

# **PEMROGRAMAN BERORIENTASI OBJEK**

**NINF615**

**SEMESTER GASAL 2016/2017**

**MODUL PRAKTIKUM**

**PEMROGRAMAN**

**BERORIENTASI OBJEK**

**DISUSUN OLEH:**


**Tim Asisten Praktikum**

**Jurusan Teknik Elektro**

**UM**

**JURUSAN TEKNIK ELEKTRO**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

	<b>PRAKTIKUM</b> <b>PEMROGRAMAN BERORIENTASI</b> <b>OBJEK</b>	<b>P-03</b>	<b>INHERITANCE</b>
	<b>KODE MATAKULIAH : NINF615</b>		

## A. TUJUAN

Tujuan dari praktikum PBO ini adalah:

1. Mahasiswa mampu memahami konsep inheritance (pewarisan)
2. Mahasiswa mampu mendefinisikan superclass dan subclass
3. Mahasiswa mampu override method dari superclass
4. Mahasiswa mampu menerapkan keyword *this*

## B. DASAR TEORI

### 1. Inheritance (Pewarisan)

Inheritance merupakan proses pewarisan data dan method dari suatu class yang telah ada kepada suatu class baru. Class yang mewariskan disebut dengan *superclass / parent class / base class*, sedangkan class yang mewarisi (class yang baru) disebut dengan *subclass / child class / derived class*. Akan tetapi subclass tidak dapat mewarisi anggota private dari superclass-nya.

Dengan inheritance, class yang baru (subclass) akan mirip dengan class yang lama (superclass), namun memiliki karakteristik yang baru. Dalam Java, subclass hanya bisa memiliki satu superclass (single inheritance) sedangkan superclass bisa memiliki satu subclass atau lebih.

Untuk menerapkan inheritance, digunakan statement “*extends*”.

```

namaSubclass extends namaSuperclass
{
    ..... //definisi class
}
```

### 2. Keyword super

Keyword “*super*” digunakan oleh subclass untuk memanggil konstruktor yang berada pada superclass-nya.

Contoh untuk memanggil konstruktor milik superclass-nya:

```

super()
super(parameter)
```

Contoh untuk memanggil method milik superclass-nya:

```

super.namaMethod(parameter)
```

### 3. Keyword this

Kata kunci “*this*” sangat berguna untuk menunjukkan suatu member dalam class-nya sendiri. Kata kunci “*this*” dapat digunakan untuk data member, untuk method, dan untuk konstruktor. Adapun format penulisannya adalah:

<code>this.data_member</code>	-> merujuk pada data member
<code>this.nama_method</code>	-> merujuk pada method
<code>this()</code>	-> merujuk pada konstruktor

### 4. Overriding

Overriding adalah menulis kembali method sama persis, mulai dari nama method dan isinya dan mengimplementasi kembali di subclassnya. Overriding dipakai ketika menggunakan method yang sama tapi berbeda implementasinya atau dengan kata lain, overriding adalah suatu keadaan dimana method pada subclass menolak method pada parent class-nya. Overriding memiliki ciri-ciri sebagai berikut:

- Nama method harus sama
- Daftar parameter harus sama
- Return type harus sama

Jadi overriding method mempunyai nama method yang sama, jumlah parameter dan tipe parameter serta nilai kembalian (return) method yang di-override. Jika kita ingin memanggil metode superclass pada subclass dapat menggunakan statemen berikut:

<code>super.namaMethod(parameter)</code>
--

## C. LATIHAN

### 1. Latihan 1 :

- a. Tulis ketiga *class* berikut dalam 3 file yang berbeda (1 project) kemudian jelaskan hasil kompilasi ketiga *class* berikut secara berurutan

```
Pertama.java x
1 class Pertama
2 {
3     private int a = 10;
4     protected void terprotek()
5     {
6         System.out.println("Method ini hanya untuk anaknya");
7     }
8     public void info()
9     {
10        System.out.println("a = "+a);
11        System.out.println("Dipanggil pada = "+this.getClass ().getName ());
12    }
13 }
```

```
Kedua.java x
1 class Kedua extends Pertama
2 {
3     private int b = 8;
4     protected void BacaSuper()
5     {
6         System.out.println("Nilai b :"+b);
7         terprotek();
8         info();
9     }
10 }
```

```
TestPertamaKedua.java x
1 class TestPertamaKedua
2 {
3     public static void main(String[] args)
4     {
5         Kedua D2 = new Kedua ();
6         D2.BacaSuper ();
7         D2.info ();
8
9         Pertama S1 = new Pertama ();
10        S1.terprotek ();
11        S1.info ();
12    }
13 }
```

- b. Hilangkan **S1.terprotek()** dari *class* TestPertamaKedua kemudian jelaskan perbedaan hasil kompilasi dengan hasil sebelumnya.
- c. Hapus semua file yang berekstensi .class dari folder kemudian kompilasi *class* TestPertamaKedua saja. Perhatikan file apa yang dihasilkan dari kompilasi tersebut.
- d. Jelaskan fungsi dari perintah **this.getClass.getName()**

## 2. Latihan 2

Penggunaan keyword *super* untuk memanggil konstruktor dari kelas induk.

a. Tulis ketiga class berikut dalam 3 file yang berbeda (1 project)

```
1 class Person
2 {
3     protected String name;
4     protected int age;
5
6     public Person(String name, int age)
7     {
8         this.name = name;
9         this.age = age;
10    }
11    //metode
12    public void info()
13    {
14        System.out.println("Nama : "+this.name);
15        System.out.println("Usia : "+this.age);
16    }
17    //akhir kelas Program
18 }
```

```
1 class Employ extends Person
2 {
3     private String noKaryawan;
4     //konstruktor
5     public Employ (String noKaryawan, String name, int age)
6     {
7         super(name, age);
8         this.noKaryawan = noKaryawan;
9     }
10
11    //metode
12    public void info()
13    {
14        System.out.println("No.karyawan : "+this.noKaryawan);
15        super.info();
16    }
17 }
```

```
1 public class KonstruktorSuperKelas
2 {
3     public static void main (String [] args)
4     {
5         Employ programmer1 = new Employ ("12345678", "Budi", 33);
6         programmer1.info();
7     }
8 }
```

### 3. Latihan 3

Tulis program berikut!

```
1 public class Hewan
2 {
3     public static void testClassMethod()
4     {
5         System.out.println("The Class Method in Hewan...");
6     }
7     public void testInstanceMethod()
8     {
9         System.out.println("The Instance Method in Hewan...");
10    }
11 }
12 //meng-overwrite method pada class Hewan
```

```
1 public class Gajah extends Hewan
2 {
3     public static void testClassMethod()
4     {
5         System.out.println("The Class Method in Hewan...");
6     }
7
8     //mengoverride method pada class pada class Hewan
9     public void testInstanceMethod()
10    {
11        System.out.println("The Instance method in Gajah...");
12    }
13    public static void main(String[] args)
14    {
15        Kucing myGajah = new Gajah();
16        Hewan myHewan = myGajah;
17        Hewan.testClassMethod();
18        myHewan.testInstanceMethod();
19    }
20 }
```

#### 4. Latihan 4

Ketikkan kode program berikut kemudian perhatikan keyword *super* pada program

```
1 class A
2 {
3     private int a;
4     public void setA (int nilai)
5     {
6         a = nilai;
7     }
8     public int getA()
9     {
10        return a;
11    }
12    public void tampilkanNilai()
13    {
14        System.out.println("Nilai a : "+getA());
15    }
16 }
```

```
1 class B extends A
2 {
3     private int b;
4     public void setB (int nilai)
5     {
6         b = nilai;
7     }
8     public int getB()
9     {
10        return b;
11    }
12    //melakukan override terhadap method tampilkanNilai()
13    //yang terhadap pada kelas A
14    public void tampilkanNilai()
15    {
16        super.tampilkanNilai(); //memanggil method dalam kelas A
17        System.out.println("Nilai b : "+getB());
18    }
19 }
```

```
1 class DemoOverride2
2 {
3     public static void main (String [] args)
4     {
5         B obj = new B();
6         obj.setA(50);
7         obj.setB(150);
8         //akan memanggil method yang terdapat pada kelas B
9         obj.tampilkanNilai();
10    }
11 }
```



## 5. Latihan 5

Tulis dan simpan kode program berikut!

```
1  class Truck
2  {
3      //the Truck class has three fields
4      public int cadence;
5      public int gear;
6      public int speed;
7
8      //the Truck class has one constructor
9      public void Truck(int startCadence, int startSpeed, int startGear)
10     {
11         gear = startGear;
12         cadence = startCadence;
13         speed = startSpeed;
14     }
15
16     //the Truck class has four methods
17     public void setCadence(int newValue)
18     {
19         cadence = newValue;
20     }
21     public void setGear(int newValue)
22     {
23         gear = newValue;
24     }
25     public void applyBrake(int decrement)
26     {
27         speed -= decrement;
28     }
29     public void speedUp(int increment)
30     {
31         speed += increment;
32     }
33 }
```

```

1 public class MountainTruck extends Truck
2 {
3     //the MountainTruck subclass adds one field
4     public int seatHeight;
5
6     //the MountainTruck subclass has one constructor
7     public void MountainTruck(int startHeight, int startCadence, int startSpeed, int startGear)
8     {
9         super.Truck(startHeight,startCadence,startGear);
10        seatHeight = startHeight;
11    }
12
13    //the MountainTruck subclass adds one method
14    public void setHeight(int newValue)
15    {
16        seatHeight = newValue;
17    }
18    public static void main(String[] abc)
19    {
20        System.out.println("Mulai Berjalan");
21        MountainTruck MB = new MountainTruck();
22        MB.MountainTruck(0,0,0,0);
23        System.out.println("gear = "+MB.gear);
24        System.out.println("speed = "+MB.speed);
25        MB.setGear(2);
26        MB.speedUp(2);
27        System.out.println("gear = "+MB.gear);
28        System.out.println("speed = "+MB.speed);
29        MB.speedUp(2);
30        System.out.println("gear = "+MB.gear);
31        System.out.println("speed = "+MB.speed);
32    }
33 }

```

## D. TUGAS PRAKTIKUM

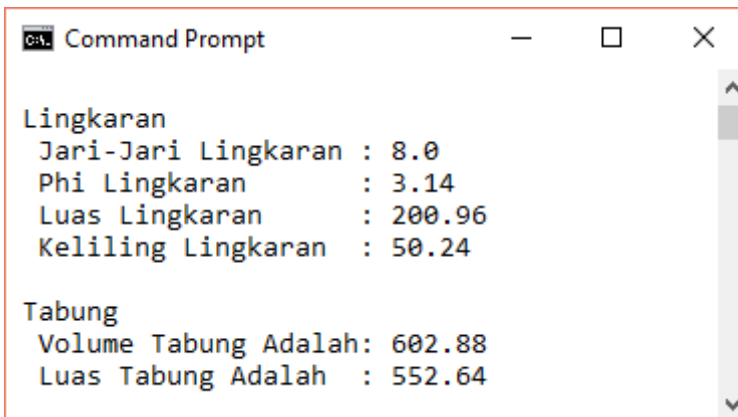
### 1. Tugas Praktikum 1

Buatlah dengan menggunakan konsep inheritance dimana kelas induknya adalah Lingkaran.java. Berikut ini merupakan script program dari kelas Lingkaran.java.

```
1 public class Lingkaran
2 {
3     float jari = 8;
4     double phi = 3.14;
5
6     public double Lingkaran()
7     {
8         return jari * jari * phi;
9     }
10    public double Keliling()
11    {
12        return(2 * jari * phi);
13    }
14 }
```

### 2. Tugas Praktikum 2

Setelah membuat kelas induk tersebut, buatlah kelas turunan dari Lingkaran yaitu Tabung.java, dan kelas Hasil.java untuk menampilkan nilai dari Kelas Lingkaran dan Kelas Tabung. Sehingga nantinya hasil run programnya seperti berikut :



```
Command Prompt
Lingkaran
Jari-Jari Lingkaran : 8.0
Phi Lingkaran      : 3.14
Luas Lingkaran     : 200.96
Keliling Lingkaran : 50.24

Tabung
Volume Tabung Adalah: 602.88
Luas Tabung Adalah  : 552.64
```

## **E. TUGAS RUMAH**

1. Buatlah sebuah program dengan menerapkan “inheritance” dalam kehidupan perkuliahan.
2. Jelakan perbedaan Overriding dan Overloading