

## LAPORAN PRAKTIKUM

# PEMROGRAMAN BERORIENTASI OBJEK

Modul : 7 Tanggal praktikum : 25-05-2023 (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.

1. Mengetahui dan Memahami konsep Abstract class
2. Mampu menerapkan Abstract class dalam oop

### 2. Dasar Teori

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

Abstract class adalah kelas yang terletak pada posisi tertinggi pada hierarki kelas. Kelas ini digunakan sebagai basis bagi penurunan kelas lainnya, sehingga abstract class tidak dapat diinstansiasi secara langsung menjadi object. Suatu abstract class bisa mengandung method kongkrit dan/atau method abstract. Abstract method merupakan suatu method yang tidak mempunyai badan method. Hanya berupa nama method dan parameter inputan method. Ketika suatu abstract class diturunkan/ diextends, maka subclass harus mengimplementasikan semua method abstract pada parent/ base class. Jika tidak mengimplementasikan semua method abstract maka subclass harus dideklarasikan sebagai abstract class. Cara untuk membuat sebuah class abstrak adalah:

```
akses_modifier abstract class
namaClass Abstrak
{
```

.....// definisi class

```
}
```

Contoh :

```
public abstract class Hewan{
protected String jenis;
public Hewan(){ public String toString() { return " Jenisku adalah "+jenis +" dan suaraku "+ suara();
} public abstract String
suara();
}
```

Java memiliki aturan-aturan dalam penggunaan method abstrak dan class abstrak sebagai berikut:

1. Class yang di dalamnya terdapat abstract method harus dideklarasikan sebagai abstract class.
2. Abstract class tidak dapat diinstansi, tetapi harus di turunkan.
3. Abstract class tidak dapat diinstansi (menjadi objek dari class abstract), tetapi kita dapat mendeklarasikan suatu variable yang bertipe abstract class dan membuat instansi dari variable tersebut yang bertipe class turunan dari abstract class tersebut (teknik polymorphism).
4. Sebuah class dapat dideklarasikan sebagai abstract class meskipun class tersebut tidak memiliki abstract method.
5. Abstract method tidak boleh mempunyai body method dan demikian juga sebaliknya bahwa method yang tidak ditulis body methodnya maka harus dideklarasikan sebagai abstract method.

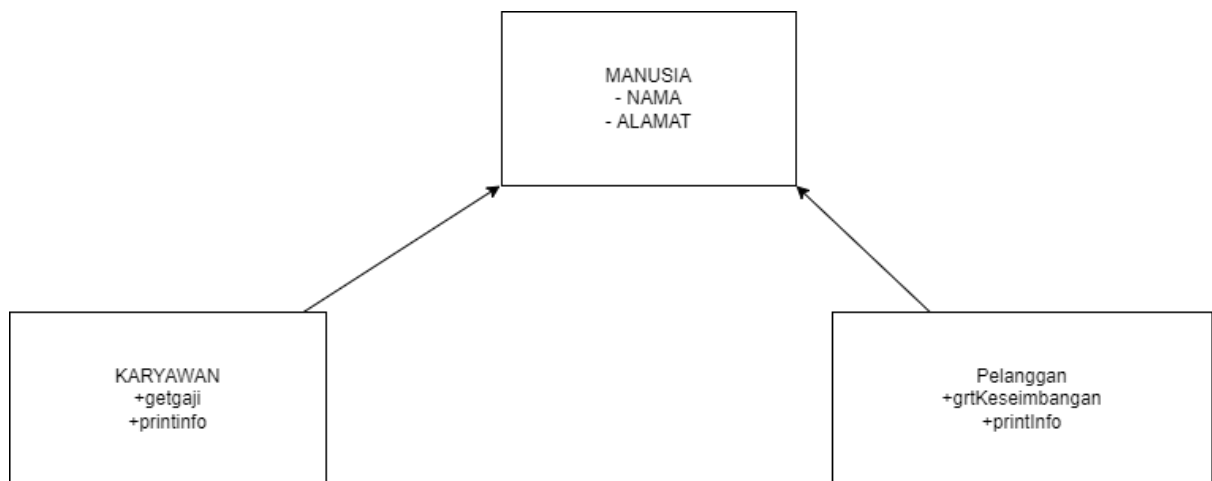
### 3. Hasil dan Pembahasan

- 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.

#### PRETEST

1. Contoh diagram Abstract Class dalam kehidupan sehari-hari

Jawaban :



#### PRAKTIKUM

##### PRAKTIKUM 1

2. Pulau.javascript java

```

10  abstract class Pelas {
11      -method final String getId() = "id_pelas";
12      public abstract String nama();
13      public String getIdRegenerasi();
14      return getId();
15  }
16
17  class Jaka version Pelas {
18      -method nama();
19      public String nama();
20      nama = "Jaka";
21      return nama;
22  }
23
24  class PelasRegenerasi version Pelas {
25      -method nama();
26      public String nama();
27      nama = "Pelas Regenerasi";
28      return nama;
29  }
30
31  class JakaRegenerasi version Jaka {
32      -method getIdRegenerasi();
33      public getIdRegenerasi() {
34          System.out.println("Id Jaka Regenerasi: " +
35              getIdRegenerasi());
36          System.out.println("Id: " + getId());
37          System.out.println("Nama: " + nama());
38          System.out.println("Nama Regenerasi: " + getIdRegenerasi());
39      }
40  }
41
42  class PelasRegenerasiTime version PelasRegenerasi {
43      -method getIdRegenerasi();
44      public getIdRegenerasi() {
45          System.out.println("Id Pelas Regenerasi: " +
46              getIdRegenerasi());
47          System.out.println("Id: " + getId());
48          System.out.println("Nama: " + nama());
49          System.out.println("Nama Regenerasi: " + getIdRegenerasi());
50      }
51  }
52
53  class Main {
54      public static void main(String[] args) {
55          JakaRegenerasi JR = new JakaRegenerasi();
56          PelasRegenerasiTime PR = new PelasRegenerasiTime();
57          JR.getIdRegenerasi();
58          PR.getIdRegenerasi();
59      }
60  }

```

Hasil running seperti dibawah ini :

```

1  C:\Users\user> javac Main.java
2
3  C:\Users\user> java Main
4
5  Id Jaka Regenerasi: 0
6  Id: 0
7  Nama: Jaka
8  Nama Regenerasi: 0
9
10 Id Pelas Regenerasi: 0
11 Id: 0
12 Nama: Pelas Regenerasi
13 Nama Regenerasi: 0

```

## Praktikum2

### 1. MakhlukHidup.java



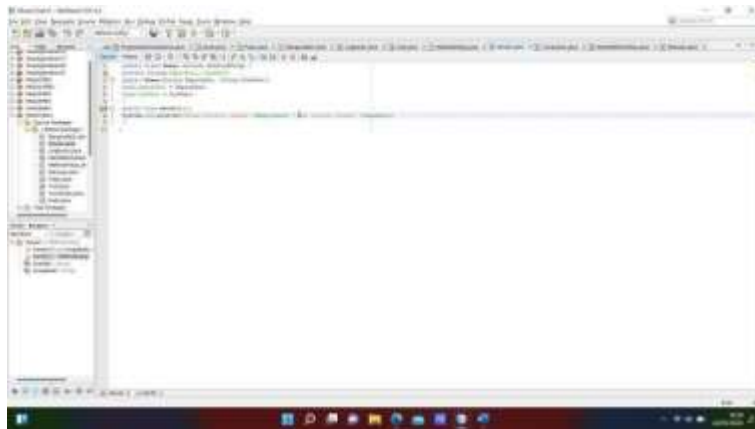
```
1 // MakhlukHidup.java
2
3 public class MakhlukHidup {
4     public void bernafas() {
5         System.out.println("Makhluk bernafas");
6     }
7     public void makan() {
8         System.out.println("Makhluk makan");
9     }
10 }
```

### 2. Manusia.java (dirubah karena error)



```
1 // Manusia.java
2
3 public class Manusia extends MakhlukHidup {
4     private String nama;
5     private String umur;
6     private String jenis;
7
8     public void bernafas() {
9         System.out.println("Manusia bernafas melalui " + jenis);
10     }
11 }
```

### 3. Hewan.java



### 4. Tumbuhan.java

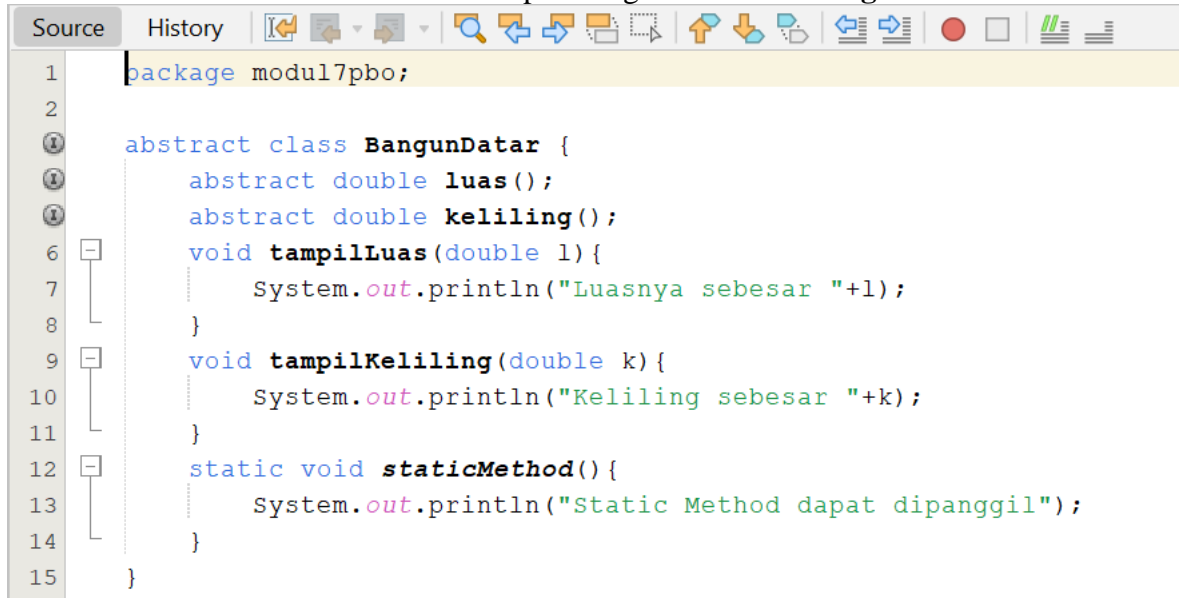


### 5. MainMakhlukHidup.java Outputnya



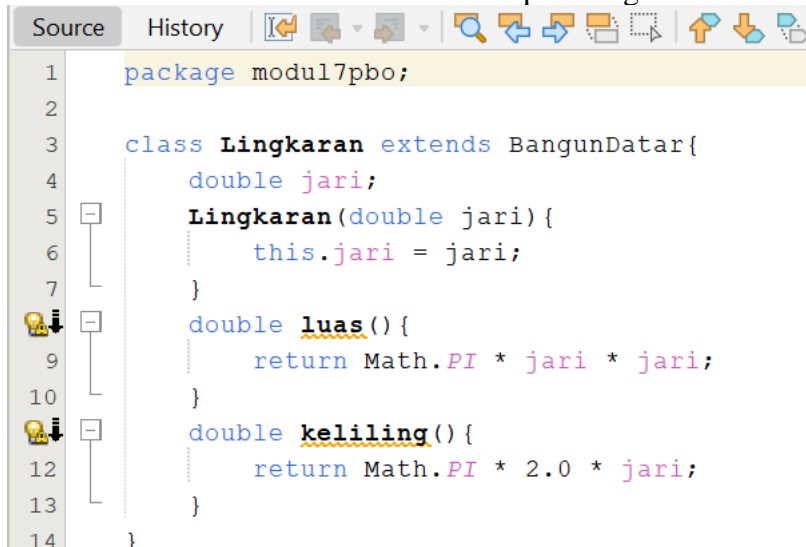
### PRAKTIKUM 3

1. Buat File Empty Java File Baru
2. Ketikan Source code berikut ini dan simpan dengan nama file **Bangun Datar.Java**



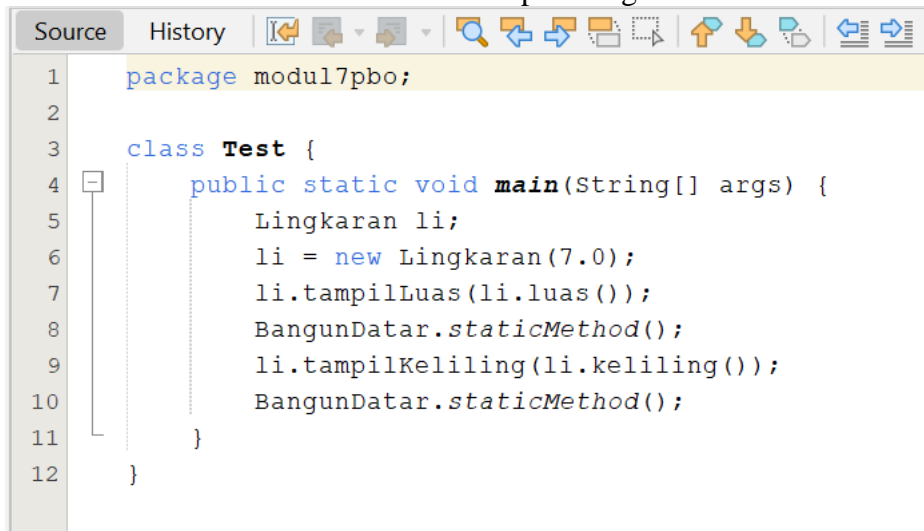
```
Source History
1 package modul7pbo;
2
3 abstract class BangunDatar {
4     abstract double luas();
5     abstract double keliling();
6     void tampilLuas(double l){
7         System.out.println("Luasnya sebesar "+l);
8     }
9     void tampilKeliling(double k){
10        System.out.println("Keliling sebesar "+k);
11    }
12    static void staticMethod(){
13        System.out.println("Static Method dapat dipanggil");
14    }
15 }
```

3. Buat File Empty Java File Baru
4. Ketikan Source code berikut ini dan simpan dengan nama file **Lingkaran.Java**



```
Source History
1 package modul7pbo;
2
3 class Lingkaran extends BangunDatar{
4     double jari;
5     Lingkaran(double jari){
6         this.jari = jari;
7     }
8     double luas(){
9         return Math.PI * jari * jari;
10    }
11    double keliling(){
12        return Math.PI * 2.0 * jari;
13    }
14 }
```

5. Buat File Empty Java File Baru
6. Ketikan Source code berikut ini dan simpan dengan nama file **Test.Java**



```
1 package modul7pbo;
2
3 class Test {
4     public static void main(String[] args) {
5         Lingkaran li;
6         li = new Lingkaran(7.0);
7         li.tampilLuas(li.luas());
8         BangunDatar.staticMethod();
9         li.tampilKeliling(li.keliling());
10        BangunDatar.staticMethod();
11    }
12 }
```

7. Jalankan File **Test.Java** tersebut,Amati Hasilnya!

Output nya:

```
compile-single:
run-single:
Luasnya sebesar 153.93804002589985
Static Method dapat dipanggil
Keliling sebesar 43.982297150257104
Static Method dapat dipanggil
BUILD SUCCESSFUL (total time: 0 seconds)
```

## POSTTEST

1. Buatlah Program **Abstract class** dengan case studi yang berbeda dari praktikum diatas.

```
abstract class Benua {
    static final String Benua = "didunia";
    public abstract String nama();
    public String ambilNegara(){ return Benua;
    }
}
class Asia extends Benua{ String namap;
public String nama(){ namap = "Benua Asia"; return namap;
}
}
class Eropa extends Benua{ String namap;
public String nama(){ namap = "Benua Eropa"; return namap;
}
}
class Jepang extends Asia{void namaNegara(){
System.out.println("Ini lah benua yang ada "+ambilNegara());
System.out.println("Ini "+ nama());
System.out.println("Ini nagara jepang");
System.out.println("Negara yang dengan julukan negara Matahari Terbit");
}
}
class Inggris extends Eropa{void namaNegara(){
System.out.println("Ini Benua yang berada di "+ambilNegara());
System.out.println("Ini "+ nama());
System.out.println("Ini negara Inggris");
System.out.println("Sering disebut negara Ratu Elizabet");
}
}
class BenuaTest{
public static void main(String[] args){Jepang Jp = new Jepang();
Inggris ing = new Inggris ();
Jp.namaNegara();
System.out.println("=====");
System.out.println(""); ing.namaNegara();
}
}
```



Output nya:

```
compile-single:
run-single:
Ini lah benua yang ada didunia
Ini Benua Asia
Ini nagara jepang
Negara yang dengan julukan negara Matahari Terbit
=====

Ini Benua yang berada di didunia
Ini Benua Eropa
Ini negara Inggris
Sering disebut negara Ratu Elizabet
BUILD SUCCESSFUL (total time: 0 seconds)
```

## TUGAS INDIVIDU

1. Apa perbedaan Abstract dan Interface, jelaskanJawab :

- Abstrak class

Abstract Class merupakan Class yang khusus dibuat untuk keperluan inheritance. Tujuan dari pembuatan abstract class ini ialah untuk membuat definisi umum bagi class-class yang akan menjadi turunan darinya.

Abstract Class tidak bisa diinstansiasi. Abstract method tidak mempunyai implementasi, sehingga penulisannya dilakukan hanya dengan mengikuti semikolon, bukan blok method {} seperti biasanya. Abstract method ini bisa digunakan oleh Class turunannya dengan melakukan override.

Abstract class hanya bisa digunakan sebagai super class, tapi juga bisa diturunkan dari class abstract lainnya. Untuk mendeklarasikan sebuah abstract class digunakan keyword

`abstract, [abstract] class [class_name]` Sebuah abstract class pada dasarnya tidak jauh bedanya dengan class lainnya, yakni juga berisi method yang menggambarkan karakteristik dari kelas abstract tersebut, bedanya yakni sebuah abstract class bisa berisi method tanpa diimplementasikan artinya sebuah method tanpa body, method seperti ini disebut method abstract.

- Interface

Interface adalah jenis khusus dari blok yang hanya berisi method signature (atau constant). Interface mendefinisikan sebuah (signature) dari sebuah kumpulan method tanpa tubuh. Interface mendefinisikan sebuah cara standar dan umum dalam menetapkan sifat-sifat dari class-class. Mereka menyediakan class-class, tanpa memperhatikan lokasinya dalam hirarki class, untuk mengimplementasikan sifat-sifat yang umum. Dengan catatan bahwa interface-interface juga menunjukkan polimorfisme, dikarenakan program dapat memanggil method interface dan versi yang tepat dari method yang akan dieksekusi tergantung dari tipe object yang melewati pemanggil method interface

Abstract	Interface
Bisa berisi abstract dan non abstract Method.	Hanya boleh berisi abstract method
Kita harus menuliskan sendirimodifiernya.	Kita tidak perlu susah2 menulis public abstract di depan nama method. Karena secara implisit, modifier untuk method diinterface adalah public danabstract
Bisa mendeklarasikan constant dan instance variable.	Method tidak boleh bersifat static
Method boleh bersifat static	Method tidak boleh bersifat static
Method boleh bersifat final	Method tidak boleh bersifat final
suatu abstact class hanya bisa mengextend satu abstract class lainnya	Suatu interface bisa mengextend satu atau lebih interface lainnya
uatu abstact class hanya bisa mengextendsatu abstract class lainnya	Suatu interface hanya bisa mengextendinterface lainnya. Dan tidak bisa mengimplement class atau interface lainnya

#### 4. Kesimpulan

**Buatlah kesimpulan dari pembahasan praktikum yang telah Anda kerjakan.**

Pada praktikum modul 6 ini membahas tentang Abstrak class Abstract class (kelas abstrak) adalah sebuah konsep dalam pemrograman berorientasi objek yang digunakan untuk membuat kerangka kerja atau cetak biru bagi kelas-kelas turunannya. Abstract class tidak dapat diinstansiasi secara langsung, tetapi berfungsi sebagai kerangka kerja untuk kelas-kelas turunan yang mewarisi sifat dan perilaku dari abstract class tersebut.

Dalam sebuah abstract class, kita dapat mendefinisikan metode-metode abstrak yang harus diimplementasikan oleh kelas-kelas turunannya, serta metode non-abstrak yang dapat memiliki implementasi konkret. Abstract class juga dapat memiliki properti dan metode biasa seperti kelas pada umumnya.

## 5. Referensi

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

Modul praktikum