

## LAPORAN PRAKTIKUM

# PEMROGRAMAN BERORIENTASI OBJEK

Modul : 5 Tanggal praktikum : 11-05-2023 (Format: DD-MM-YYYY)

---

Nama : Muhamad Fahmi

NIM : 20220810029

Kelas : TINFC-2022-02 (contoh: TINFC-2022-01)

---

## JUDUL/TEMA MODUL

### 1. Tujuan Pembelajaran

Tuliskan tujuan pembelajaran/praktikum. Sesuaikan dengan tujuan pembelajaran pada modul praktikum.

#### A. TUJUAN

1. Memahami Inheritance
2. Memahami Konsep inheritance
3. Mampu Membuat Program java inheritance

### 2. Dasar Teori

Pada bagian ini, silahkan kalian tulis/tambahkan materi pendukung untuk kegiatan praktikumnya. Tambahkan gambar / tabel/ flowchart apabila diperlukan.

**Konsep pewaris (inheritance):** Proses pembentukan kelas baru dari kelas yang sudah menjadi kelas Superclass. Proses pembentukan kelas menghasilkan hireraki kelas yaitu puncak hirerarki yang disebut **kelas abstrak**.

**Inheritance** adalah sebuah konsep pewarisan sifat berupa variabel dan fungsi yang dimiliki oleh class untuk diwariskan kepada kelas-kelas yang lain (Zarita, 2021)

**Konsep dari inheritance** sendiri yaitu untuk membuat struktur class pada pemrograman yang dimana struktur tersebut terdapat sebuah Parentclass atau Superclass sebagai induk kelas dan Subclass sebagai anak kelas. Konsep tersebut merupakan sebuah percabangan dari sebuah class (Superclass) yang memiliki sifat umum menjadi sebuah class(Subclass) yang memiliki sifat lebih spesifik (Zarita, 2021).

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

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

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

Keyword **“super”** digunakan oleh subclass untuk memanggil constructor atau method yang ada pada superclass-nya.

Contoh untuk memanggil constructor milik superclass-nya:

```
super ()  
super(parameter)  
Contoh untuk  
memanggil method  
Milik superclass-nya:
```

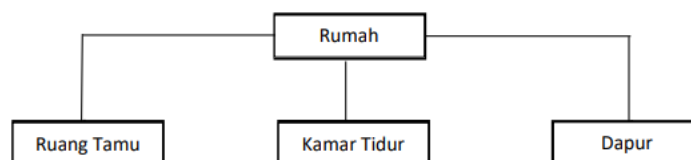
```
super.namaMethod(parameter)
```

### 3. Hasil dan Pembahasan

- Bagian ini berisi hasil praktikum yang telah anda kerjakan yang meliputi *pretest*, tahapan praktikum, tugas, dan *posttest*.
- Lampirkan hasil pekerjaan berupa tangkapan layar (*schreenshoot*) kegiatan praktikum sesuai dengan langkah kerja yang ada.

#### I. PRETEST

1. Gambarkan contoh diagram inheritance dalam kehidupan sehari-hari!



## PRAKTIKUM 1

```
1 //superClass
2 class Person{
3     private String Nama;
4     private String Alamat;
5     private int Umur;
6     public Person(){
7     }
8     public Person(String Nama, String Alamat, int Umur){
9         this.Nama = Nama;
10        this.Alatmat = Alamat;
11        this.Umur = Umur;
12    }
13    public String getName(){
14        return Nama;
15    }
16    public String getAlamat(){
17        return Alamat;
18    }
19    public int getUmur(){
20        return Umur;
21    }
22    public void setName(String Nama){
23        this.Nama = Nama;
24    }
25    public void setAlamat(String Alamat){
26        this.Alatmat = Alamat;
27    }
28    public void setUmur(int Umur){
29        this.Umur = Umur;
30    }
31    public void infoPerson(){
32        System.out.println("\nInfo Person");
33        System.out.println("Nama\t\t: " + getName());
34        System.out.println("Alamat\t\t: " + getAlamat());
35        System.out.println("Umur\t\t: " + getUmur());
36    }
37 }
```

### Modul5\_inputSuperClass(Person)

Analisis : class person ini disebut super class yang fungsinya untuk memanggil constructor pada kata kunci this digunakan pada pembuatan kelas dan untuk menyatakan objek sekarang

```
1 //Sub Class
2 class Dosen extends Person{
3     private String idDosen;
4     public Dosen(){
5     }
6     public Dosen(String Nama, String Alamat, int Umur, String idDosen){
7         super(Nama, Alamat, Umur);
8         this.idDosen = idDosen;
9     }
10    public String getIdDosen(){
11        return idDosen;
12    }
13    public void setIdDosen(String idDosen){
14        this.idDosen = idDosen;
15    }
16    public void infoDosen(){
17        System.out.println("\nInfo Dosen");
18        System.out.println("idDosen\t\t: " + getIdDosen());
19        System.out.println("Nama\t\t: " + getName());
20        System.out.println("Alamat\t\t: " + getAlamat());
21        System.out.println("Umur\t\t: " + getUmur());
22    }
23 }
```

### Modul5\_inputSubClass(Dosen)

Analisis : pada class ini disebut sub class sama saja untuk memanggil constructor dari superclass terdekat dengan pemanggil constructor super, fungsi extends disini untuk memanggil person.java sehingga dalam file dosen.java tidak menggunakan banyak script

```

1 //Sub Class
2 class Mahasiswa extends Person
3 {
4     private String NIM;
5     public Mahasiswa()
6     {
7     }
8     public Mahasiswa(String Nama, String Alamat, int Umur, String NIM)
9     {
10        super(Nama, Alamat, Umur);
11        this.NIM = NIM;
12    }
13    public String getNIM()
14    {
15        return NIM;
16    }
17    public void setNIM(String NIM)
18    {
19        this.NIM = NIM;
20    }
21    public void infoMahasiswa()
22    {
23        System.out.println("\nInfo Mahasiswa");
24        System.out.println("idDosen\t\t: " + getNIM());
25        System.out.println("Nama\t\t: " + getName());
26        System.out.println("Alamat\t\t: " + getAlamat());
27        System.out.println("Umur\t\t: " + getUmur());
28    }
29 }

```

### Modul5\_inputSubClass(Mahasiswa)

Analisis : pada class ini pun sama seperti diatas semua script disimpan di file person.java

```

1 class tampil{
2     public static void main(String[] args) {
3         Person a = new Person ("Bere", "Cijoho", 29);
4         a.infoPerson();
5         Dosen b = new Dosen ("Gugun", "Kuningan", 45, "17.05.05");
6         b.infoDosen();
7         Mahasiswa c = new Mahasiswa ("Ina", "Lebakwangi", 17, "13.11.7066");
8         c.infoMahasiswa();
9         System.out.println("-----");
10    }
11 }

```

### Modul5\_inputClassTampil

Analisi : class ini untuk menampilkan semua script program diatas dengan pemanggilan a.infoPerson b.infoDosen c.infoMahasiswa

```

Output - MODUL 5 (run) X
run:

Info Person
Nama      : Bere
Alamat    : Cijoho
Umur      : 29

Info Dosen
idDosen   : 17.05.05
Nama      : Gugun
Alamat    : Kuningan
Umur      : 45

Info Mahasiswa
idDosen   : 13.11.7066
Nama      : Ina
Alamat    : Lebakwangi
Umur      : 17
-----
BUILD SUCCESSFUL (total time: 2 seconds)

```

### Modul5\_Output

## PRAKTIKUM 2

```
1 public class Kakek
2 {
3     protected String namekakek;
4     protected String address;
5     public Kakek()
6     {
7         System.out.println("\n Program Demo Inheritance");
8         System.out.println("-----");
9         System.out.println("  Masukan Konstruktor Kakek  ");
10        System.out.println("-Dijalankan oleh class Bapak-");
11        namekakek = "Joyo Cokro Samudro";
12        address = "Siemam Djogjakarta";
13    }
14    public Kakek (String namekakek, String address)
15    {
16        this.namekakek = namekakek;
17        this.address = address;
18    }
19    @
20    public String getName()
21    {
22        return namekakek;
23    }
24    @
25    public String getAddress()
26    {
27        return address;
28    }
29 }
```

### Modul5\_inputSuperClass(kakek)

```
1 public class bapak extends Kakek
2 {
3     protected String namebapak;
4     protected String addressbapak;
5     public bapak()
6     {
7         System.out.println("Name Kakek : "+namekakek);
8         System.out.println("Address Kakek : "+address);
9         System.out.println("\n");
10        System.out.println("Masukkan Konstruktor Bapak");
11        System.out.println("-Dijalankan oleh class oom-");
12        namebapak = "BOMARDI";
13        addressbapak = "KONINGAN";
14    }
15    public bapak (String namebapak, String addressbapak)
16    {
17        this.namebapak = namebapak;
18        this.addressbapak = addressbapak;
19    }
20    @
21    public String getName()
22    {
23        return namebapak;
24    }
25    @
26    public String getAddress()
27    {
28        return addressbapak;
29    }
30    public static void main(String[] args)
31    {
32        bapak bari = new bapak();
33    }
```

### Modul5\_inputSubClass(Bapak)

Analisis : pada program ke 1 & 2 itu sama saja yang membedakan hanya nama classnya pada perintah protected digunakan untuk menyatakan bahwa kelas/method/attribute tersebut dapat diakses oleh class lain yang berada dalam satu package atau kelas lain tersebut merupakan turunannya protected ini sama seperti public hanya dapat digunakan dalam class yang sama

```

1 public class cucu extends bapak
2 {
3     public cucu ()
4     {
5         System.out.println("Nama Bapak :"+namebapak);
6         System.out.println("Address Bapak :"+addressbapak);
7     }
8     public static void main(String[] args)
9     {
10        String namaucucu;
11        cucu kagan = new cucu();
12        System.out.println("\nSaya adalah CUCU");
13        System.out.println("Nama Saya : Rio Andriyat Krisdiawan");
14        System.out.println("");
15    }
16 }

```

### Modul5\_inputSubClass(cucu)

Analisis : class ini untuk menampilkan seluruh script yang tersimpan pada file bapak.java

```

Output - MODUL 5 (run) X
run:

Program Demo Inheritance
=====
Masukkan Konstruktor Kakek
-Dijalankan oleh class Bapak-
Nama Kakek : Joyo Cokro Aminoto
Address kakek : Sleman Djogjakarta

Masukkan Konstruktor Bapak
-Dijalankan oleh class cucu-
Nama Bapak :ROHAEDI
Address Bapak :KUNINGAN

Saya adalah CUCU
Nama Saya : Rio Andriyat Krisdiawan

BUILD SUCCESSFUL (total time: 0 seconds)
|

```

### Modul5\_Output

## II. POSTTEST

1. Buatlah program Konstruktor dengan case studi kasus yang berbeda dari praktikum diatas.

Jawaban:

Class barang.java

```
1 public class barang{
2     String merkbarang;
3     int hargabarang;
4     public barang(String merkbarang,int hargabarang){
5         this.merkbarang = merkbarang;
6         this.hargabarang = hargabarang;
7     }
8     public String getNama(){
9         return merkbarang;
10    }
11    public int getHarga(){
12        return hargabarang;
13    }
14    public void setNama(String nama){
15        this.merkbarang = nama;
16    }
17    public void setHarga(int harga){
18        this.hargabarang = harga;
19    }
20    public void infobarang(){
21        System.out.println("\n info barang");
22        System.out.println("Nama Barang\t:" + getNama());
23        System.out.println("Harga Barang\t:" + getHarga());
24    }
25 }
26 }
27 }
```

Class mobil.java

```
1 public class mobil extends barang{
2     String warna;
3     public mobil (String merkbarang,int hargabarang,String warna){
4         super(merkbarang,hargabarang);
5         this.warna = warna;
6     }
7     public String getWarna(){
8         return warna;
9     }
10    public void setWarna(String warna){
11        this.warna = warna;
12    }
13    public void infomobil(){
14        System.out.println("\nINFO MOBIL");
15        System.out.println("Merk Mobil\t: " + getNama());
16        System.out.println("Harga Mobil\t: " + getHarga());
17        System.out.println("Warna Mobil\t: " + getWarna());
18    }
19 }
20
```

Class motor.java



```

1 public class motor extends mobil{
2     public motor(String merkbarang,int hargabarang,String warna){
3         super(merkbarang,hargabarang,warna);
4     }
5     public void infomotor(){
6         System.out.println("\nINFO MOTOR");
7         System.out.println("Merk Motor\t: " + getNama());
8         System.out.println("Harga Motor\t: " + getHarga());
9         System.out.println("Warna Motor\t: " + getWarna());
10    }
11 }
12

```

Class tampilanbarang.java

```

1 class tampilanbarang{
2     public static void main(String[] args){
3         barang a =new barang("Laptop",7000000);
4         a.infobarang();
5         mobil b =new mobil("Honda",200000000,"putih");
6         b.infomobil();
7         motor c =new motor("Yamaha",10000000,"Hitam");
8         c.infomotor();
9     }
10 }
11

```

Hasil output

#### Output - modul 5 (run)

```
run:
  info barang
  Nama Barang      :Laptop
  Harga Barang     :7000000

  INFO MOBIL
  Merk Mobil       : Honda
  Harga Mobil      : 200000000
  Warna Mobil      : putih

  INFO MOTOR
  Merk Motor       : Yamaha
  Harga Motor      : 10000000
  Warna Motor      : Hitam
BUILD SUCCESSFUL (total time: 0 seconds)
```

### III. TUGAS MANDIRI

1. Modifikasi program Praktikum 2, sehingga terdapat sub class cicit. Jawab :

```

1 public class cucu extends bapak
2 {
3     protected String namecucu;
4     protected String addresscucu;
5     public cucu ()
6     {
7         System.out.println("Nama Bapak : "+namebapak);
8         System.out.println("Address Bapak : "+addressbapak);
9         System.out.println("\n");
10        System.out.println("Buatkan Constructor Cucu");
11        System.out.println("-Buatlah class cicit-");
12        namecucu = "Rio Andriyat Kristiawan";
13        addresscucu = "HONGKONG";
14    }
15    public cucu (String namecucu, String addresscucu)
16    {
17        this.namecucu = namecucu;
18        this.addresscucu = addresscucu;
19    }
20    public String getName()
21    {
22        return namecucu;
23    }
24    public String getAddress()
25    {
26        return addresscucu;
27    }
28    public static void main(String[] args)
29    {
30        cucu lagan = new cucu();
31    }

```

Modul5\_inputSubClass(cucu)

```

1 public class cicit extends cucu
2 {
3     public cicit()
4     {
5         System.out.println("Nama Cucu : " + namecucu);
6         System.out.println("Address Cucuc : " + addresscucu);
7     }
8     public static void main(String[] args)
9     {
10        String namacicit;
11        cicit lagan = new cicit();
12        System.out.println("\nSaya adalah CICIT");
13        System.out.println("Nama Saya : Maya Ratnasari");
14        System.out.println("");
15    }
16 }

```

Modul5\_inputSubClass(cicit)

```

Output - MODUL 5 (run) X
SUM:
Program Demo Inheritance
=====
Masukkan Konstruktor Kakek
-Dijalankan oleh class Bapak-
Nama Kakek : Joyo Cokro Aminoto
Address kakek : Sleman Djogjakarta

Masukkan Konstruktor Bapak
-Dijalankan oleh class cucu-
Nama Bapak : ROHAEDI
Address Bapak : KUNINGAN

Masukkan Konstruktor Cucu
-Dijalankan oleh class cicit-
Nama Cucu : Rio Andrayat Kriediawan
Address Cucu : SUNINGAN

Saya adalah CICIT
Nama Saya : Maya Ratnasari

BUILD SUCCESSFUL (total time: 0 seconds)

```

Modul5\_Output

1.

#### 4. Kesimpulan

Buatlah kesimpulan dari pembahasan praktikum yang telah Anda kerjakan.

Pada praktikum modul 4 ini membahas bagaimana cara penerapan inheritance dalam program java. dan membuat efisien dengan menggunakan inheritance program java

#### 5. Referensi

Pada bagian ini cantumkan daftar pustaka/referensi yang ada gunakan dalam menyusun laporan praktikum ini.

Modul praktikum

