

VERSION 0.1
APRIL 20, 2018



PEMOGRAMAN BERORIENTASI OBJEK

ABSTRACT CLASS & INTERFACE. MODUL 5

DISUSUN OLEH: ASISTEN 2016

DIAUDIT OLEH:- YUFIZ AZHAR, S.KOM., M.KOM.

- MASKUR, S.KOM., M.KOM.

- YUDA MUNARKO, S.KOM., M.SC.

PRESENTED BY: TIM LAB-IT
UNIVERSITAS MUHAMMADIYAH MALANG

PEMOGRAMAN BERORIENTASI OBJEK

PERSIAPAN MATERI

Materi yang harus dipelajari sebelum mengerjakan tugas

A. ABSTRACT CLASS

Adalah class yang sifatnya abstrak yang biasanya digunakan untuk mendefinisikan konsep class yang sifatnya umum. Sebagai contoh jika ada class hewan, kucing dan kelinci. karena hewan sifatnya sangat umum, maka bisa dideklarasikan sebagai abstract class. Dalam implementasinya, abstract class tidak bisa dibuat objectnya, namun bisa dimanfaatkan dalam konsep polimorfisme. Di dalam abstract class bisa dideklarasikan variabel class, concrete method dan abstract method. Concrete method adalah method yang memiliki kode program / body program, artinya sudah diimplementasi. Sedangkan abstract method adalah method yang hanya berupa deklarasi method saja, yang terdiri dari nama method, jenis luaran, modifier dan input parameter tanpa kode program. Selanjutnya, untuk class yang merupakan turunan dari abstract class, jika itu adalah concrete class maka harus mengimplementasi abstract class (bentuknya seperti overriding), sedangkan jika abstract class juga maka bisa tidak mengimplementasi abstract class.

Percobaan

Buatlah class-class berikut dalam 1 package

Class abstract hewann

```
10  //  
11  public abstract class hewann {  
12      //deklarasi abstract method  
13      abstract void setName();  
14      abstract void setMakanan();  
15  }  
16
```

Class Kucing

```

10  */
11  public class kucing extends hewann {
12      public void setName() {
13          System.out.println("nama hewan adalah kucing");
14      }
15      public void setMakanan() {
16          System.out.println("makanan kucing adalah ikan");
17      }
18      public void setWarna() {
19          System.out.println("warna putih");
20      }
21  }
22

```

Class kelinci

```

11  public class kelinci extends hewann {
12      public void setName() {
13          System.out.println("nama hewan kelinci");
14      }
15      public void setMakanan() {
16          System.out.println("makanan kelinci adalah wortel");
17      }
18      public void setUkuran() {
19          System.out.println("Ukuran kecil");
20      }
21  }
22

```

Driver class

```

1  public class main {
2      public static void main(String[] args) {
3          kucing kcg = new kucing();
4          kcg.setName();
5          kcg.setMakanan();
6          kcg.setWarna();
7
8          kelinci klc = new kelinci();
9          klc.setName();
10         klc.setMakanan();
11         klc.setUkuran();
12     }
13 }

```

B. INTERFACE

Interface bukanlah class, meskipun deklarasinya menyerupai. Penamaan interface biasanya menggunakan kata sifat seperti Fireable, Rideable, Movable, dan lain-lain. Hal ini berhubungan dengan konsep interface yang digunakan untuk mendeklarasikan sifat-sifat tertentu. Sifat-sifat ini yang nantinya akan diimplementasi oleh class sesuai dengan kebutuhan. Sebagai contoh jika ada interface Fireable dan di dalamnya ada method shot() – menembak, Maka interface ini bisa diimplementasi oleh class Pistol, Tank, Bazooka, PesawatTempur yang memang memiliki kemampuan untuk menembak. Interface hanya memiliki abstract method saja, tidak ada yang concrete, sehingga pada saat menulis deklarasi method bisa tidak menggunakan keyword “abstract”, meskipun boleh juga menggunakan keyword “abstract”. Interface bisa juga memiliki variable, namun variable itu harus diberi nilai serta otomatis menjadi konstanta dengan modifier public, final dan static. Keyword yang digunakan untuk mengimplementasikan interface adalah “**implements**”.

Percobaan

Buatlah class-class berikut dalam 1 package

Interface Fireable

```

10  L  */
    ④  public interface Fireable {
    ⑤      public void shot();
13      }
14

```

Interface moveable

```

    ④  public interface moveable {
    ⑤      public void move();
13      }
14

```

Class Tank

```

11 public class Tank implements Fireable, moveable{
12     public void shot() {
13         System.out.println("tank mengebom bum bum");
14     }
15     public void move() {
16         System.out.println("tank bergerak menyerang");
17     }
18 }
19

```

Class pistol

```

11 public class pistol implements Fireable{
12     public void shot() {
13         System.out.println("pistol menembak dor dor");
14     }
15 }
16

```

Driver Class

```

11 public class driver {
12     public static void main(String[] args){
13         moveable m = new Tank();
14         Fireable f = new pistol();
15
16         m.move();
17         f.shot();
18         f = (Fireable) m;
19         f.shot();
20     }
21 }

```

Apa Itu Polimorfisme ?

Polimorfisme merupakan kemampuan objek-objek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama. Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan *method* mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.

TUJUAN

Mahasiswa memahami struktur dasar pemograman java yang meliputi :

1. Mampu memahami konsep Abstract Class dan Interface
2. Mampu memahami Abstract Class dan membuat implementasi Abstract Class
3. Mampu memahami Interface dan membuat implementasi Interface

TARGET MODUL

- Mahasiswa mampu menguasai konsep pemograman
- Mahasiswa mampu menggunakan notasi flowchart dengan benar

PERSIAPAN SOFTWARE/APLIKASI

1. Compiler java (JDK), JRE
2. Editor Java (Gel, Eclipse, Jcreator, dll)

TUGAS

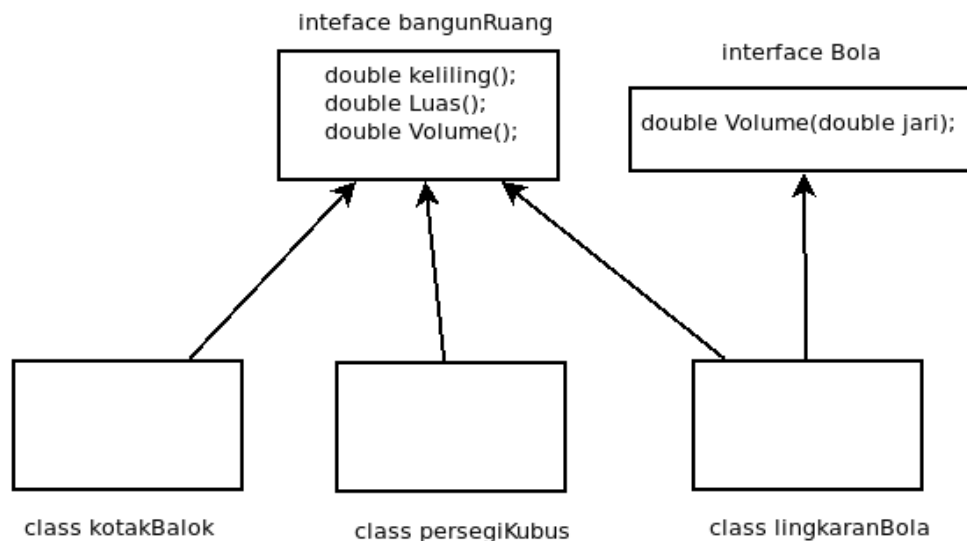
TUGAS 1

Buatlah 2 abstract method yaitu `getLuas` dan `getVolume` di dalam kelas `Abstract Shape`. Kemudian turunkan dua method ini kedalam kelas `Lingkaran` dan `Persegi` (ingat bahwa masing masing menerapkan rumus matematika sendiri). Kemudian buatlah 1 buah driver class untuk outputannya

Output :

```
run:
luas persegi:542.89
keliling persegi :8.8
Luas Lingkaran :555.4346
Keliling Lingkaran :20.096000000000004
BUILD SUCCESSFUL (total time: 0 seconds)
```

TUGAS 2



Buatlah sebuah kerangka program penghitungan keliling, luas dan volume sesuai dengan gambar yang telah diberikan di atas dengan syarat :

1. Class `kotakbalok`, `persegiKubus` dan `lingkaranBola` mengimplementasikan method yang ada pada interface `bangunRuang`
2. Khusus class `lingkaranBola` mengimplementasikan 2 interface diatas untuk menghitung volume bola dengan parameter jari-jari yang telah diberikan

KRITERIA & DETAIL PENILAIAN

Soal	Poin
Tugas 1	50
Tugas 2	50