laporan praktikum

Pemrogramam berorientasi objek

Modul : 6 Tanggal praktikum : 17-05-2023 *(Format: DD-MM-YYYY)*

Nama : Muhamad Fahmi

NIM : 20220810029

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

# jUDUL/TEMA MODUL

## Tujuan Pembelajaran

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

1. Memahami Polimorphism
2. Memahami Konsep Polimorphism
3. Mampu Membuat Program Java Polimorphism

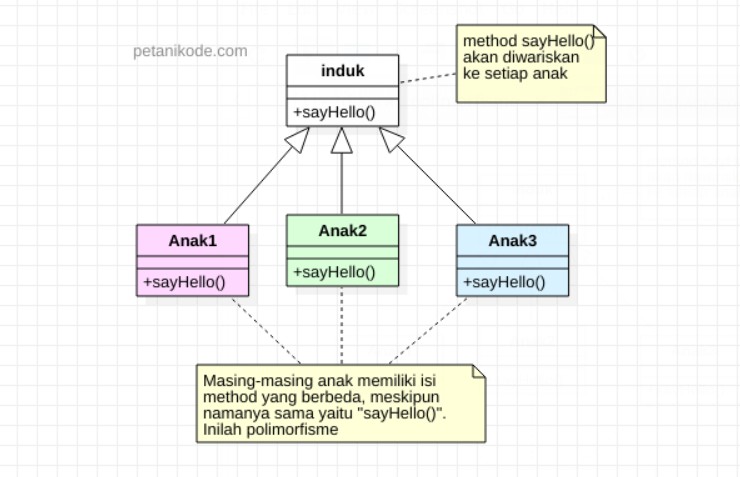
## Dasar Teori

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

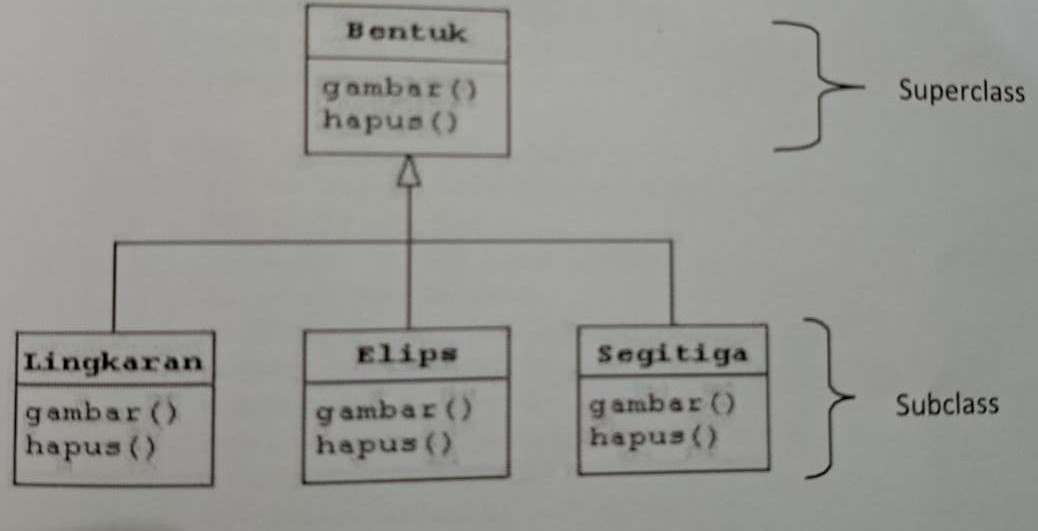
### Polymorphism

Polymorphism mempunyai makna sesuatu yang memiliki banyak bentuk, yaitu memiliki nama sama, tetapi memiliki kelakuan (behaviour) yang berbeda.

Poly artinya banyak, morfisme artinya bentuk. Polimorfisme (bahasa inggris polymorphism) adalah sebuah prinsip dalam biologi di mana oraganisme atau spesias dapat memiliki banyak bentuk atau tahapan (Muhardian, 2019).



Contoh Gambar Polimoriphism diambil dari (Muhardian, 2019).



Perhatikan gambar diagram diatas!

Class Bentuk meupakan class induk (*superclass*) dari class Lingkaran, Elips, dan Segitiga mempunyai method **gambar()** dan **hapus()**. Class-class anak (*subclass*) juga mempunyai method **gambar()** dan **hapus()**. Meskipun keempat class tersebut mempunyai nama method yang sama, tetapi isi (source code / code yang dilakukan / output) dari masing-masing method tersebut berbeda.

Jika kita menginginkan sebuah objek yang dapaat memanggil setiap method (yaitu method gambar & hapus) yang ada pada setiap class (pada superclass maupun subclass), maka gunakanlah teknik **Polymorphism**. Polymorphism hanya berlaku pada method dan tidak berlaku pada atribut.

Untuk mendapatkan operasi polymorphism dari suatu method, maka method tersebut haruslah merupakan method yang ada di class induk (lihat diagram diatas bahwa method gambar() & hapus(), selain terdapat class-class turunan class Bentuk, juga terdapat di class Bentuk).

## 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. |

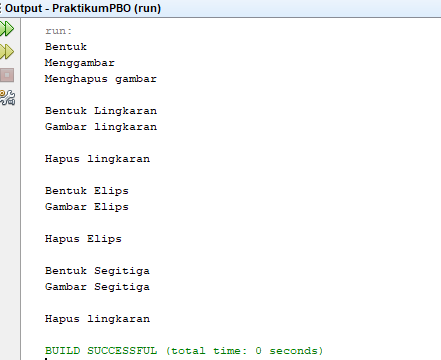
PRETEST

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

1. PRAKTIKUM

PRAKTIKUM 1

* 1. Buatlah Project Baru di Netbeans dengan nama modul5PBO
  2. Buat File Empty Java File Baru
  3. Ketikkan Source Code berikut ini dan simpan dengan nama file **TesBentuk.java**
  4. Jalankan file **TesBentuk.java** tersebut, amati hasilnya

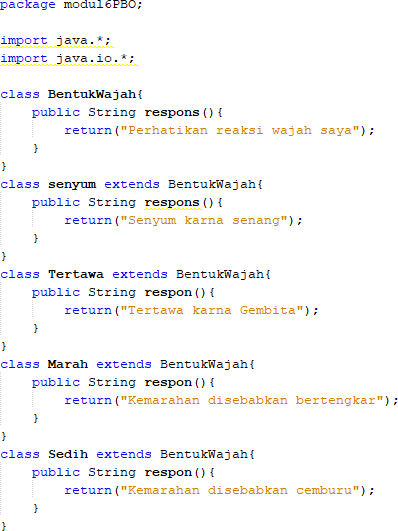


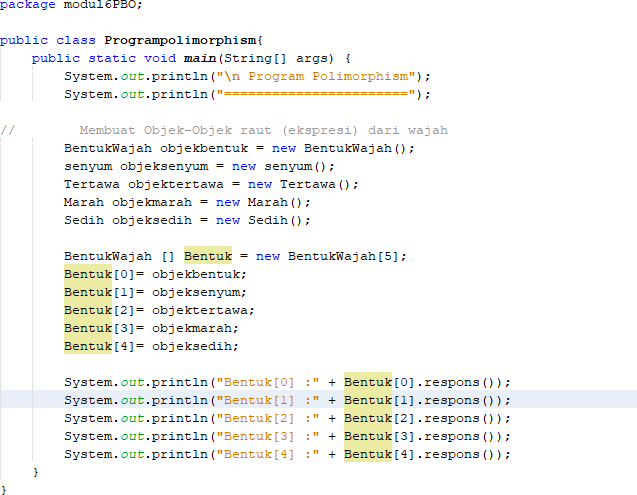
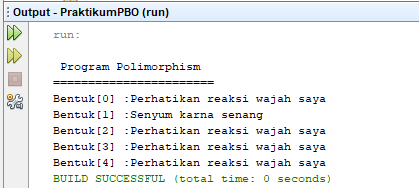
Pembahasan:

Pada praktikum 1 ini hasil dari program ini penerapan Polimorphism dimana Bentuk sebagai superclass dan lingkaran, elips, dan segitiga sebagai subclass pada subclass ditandai dengan ***extends.*** Lalu dicetak pada class TesBentuk

PRAKTIKUM 2

1. Buat File Empty Java File Baru dengan nama **BentukWajah.java**
2. Ketikkan Soucre Code berikut ini:



1. Buat File Empty Java File Baru dengan nama **Programpolimorphism.java**
2. Ketikkan Source Code Berikut ini:
3. Jalankan file **Programpolimorphism.java** tersebut, amati hasilnya

Pembahasan:

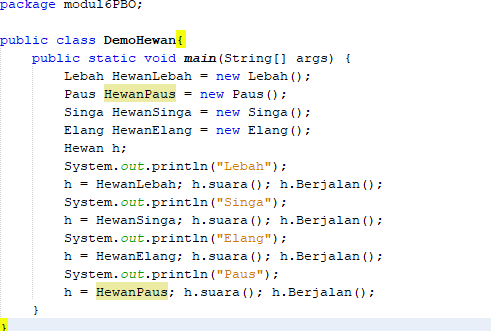
Pada praktikum 2 ini hasil dari program ini penerapan Polimorphism dimana BentukWajah sebagai superclass dan senyum, tertawa, marah, dan sedih sebagai subclass pada subclass ditandai dengan ***extends.*** Lalu dicetak pada file ProgramPolimorphism.java

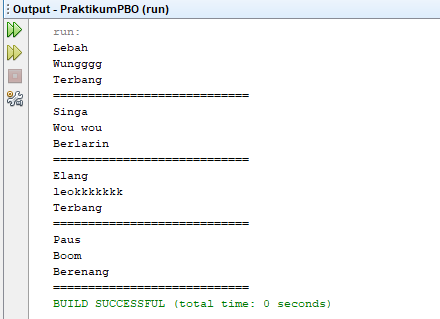
PRAKTIKUM 3

1. Buat File Empty Java File Baru dengan nama **Hewan.java**
2. Ketikkan Soucre Code berikut ini:



1. Buat File Empty Java File Baru dengan nama **DemoHewan.java**
2. Ketikkan Source Code Berikut ini:



1. Jalankan file **DemoHewan.java** tersebut, amati hasilnya

Pembahasan:

Pada praktikum 3 ini hasil dari program ini penerapan Polimorphism dimana Hewan sebagai superclass dan Singa, Elang, Lebah, dan Paus sebagai subclass pada subclass ditandai dengan ***extends.*** Lalu dicetak pada file DemoHewan.java

**POSTTEST**

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

import java.\*; import java.io.\*;

/\*class induk dengan nama PabrikanMotor\*/ class PabrikanMotor{

//Membuat sebuah method respons pada class induk dimana mengembalikan

public String respons()

{return("Top speed pabrikan motor\n");

}

}

/\*Membuat class-class lain dengan turunan dari class pabrikan motor\*/

class Yamaha extends PabrikanMotor{ public String respons()

{return("299 km/jam \n");}

}

class Honda extends PabrikanMotor{ public String respons()

{return("301 km/jam \n");}

}

class Suzuki extends PabrikanMotor{ public String respons()

{return("299 km/jam \n");}

}

class Kawasaki extends PabrikanMotor{ public String respons()

{return("329 km/jam \n");}

}

Programpolimorphism2.java

public class Programpolimorphism2{

public static void main(String []args)

{

System.out.println("\n Program polimorphs"); System.out.println("====================="); PabrikanMotor objpabrikan=new PabrikanMotor(); Yamaha objyamaha=new Yamaha();

Honda objhonda=new Honda(); Suzuki objsuzuki=new Suzuki();

Kawasaki objkawasaki=new Kawasaki(); PabrikanMotor []Pabrikan=new PabrikanMotor[5]; Pabrikan[0]=objpabrikan; Pabrikan[1]=objyamaha;

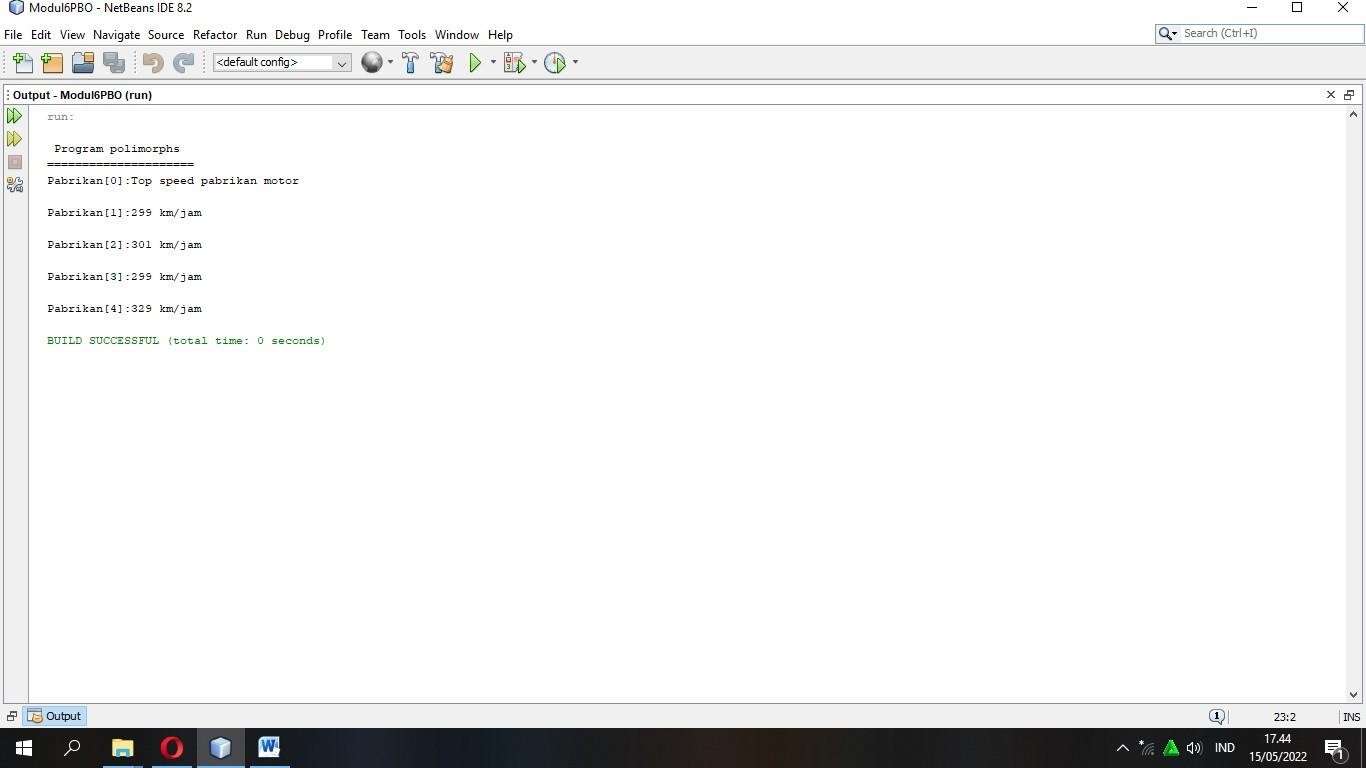
Pabrikan[2]=objhonda; Pabrikan[3]=objsuzuki; Pabrikan[4]=objkawasaki;

System.out.println("Pabrikan[0]:"+Pabrikan[0].respons()); System.out.println("Pabrikan[1]:"+Pabrikan[1].respons()); System.out.println("Pabrikan[2]:"+Pabrikan[2].respons()); System.out.println("Pabrikan[3]:"+Pabrikan[3].respons());

System.out.println("Pabrikan[4]:"+Pabrikan[4].respons());

}

}



**TUGAS INDIVIDU**

* 1. Apa perbedaan Inheritence dengan Polimorphism, Jelaskan Inheritance (Pewarisan)

Inheritance (penurunan sifat / pewarisan), ini merupakan ciri khas dari OOP yang tidak terdapat pada pemograman procedural gaya lama. Dalam hal ini, inheritance bertujuan membentuk obyek baru yang memiliki sifat sama atau mirip dengan obyek yang sudah ada sebelumnya (pewarisan)

Sedangkan

Polymorphism, suatu aksi yang memungkinkan pemrogram menyampaikan pesan tertentu keluar dari hirarki obyeknya, dimana obyek yang berbeda memberikan tanggapan / respon terhadap pesan yang sama sesuai dengan sifat masing-masing obyek.

Atau Polymorphism dapat berarti banyak bentuk, maksudnya yaitu dapat menimpa (override), suatu method, yang dimana object tersebut diturunkan, sehingga memiliki kelakuan yang berbeda.

## Kesimpulan

|  |
| --- |
| Buatlah kesimpulan dari pembahasan praktikum yang telah Anda kerjakan. |

Pada praktikum modul 6 ini membahas tentang polymorphism dalam program java.yaitu memiliki nama sama,tetapi memiliki kelakuan (behavior) yang berbeda.

## Referensi

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

## Modul praktikum