:

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 Jelaskan bagaimana urutan proses jalannya konstruktor saat objek test dibuat!

Kode yang Anda berikan menghasilkan error ketika `super()` dipindahkan ke baris kedua dalam konstruktor default ClassC karena konstruktor harus selalu memanggil `super()` pada baris pertama jika kelas ini mewarisi kelas lain (dalam hal ini, ClassC meng-extend ClassB). Ini adalah aturan dalam Java bahwa konstruktor subclass harus memanggil konstruktor superclass sebelum melakukan tugas lain.

Output :

```
7979e\bin> josuneets.prac4
konstruktor A dijalankan
konstruktor B dijalankan
konstruktor C dijalankan
PS C:\Dev\OOP>
```

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

Fungsi `super()` di dalam konstruktor ClassC digunakan untuk memanggil konstruktor dari superclass-nya, yaitu konstruktor dari ClassB. Oleh karena itu, ketika Anda membuat objek dari ClassC, konstruktor ClassC akan terlebih dahulu memanggil konstruktor ClassB (superclass) menggunakan `super()`, dan kemudian mencetak "konstruktor C dijalankan". Jadi, `super()` di sini berfungsi untuk menjalankan konstruktor superclass sebelum melanjutkan dengan kode dalam konstruktor ClassC.

Questions

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.

```
1 package josbheet5.tasks;
2
3 public class Main {
4     public static void main(String[] args) {
5         DaftarGaji daftarGaji = new DaftarGaji(5);
6
7         Dosen dosen1 = new Dosen("12345", "Imen", "Jl. Contoh 1");
8         dosen1.setSKS(6);
9
10        Dosen dosen2 = new Dosen("67890", "Zahrawa", "Jl. Contoh 2");
11        dosen2.setSKS(8);
12
13        daftarGaji.addPegawai(dosen1);
14        daftarGaji.addPegawai(dosen2);
15
16        daftarGaji.printSemuaGaji();
17    }
18 }
```

```
1 package josbheet5.tasks;
2
3 class Pegawai {
4     private String nip;
5     private String nama;
6     private String alamat;
7
8     public Pegawai(String nip, String nama, String alamat) {
9         this.nip = nip;
10        this.nama = nama;
11        this.alamat = alamat;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public int getGaji() {
19        // Mengembalikan gaji default untuk pegawai
20        return 0;
21    }
22 }
23
```



```
1  package josbheet5.tasks;
2
3  class Dosen extends Pegawai {
4      private int jumlahSKS;
5      private static final int TARIF_SKS = 50000;
6
7      public Dosen(String nip, String nama, String alamat) {
8          super(nip, nama, alamat);
9          this.jumlahSKS = 0;
10     }
11
12     public void setSKS(int jumlahSKS) {
13         this.jumlahSKS = jumlahSKS;
14     }
15
16     @Override
17     public int getGaji() {
18         // Menghitung gaji dosen berdasarkan jumlah SKS yang diajar
19         return jumlahSKS * TARIF_SKS;
20     }
21 }
```



```
1  package josbheet5.tasks;
2
3  public class DaftarGaji {
4      private Pegawai[] listPegawai;
5      private int numPegawai;
6
7      public DaftarGaji(int maxPegawai) {
8          listPegawai = new Pegawai[maxPegawai];
9          numPegawai = 0;
10     }
11
12     public void addPegawai(Pegawai pegawai) {
13         if (numPegawai < listPegawai.length) {
14             listPegawai[numPegawai] = pegawai;
15             numPegawai++;
16         } else {
17             System.out.println("Daftar gaji sudah penuh");
18         }
19     }
20
21     public void printSemuaGaji() {
22         System.out.println("Daftar Gaji Pegawai:");
23         for (int i = 0; i < numPegawai; i++) {
24             System.out.println("Nama: " + listPegawai[i].getNama());
25             System.out.println("Gaji: " + listPegawai[i].getGaji());
26             System.out.println();
27         }
28     }
29 }
30
```