

MODUL 5

KELAS DAN OBJEK

1. Class

Class merupakan cetak biru (blue print) dari objek atau dengan kata lain sebuah class menggambarkan ciri-ciri objek secara umum.

Sebagai contoh Yamaha MX, Suzuki Smash, Honda Supra Fit, dan Kawasaki Ninja merupakan objek dari class sepeda motor. Objek-objek tersebut mempunyai kesamaan atribut (merk, tipe, berat, kapasitas bensin, tipe mesin, warna, harga, dll) dan method untuk mengakses data pada atributnya (misal fungsi untuk menginput data merk, tipe, berat, dsb serta fungsi untuk mencetak data merk, tipe, berat, dsb)

Contoh :

```
class SepedaMotor{
private String merk, tipe;
private int tangki;
private long harga;
public void inputMerk (String merk){
    this.merk = merk;
}
public String tampilMerk(){
    return merk;
}
}
```

2. Objek

Objek merupakan segala sesuatu yang ada di dunia ini, yaitu manusia, hewan, tumbuhan, rumah, kendaraan, dan lain sebagainya. Contoh objek yang telah disebutkan di atas merupakan contoh objek nyata pada kehidupan kita. Pada pemrograman berorientasi objek, kita akan belajar bagaimana membawa konsep objek dalam kehidupan nyata menjadi objek dalam dunia pemrograman.

Setiap objek dalam dunia nyata pasti memiliki 2 elemen penyusunnya yaitu keadaan (state) dan perilaku/sifat (behaviour).

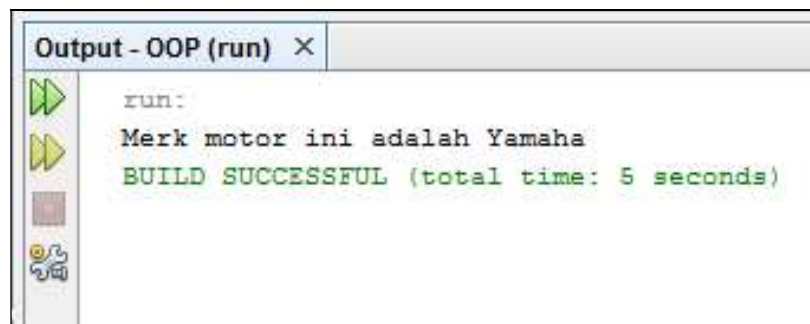
Sintak untuk membuat objek :

```
namaClass namaObjek = new namaClass ()
```

Class utama dari program :

```
public class Yamaha {  
    public static void main(String[] args) {  
        SepedaMotor motor = new SepedaMotor();  
        motor.inputMerk("Yamaha");  
        System.out.println("Merk motor ini adalah "+motor.tampilMerk());  
    }  
}
```

HASIL :



3. Variabel Static dan Variabel Instan

Variabel static adalah variabel yang didefinisikan/dideklarasikan dengan menggunakan keyword “static”, contoh :

```
static int jumlah ;
```

Variabel-variabel yang dideklarasikan dengan tidak menggunakan keyword “static”, maka variabel tersebut disebut dengan variabel instant.

Jika sebuah variabel merupakan variabel instant, maka masing-masing objek dari class tersebut akan memiliki variabel yang sama dengan variabel instant di satu objek tidak akan berpengaruh pada variabel instant di objek yang berbeda.

Jika sebuah variabel merupakan variabel static (pada suatu class), maka variabel static tersebut adalah variabel yang sama di semua objek dari class tersebut. Sehingga perubahan nilai pada variabel static tersebut di suatu objek akan berpengaruh juga terhadap objek yang lainnya.

4. Input Data

Untuk menginputkan data dari keyboard ada 2 cara, yaitu :

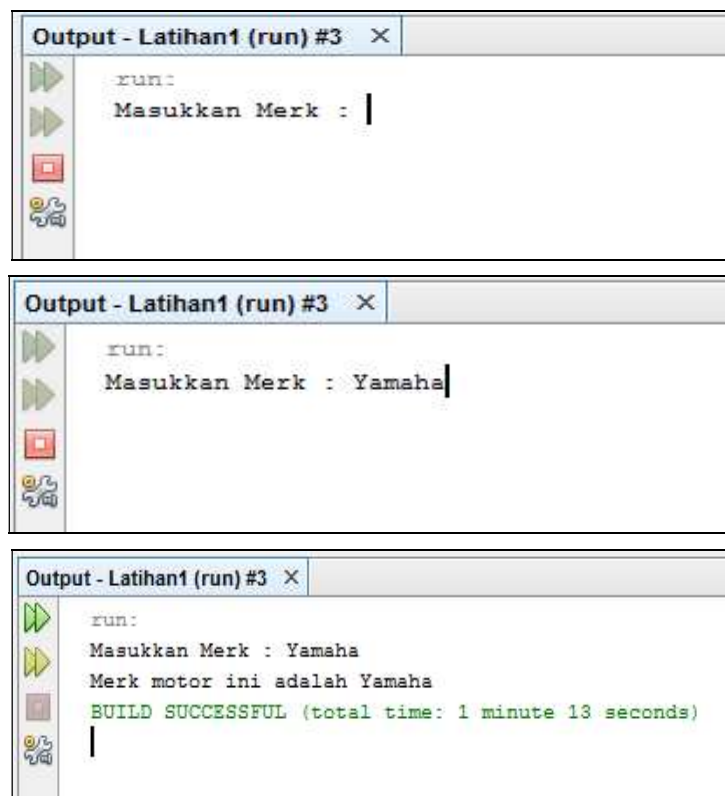
- Input dari mode console yaitu dengan memanfaatkan class **BufferedReader** dan **InputStreamReader**.

Untuk dapat mengakses class **BufferedReader** maka perlu mengimpor dari package **java.io.*** dan menambahkan statemen **throws IOException** pada header method main.

Contoh :

```
import java.io.*;

public class InputMerk1    {
    public static void main(String[] args) throws IOException{
        BufferedReader input = new BufferedReader (
            new InputStreamReader(System.in));
        String merk;
        System.out.print("Masukkan Merk : ");
        merk = input.readLine();
        System.out.println("Merk motor ini adalah " + merk);
    }
}
```

HASIL :

- Inputan dengan memanfaatkan class **JOptionPane**.

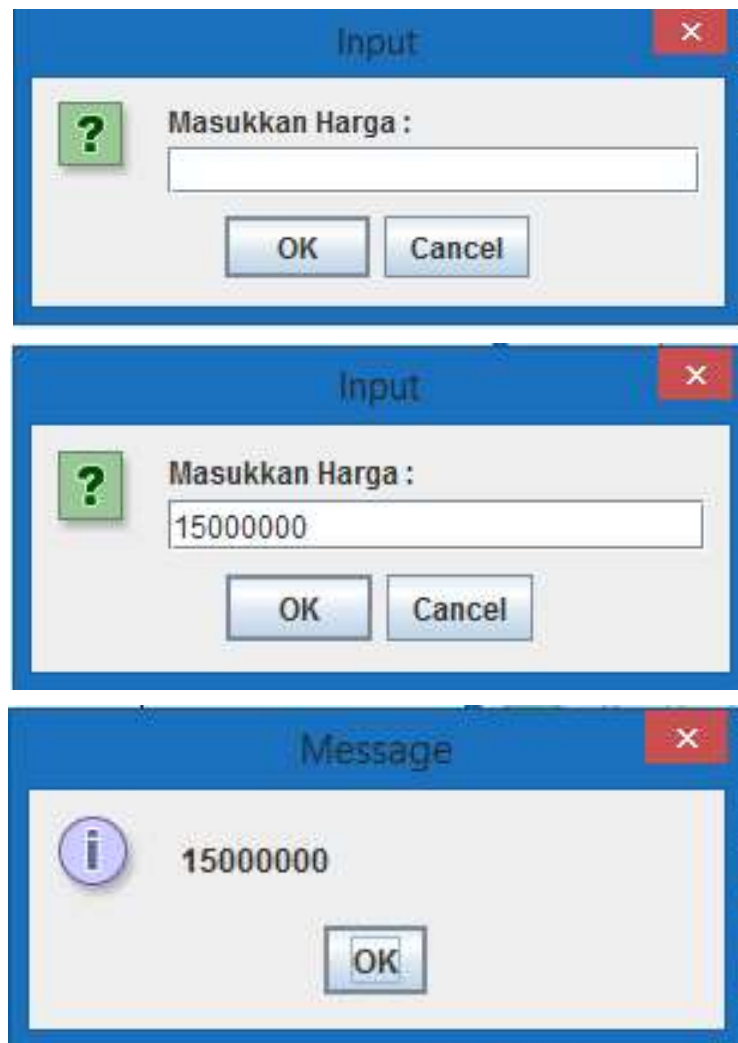
Untuk bisa menggunakan class **JOptionPane**, maka perlu mengimport dari package **javax.swing.*** dan gunakan method **showInputDialog()** yang terdapat pada class **JOptionPane**.

Contoh :

```
import javax.swing.*;

public class InputMerk2 {
    public static void main(String[] args) {
        String input;
        input = JOptionPane.showInputDialog("Masukkan Harga : ");
        long harga = Integer.valueOf(input).intValue();
        JOptionPane.showMessageDialog(null, harga);
    }
}
```

HASIL :



Catatan :

Wrapper	Conversion Method
Double	static double <code>parseDouble(String str)</code> throws <code>NumberFormatException</code>
Float	static float <code>parseFloat(String str)</code> throws <code>NumberFormatException</code>
Long	static long <code>parseLong(String str)</code> throws <code>NumberFormatException</code>
Integer	static int <code>parseInt(String str)</code> throws <code>NumberFormatException</code>
Short	static short <code>parseShort(String str)</code> throws <code>NumberFormatException</code>
Byte	static byte <code>parseByte(String str)</code> throws <code>NumberFormatException</code>

Sumber : *Herbert Schildt - A Beginner's Guide, 3rd Edition (2005;396-397)*

5. Latihan Satu

Buatlah program pertambahan sederhana seperti dibawah ini.



KODE PROGRAM :

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

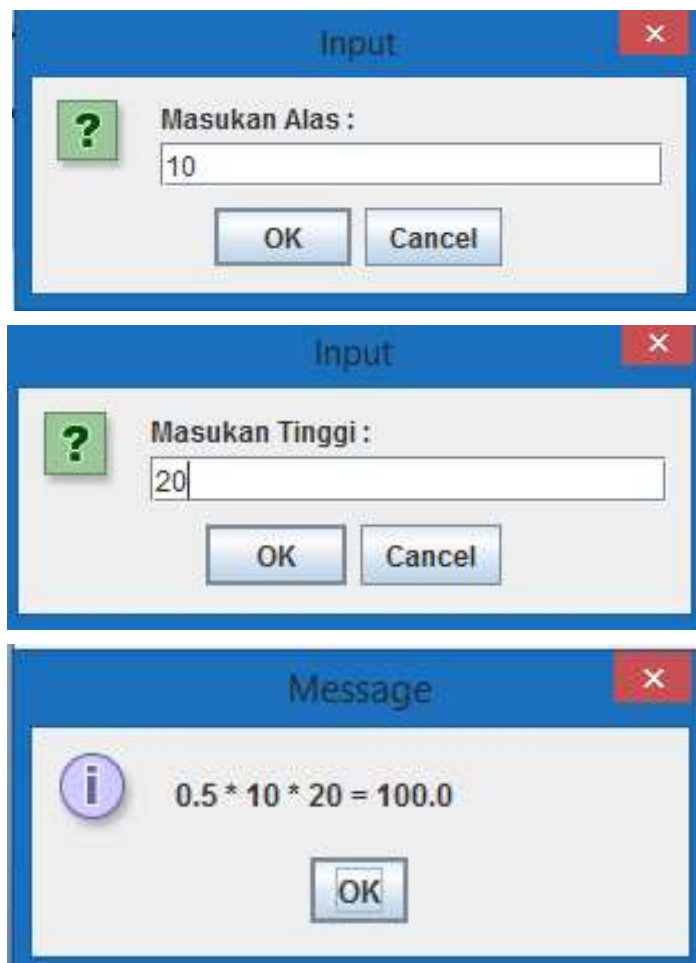
6. Latihan Dua

Buatlah kelas baru dengan nama segitiga kemudian lengkapi kode program berikut :

```
import javax.swing.*;
class segitiga {
String alas,tinggi;
public void input(){
    ....
    ....
}
public double hitung(){
    long a = Long.parseLong(alas);
    long t = Long.parseLong(tinggi);
    return (0.5*a*t);
}
```

```
public void hasil(){
    JOptionPane.showMessageDialog(null,"0.5 * " + .... + " * " + ....
+ " = " + .... );
}
}
class luas_segitiga{
public static void main(String[] args) {
    ....
    ....
    ....
    ....
}
}
```

HASIL :



MODUL 6

ENCAPSULATION

Salah satu keistimewaan java adalah pengkapsulan. Pengkapsulan adalah mengkombinasikan suatu struktur dengan fungsi dalam satu kerangka yaitu kelas(*class*).

Kelas akan menutup rapat baik data maupun kode. Akses item di dalam kelas dikendalikan. Pengendalian ini tidak hanya berupa data tetapi juga kode. Saat kelas akan digunakan, kelas harus sudah dideklarasikan. Yang penting, pemakai kelas mengetahui deskripsi kelas, tetapi bukan implementasinya. Bagi pemakai, detail internal kelas tidak penting. Konsep ini disebut penyembunyian informasi (*information hiding*).

Untuk menggunakan kelas, kita perlu mengetahui sesuatu tentangnya. Kita perlu mengetahui fungsi apa yang bisa digunakan dan data apa saja yang dapat diakses. Fungsi yang dapat digunakan dan data yang dapat diakses disebut antarmuka pemakai (*userinterface*). Antarmuka pemakai menceritakan bagaimana kelas berperilaku, bukan bagaimana kelas dibuat. Kita tidak perlu mengetahui implementasi kelas. Sekali kelas dibuat, kita bisa memakainya berulang-ulang. Bagi pandangan pemakai, kelas adalah kotak hitam dengan perilaku tertentu.

1. Kendali Akses terhadap Kelas

Tugas kelas adalah untuk menyembunyikan informasi yang tidak diperlukan oleh pemakai. Ada tiga macam pemakai kelas:

- kelas itu sendiri
- pemakai umum
- kelas turunan

Setiap macam pemakai mempunyai hak aksesnya masing-masing.

2. Hak akses

Kelas pada java menawarkan tiga hak akses untuk anggota kelas (baik anggota data maupun fungsi anggota), yaitu :

- Private

Tingkat akses ini berguna untuk memberikan hak akses data hanya kepada kelas yang bersangkutan saja. Artinya kelas-kelas turunan ataupun lingkungan luar di dalam program tidak diizinkan untuk mengakses data tersebut. Dalam java untuk menentukan data tersebut bersifat privasi, maka harus menggunakan kata kunci **private**. Secara default, jika tidak menuliskan tingkat akses dalam pendeklarasian data dalam sebuah kelas maka data-datanya akan dianggap data private.

- Public

Tingkat akses ini dapat diakses oleh fungsi anggota kelas itu sendiri, lingkungan luar dan fungsi anggota kelas turunan. Suatu kelas agar bisa diakses dari luar kelas, misalnya dalam fungsi `main()`, perlu mempunyai hak akses publik. Hak akses ini yang biasanya digunakan sebagai perantara antara

kelas dengan dunia luar. Dalam java untuk menentukan data tersebut bersifat umum, maka harus menggunakan kata kunci **public**.

- Protected

Tingkat akses ini digunakan untuk memberikan hak akses terhadap data dalam suatu kelas sehingga data tersebut dapat diakses oleh kelas turunannya, namun lingkungan luar di dalam program masih tetap tidak diberi hak untuk mengaksesnya. Dalam java untuk menentukan data tersebut bersifat diproteksi, maka harus menggunakan kata kunci **protected**.

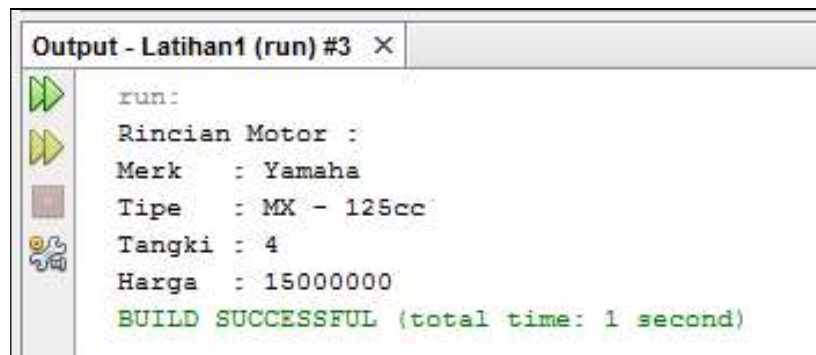
Contoh :

```
class SepedaMotor{
private String merk, tipe;
private int tangki;
private long harga;
public void inputMerk (String merk){
    this.merk = merk;
}
public void inputTipe (String tipe){
    this.tipe = tipe;
}
public void inputTangki(int tangki){
    this.tangki = tangki;
}
public void inputHarga (long harga){
    this.harga = harga;
}
public String tampilMerk(){
    return merk;
}
public String tampilTipe(){
    return tipe;
}
public int tampilTangki(){
    return tangki;
}
public long tampilHarga(){
```

```
        return harga;
    }

    public static void main(String[] args) {
        SepedaMotor motor = new SepedaMotor();
        motor.inputMerk("Yamaha");
        motor.inputTipe("MX - 125cc");
        motor.inputTangki(4);
        motor.inputHarga(15000000);
        System.out.println("Rincian Motor :");
        System.out.println("Merk    : " + motor.tampilMerk());
        System.out.println("Tipe    : " + motor.tampilTipe());
        System.out.println("Tangki  : " + motor.tampilTangki());
        System.out.println("Harga   : " + motor.tampilHarga());
    }
}
```

HASIL :



```
Output - Latihan1 (run) #3 X
run:
Rincian Motor :
Merk    : Yamaha
Tipe    : MX - 125cc
Tangki  : 4
Harga   : 15000000
BUILD SUCCESSFUL (total time: 1 second)
```

3. Latihan

Implementasikan encapsulation kedalam kelas barang, dengan jenis barang yang diambil adalah buku gambar. Yang mana memiliki Kode, Nama, HPP, Harga Jual, dan Stok. Ketika dilakukan transaksi penjualan terdapat atribut lain yaitu Jumlah Jual. Yang diminta pada kasus ini adalah memunculkan Stok Sekarang, Total Bayar, dan Laba. Proses yang dibutuhkan yaitu input, hitung stok, hitung total, hitung laba, dan tampil.



```
run:
INPUT:
Kode Barang      :B001
Nama Barang      :BUKU GAMBAR
HPP              :2500
Harga Jual       :3000
Stok             :5
Jumlah Jual      :2
OUTPUT:
Stok Sekarang    :3
Total Bayar      :6000
Laba             :1000

BUILD SUCCESSFUL (total time: 2 seconds)
```

This image shows a full page of white paper with horizontal dashed lines, typical of primary school handwriting practice paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

PEMROGARAMAN VISUAL

Dosen : BEBEN SUTARA, S.Kom., M.T

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

MODUL 7

CONSTRUCTOR DAN INHERITANCE

1. Konstruktor

Constructor adalah method yang secara otomatis dipanggil/dijalankan pada saat sebuah class diinisiasi. Jika dalam sebuah class tidak terdapat constructor maka secara otomatis Java akan memuatkan sebuah default constructor. Nama constructor harus sama dengan nama class dan tidak boleh memiliki tipe *return value*. Sama halnya dengan method, constructor dapat memiliki satu atau banyak parameter maupun tanpa parameter.

Constructor biasanya digunakan untuk memberi nilai awal dari atribut-atribut class tersebut.

Contoh :

```
class Login {  
    private String user, pass;  
    Login() {  
        this.user = "";  
        this.pass = "";  
    }  
}
```

2. Multiple Constructor

Java tidak membatasi jumlah constructor dalam satu class, sehingga memungkinkan sebuah class memiliki lebih dari satu constructor. Multiple constructor adalah adanya lebih dari satu constructor untuk sebuah class. Yang membedakan antara satu constructor dengan constructor lainnya adalah pada parameternya (nama constructornya sama)

Contoh :

```
class Login {  
    private String user, pass;  
    Login() {  
        this.user = "";  
        this.pass = "";  
    }  
    Login (String user, String pass) {  
        this.user = user;  
        this.pass = pass;  
    }  
}
```

```
public void inputUser(){
    this.user = user;
}
public void inputPass(){
    this.pass = pass;
}
}
```

3. Inheritance

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. 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, 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)
```

Contoh inheritance :

```
class Pelajar {
private String nama;
private long tinggi;
private long berat;
```

```
public Pelajar (String nama, long tinggi, long berat){
    this.nama = nama;
    this.tinggi = tinggi;
    this.berat = berat;
}
public String tampilPelajar(){
    return ("Nama\t: " + nama + "\nTinggi\t: " + tinggi +
"\nBerat\t: "+ berat +"\n");
}
}
class Siswa extends Pelajar {
    private String nim, asalsekolah;
    private long nilai;
    public Siswa (String nama, long tinggi, long berat,
        String nim, String asalsekolah, long nilai) {
        super(nama,tinggi,berat);
        this.nim = nim;
        this.asalsekolah = asalsekolah;
        this.nilai = nilai;
    }
    public String tampilSiswa(){
        return (super.tampilPelajar()+"nim\t: " + nim + "\nAsal Sekolah
:" + asalsekolah + "\nNilai\t:"+ nilai);
    }
}
class main {
    public static void main(String[] args) {
        Siswa ti = new Siswa ("Ahmad", 180,78,"1390192","SMK
Informatika",90);
        System.out.println(ti.tampilSiswa());
    }
}
```

Catatan :

Escape Sequence	Description
\'	Single quote
\"	Double quote
\\	Backslash
\r	Carriage return
\n	New line
\f	Form feed
\t	Horizontal tab
\b	Backspace
\ddd	Octal constant (where ddd is an octal constant)
\uxxxx	Hexadecimal constant (where xxxx is a hexadecimal constant)

Table 2-2 Character Escape SequencesSumber : *Herbert Schildt - A Beginner's Guide, 3rd Edition (2005;37)***HASIL :**

```
run:
Nama      : Ahmad
Tinggi    : 180
Berat     : 78
nim       : 1390192
Asal Sekolah : SMK Informatika
Nilai     : 90
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Latihan

Implementasikan inheritance kedalam kelas barang, dengan jenis barang yang diambil adalah masih buku gambar. Ketentuan sama namun untuk proses input dibuat dalam bentuk constructor . Proses yang dibutuhkan yaitu hitung stok, hitung total, hitung laba, dan tampil.

Keterangan :

- Barang (Super Class) = **Kode** dan **Nama**
- Buku Gambar (Sub Class) = **Kode**, **Nama**, HPP, Harga Jual, dan Stok



```

run:
INPUT:
Kode Barang      :B001
Nama Barang      :BUKU GAMBAR
HPP              :2500
Harga Jual       :3000
Stok             :5
Jumlah Jual      :2
OUTPUT:
Stok Sekarang    :3
Total Bayar      :6000
Laba             :1000

BUILD SUCCESSFUL (total time: 2 seconds)

```

This image shows a full page of white paper with horizontal dashed lines, typical of primary-ruled notebook paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

PEMROGARAMAN VISUAL

Dosen : BEBEN SUTARA, S.Kom., M.T

This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

MODUL 8

POLYMORPHISME

1. Pengertian Polymorphisme

Salah satu konsep Pemrograman Berorientasi Objek adalah polymorfisme yaitu kemampuan beberapa objek bertipe sama bereaksi secara berbeda terhadap “pesan” yang sama.

Contoh :

```
class Manusia {
    public String nama, hobby, pekerjaan;
    public Manusia (String nm, String hobby, String kerja) {
        this.nama= nm;
        this.hobby = hobby;
        this.pekerjaan = kerja;
    }
    public String tampilManusia (){
        return ("Nama\t\t: " + nama + "\n"+
               "Hobby\t\t: " + hobby + "\n"+
               "Pekerjaan\t: " + pekerjaan + "\n");
    }
}

class programmer extends Manusia {
    public programmer(String nm, String hobby, String kerja) {
        super(nm, hobby, kerja);
    }
}

class polisi extends Manusia {
    public polisi(String nm, String hobby, String kerja) {
        super(nm,hobby,kerja);
    }
}
```

```
class kegiatan {  
public static void main(String args[]) {  
Manusia[] informasi= new Manusia[2];  
informasi[0]=new programmer("Programmer", "Futsal", "Mengetik  
Sintak");  
informasi[1]=new polisi("Polisi", "Fusal", "Operasi keliling");  
for (int i=0; i<2; i++) {  
System.out.println("-----");  
System.out.println(informasi[i].tampilManusia());  
    }  
}  
}
```

HASIL :

```
Output - manusia (run) X  
run:  
-----  
Nama          : Programmer  
Hobby         : Futsal  
Pekerjaan     : Mengetik Sintak  
-----  
Nama          : Polisi  
Hobby         : Fusal  
Pekerjaan     : Operasi keliling  
BUILD SUCCESSFUL (total time: 1 second)
```

2. Latihan

Implementasikan konsep polymorphisme kedalam kelas barang, dengan jenis barang yang diambil adalah buku gambar dan pensil. Ketentuan sama namun untuk proses input dibuat dalam bentuk constructor . Proses yang dibutuhkan yaitu hitung stok, hitung total, hitung laba, dan tampil. Ubahlah latihan sebelumnya menjadi berkonsep polymorphisme.

Keterangan :

- Barang (Super Class) = **Kode** dan **Nama**
- Buku Gambar dan Pensil 2B (Sub Class) = **Kode**, **Nama**, HPP, Harga Jual, dan Stok

HASIL :

```
Output - barang (run) X
Kode      : B001
Nama      : BUKU GAMBAR
HPP       : 2500
Harga Jual : 3000
Stok      : 5
Jumlah Jual : 2
OUTPUT:
Stok Sekarang : 3
Total Bayar   : 6000
Laba          : 1000
-----
INPUT:
Kode      : P002
Nama      : PENSIL 2B
HPP       : 1500
Harga Jual : 1750
Stok      : 10
Jumlah Jual : 1
OUTPUT:
Stok Sekarang : 9
Total Bayar   : 1750
Laba          : 250
BUILD SUCCESSFUL (total time: 1 second)
```

KODE PROGRAM :

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

PEMROGARAMAN VISUAL

Dosen : BEBEN SUTARA, S.Kom., M.T

[illegible]