

OBJECT ORIENTED PROGRAMMING

JOBSHEET 6 INHERITANCE 2



Nama : Fahrudin Zaim Ibrahim Wicaksono
NIM : 2241720253
No. Absen : 9

STATE POLYTECHNIC OF MALANG
2023

Praktikum 1

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

- 1.
2. Penggunaan yang Tidak Tepat dari Inheritance: Jika ClassB seharusnya mewarisi ClassA tetapi tidak melakukan itu, tidak menggunakan kata kunci extends secara benar saat mendefinisikan kelas turunan.

Praktikum 2

```
1 public class ClassA {  
2     protected int x;  
3     protected 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 }
```

- 1.
2. Variabel int x dan y menggunakan encapsulasi private

Praktikum 3

1. Digunakan untuk merujuk ke superclass dari kelas saat ini. Jika Tabung adalah subclass yang memiliki superclass, penggunaan super.phi dan super.r mengacu pada variabel phi dan r yang didefinisikan di superclass. Dengan kata lain, super.phi = phi; dan super.r = r; digunakan untuk menginisialisasi nilai dari variabel phi dan r pada superclass, bukan pada kelas Tabung sendiri.

2. Digunakan untuk merujuk ke superclass dari kelas saat ini. Sedangkan 'this' digunakan untuk merujuk ke objek saat ini atau instance dari kelas tempat ia digunakan, yang artinya masih dalam 1 class.
3. Sebuah kelas dapat mengakses atribut-atribut yang ada di superclass menggunakan kata kunci 'super', asalkan atribut tersebut dideklarasikan dengan hak akses yang memungkinkan akses dari kelas turunannya.

Praktikum 4

1. Superclass : ClassA superclass dari ClassB, ClassB superclass dari ClassC
Subclass : ClassB subclass dari ClassA, ClassC subclass dari ClassB
2. Iya
3. ClassC() : Konstruktor dari ClassC akan dipanggil terlebih dahulu karena Anda membuat objek dari ClassC.
ClassB(): Setelah konstruktor ClassC memanggil, konstruktor dari ClassB dijalankan.
ClassA(): Selanjutnya, konstruktor dari ClassA dijalankan.
4. Untuk memanggil konstruktor dari kelas induknya, yaitu konstruktor dari ClassB

Assignment

```

class DaftarGaji {
    private Pegawai1[] listPegawai;
    private int jumlahPegawai;

    public DaftarGaji(int jumlah) {
        listPegawai = new Pegawai1[jumlah];
        jumlahPegawai = 0;
    }

    public void addPegawai(Pegawai1 p1) {
        if (jumlahPegawai < listPegawai.length) {
            listPegawai[jumlahPegawai] = p1;
            jumlahPegawai++;
        } else {
            System.out.println("Daftar pegawai sudah penuh");
        }
    }

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

```

```

1  class Pegawai1 {
2      private String nip;
3      private String nama;
4      private String alamat;
5      protected int gaji;
6
7      public Pegawai1(String nip, String nama, String alamat) {
8          this.nip = nip;
9          this.nama = nama;
10         this.alamat = alamat;
11         this.gaji = 0;
12     }
13
14     public String getNama() {
15         return nama;
16     }
17
18     public int getGaji() {
19         return gaji;
20     }
21 }

```

```

1  class Dosen1 extends Pegawai1 {
2      private int jumlahSKS;
3      private int tarifSKS;
4
5      public Dosen1(int jumlahSKS, int tarifSKS, String nip, String nama, String alamat) {
6          super(nip, nama, alamat);
7          this.jumlahSKS = jumlahSKS;
8          this.tarifSKS = tarifSKS;
9      }
10
11     public void setSKS(int sks) {
12         jumlahSKS = sks;
13     }
14
15     public int getGaji() {
16         return jumlahSKS * tarifSKS;
17     }
18 }

```

```

Run | Debug
2  public static void main(String[] args) {
3      DaftarGaji daftar = new DaftarGaji(jumlah:3);
4
5      Pegawai1 p1 = new Pegawai1(nip:"001", nama:"John", alamat:"Jl. ABC");
6      Dosen1 d1 = new Dosen1(jumlahSKS:10, tarifSKS:500000, nip:"002", nama:"Jane", alamat:"Jl. XYZ");
7
8      daftar.addPegawai(p1);
9      daftar.addPegawai(d1);
10
11     daftar.printSemuaGaji();
12
13 }
14 }
15

```

```

Nama Pegawai: John, Gaji: 0
Nama Pegawai: Jane, Gaji: 5000000

```