

## PRAKTIKUM I Class dan Obyek

Obyek merupakan dasar dari struktur Pemograman Berorientasi Obyek (OOP). Fitur dari OOP adalah Abstraksi, Enkapsulasi, Pewarisan, dan Polymorphism.

Fitur Abstraksi yaitu mengkarakteristikan obyek yaitu dengan pemberian atribut pada suatu obyek, hal ini akan diimplementasikan dengan pembuatan kelas (class). Kelas mempunyai 2 komponen yaitu Atribut dan Method.

### Pendefiniasian Class

```
<modifier> class <name> {
    <attributeDeclaration>*
    <constructorDeclaration>*
    <methodDeclaration>*
}
```

### PRAKTIKUM 1

```
class SepedaMotor
{
    private String merk, tipe;
    private int tangki;
    private long harga;

    public void setMerk(String merk)    {
        this.merk = merk;
    }
    public String getMerk(){
        return merk;
    }
}

class Latihan1a
{
    public static void main (String
[]args){
        SepedaMotor    motor    =    new
SepedaMotor();
        motor.setMerk("Suzuki");

        System.out.println("Motor    ini
bermerk " +
        motor.getMerk());
    }
}
```

TEKNIK INFORMATIKA  
PRAKTIKUM 2

Buatlah Kelas manusia mempunyai atribut : nama. Selain itu kelas manusia juga mempunyai method:

tampilkanNama, kerja, makan.

```
public class manusia
{
public String nama;
public manusia(String n)
{
this.nama = n;
}
public String tampilkanNama()
{
return nama;
}
public void makan()
{
System.out.println("Nyam... nyam... nyam...");
}
public void kerja()
{ System.out.println("Kerja...kerjaa.");
}
}
```

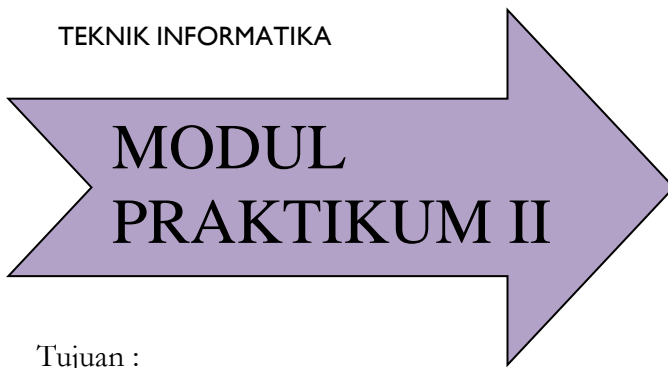
## TEKNIK INFORMATIKA

Adapun kode untuk menginstantiasi kelas manusia menjadi objek Ana yang mengimplementasikan method: tampilkanNama dan makan adalah sebagai berikut.

```
class praktikum2
{
public static void main(String arg[])
{
manusia ana= new manusia("Ana");
System.out.println("Nama= "+ ana
tampilkanNama()); ana.makan();
}
}
```

## TUGAS PRAKTIKUM 1

Buatlah 3 objek untuk menginstantiasi kelas mahasiswa dengan atribut nama dan nim Anda sendiri dan 2 teman Anda dan mengimplementasikan beberapa method yang telah Anda definisikan dalam kelas mahasiswa.



Tujuan :

- Mampu membuat Method Konstruktur.
- Mampu membuat Method Overloading pada konstruktur

## 1. KONSTRUKTOR

Konstruktur adalah bagian dari definisi suatu kelas yang berfungsi menciptakan instans dari kelas tersebut. Konstruktur ini bisa kita buat sendiri, atau bila kita tidak mendefinisikannya, maka kompilator Java akan membuatkan konstruktur default untuk kelas tersebut pada saat kompilasi. Yang perlu diperhatikan adalah bahwa suatu konstruktur tidak termasuk anggota suatu kelas seperti metode dan variabel dan bahwa konstruktur bisa dibuat lebih dari satu.

Metode konstruktur ini akan dipanggil secara otomatis oleh java ketika **new** dipakai untuk menciptakan instant kelas. Konstruktur mempunyai sifat:

- a. Namanya sama dengan nama kelas
- b. Tidak memiliki nilai balik ( termasuk tidak boleh ada kata-kunci void).
- c. Dipanggil dengan *keyword* **new**

Contoh Program Konstruktur 1

```
class mobil
{
    private String warna;
    private int tahunproduksi;
    //DEKLARASI KONSTRUKTOR
    public mobil (String warna, int tahunproduksi)
    {
        this.warna = warna;
        this.tahunproduksi = tahunproduksi;
    }

    // METODE
    public void info ()
    {
        System.out.println("Warna Mobil :" + this.warna);
        System.out.println("Tahun Produksi Mobil:" + this.tahunproduksi);
    }
}

public class Konstruktur {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        mobil mobilku = new mobil ("Ungu", 2010);
        mobilku.info();
    }
}
```

```
class tampilNilai{
    private String nilai;
    private String kategori;

    public tampilNilai(char huruf){
        switch (huruf){
            case 'A' :
                this.nilai="antara 80 sampai 100";
                this.kategori="Istimewa";
                break;
            case 'B':
                this.nilai="antara 65 sampai 79";
                this.kategori="Baik";
                break;
            case 'C':
                this.nilai="antara 56 sampai 64";
                this.kategori="Cukup";
                break;
            case 'D':
                this.nilai="antara 40 sampai 55";
                this.kategori="Kurang";
                break;
            case 'E':
                this.nilai="antara 0 sampai 39";
                this.kategori="Buruk";
                break;
        }
    }
    public void info(){
        System.out.println("Nilaiiku : "+this.nilai);
        System.out.println("Kategori: "+this.kategori);
    }
}

public class Konstruktor2 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        tampilNilai obj=new tampilNilai('A');
        obj.info();
    }
}
```

## 2. OVERLOADING

Dua buah metode dalam sebuah class diperbolehkan memiliki nama yang sama dengan signature yang berbeda.

**CONTOH PROGRAM OVERLOADING SEDERHANA**

```
class mobil {
    private String warna;
    private int tahunproduksi;

    //KONSTRUKTOR PERTAMA
    public mobil (String warna, int tahunproduksi){
        this.warna =warna;
        this.tahunproduksi= tahunproduksi;

    }
    //KONSTRUKTOR KEDUA
    public mobil(){

    }
    //METOD
    public void info(){
        System.out.println ("warna :" +this.warna);
        System.out.println("tahun Produksi :" +this.tahunproduksi);

    }
}

public class Overmobil {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        mobil mobilku = new mobil ("Merah", 2003);
        mobilku.info();

        mobil mobilmu = new mobil();
        mobilmu.info();

    }
}
```

TEKNIK INFORMATIKA  
CONTOH OVERLOADING PADA KONSTRUKTOR

```
class Shape {
    int x,y;
    float rate;

    Shape (int x, int y){ //KONSTRUKTOR
        this.x=x;
        this.y=y;
        rate= (float) (x+y)/2;
    }

    Shape(int x, int y, float rate){ //KONSTRUKTOR
        this.x=x;
        this.y=y;
        this.rate=rate;
    }

    Shape () { //KONSTRUKTOR
        rate=x=y=0;
    }

    public void printfield(){
        System.out.println("\n x=" + x +
            "\n y="+ y +
            "\n rate="+ rate);
    }
}

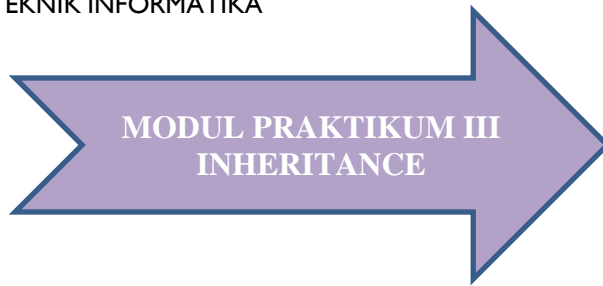
public class Over {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Shape box = new Shape(120,30);
        Shape round = new Shape(120,30,20);
        Shape rect = new Shape();
        box.printfield();
        round.printfield();
        rect.printfield();
    }
}
```

```
class matematika {  
  
    static public double kuadrat (Double nilai){  
        return nilai * nilai;  
    }  
  
    static public int kuadrat (int nilai){  
        return nilai * nilai;  
    }  
  
    static public double kuadrat (String nilai){  
        double bilangan;  
        bilangan = Double.valueOf (nilai).doubleValue();  
        return bilangan * bilangan;  
    }  
},  
public class Over3 {  
    /**  
     * @param args the command line arguments  
     */  
    public static void main (String[] args) {  
        // TODO code application logic here  
        System.out.println(matematika.kuadrat(4.0));  
        System.out.println(matematika.kuadrat(4));  
        System.out.println(matematika.kuadrat("4"));  
    }  
}
```

## TUGAS PRAKTIKUM II

Buatlah Program Overloading dari contoh program konstruktor yang ke 2, sehingga program dapat menampilkan data case tidak hanya 1





- Inheritance mekanisme di OOP yang memungkinkan class baru dibuat berdasarkan class yang sudah ada sebelumnya.
- Suatu sifat atau method didefinisikan dalam superclass, sifat ini secara otomatis diwariskan dari semua subclasses.

#### LATIHAN PRAKTIKUM INHERITANCE

##### 1. Buat class kakek

Class kakek merupakan class yang sifatnya akan diwariskan pada class bapak dan class cucu.

Tuliskan script berikut pada class kakek

```
public class kakek {
    protected String namegrandpa;
    protected String address;

    public kakek()
    {
        System.out.println("\n Program Demo Inheritance");
        System.out.println("=====");
        System.out.println("Masukan konstruktor kakek");
        System.out.println("-Dijalankan oleh class Bapak-");
        namegrandpa="Doni wijaya";
        address="Merauke";
    }
    public kakek (String name,String address)
    {
        this.namegrandpa=name;
        this.address=address;
    }

    public String getAddress() {
        return address;
    }

    public String getNamegrandpa() {
        return namegrandpa;
    }
}
```

## TEKNIK INFORMATIKA

### 2. Buat class bapak

Gunakan perintah `extends` untuk memanggil data class yang mewarisi

```
class bapak extends kakek

    protected String namefather;
    protected String addressfather;
    public bapak()

        System.out.println("Nama Kakek : " +namegrandpa);
        System.out.println("Address kakek :"+address);
        System.out.println("\n");
        System.out.println("Masukan Konstruktor Bapak ");
        System.out.println("--dijalankan class cucu--");
        namefather="Ridho Saputra";
        addressfather="Yogyakarta";
    }
    public bapak(String namefather,String addressfather)
    {
        this.namefather=namefather;
        this.addressfather=addressfather;
    }

    public String getNamefather() {
        return namefather;
    }

    public String getAddressfather() {
        return addressfather;
    }

}
```

### 3. Buat class cucu

```
    * @param args the command line arguments
    */
    public Cucu()
    {
        System.out.println("Nama Bapak :"+namefather);
        System.out.println("Address Bapak :"+addressfather);
    }

    public static void main(String[] args) {
        // TODO code application logic here
        String namacucu;
        Cucu cucu2 =new Cucu();
        System.out.println("\nSaya adalah CUCU");
        System.out.println("Nama Saya : Ane");
        System.out.println("");
    }
}
```

## TUGAS PRAKTIKUM III

Buatlah class person yang berisi data nama, alamat yang diwariskan ke data student.



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

#### LATIHAN 1

1. Buat project dengan nama bhnbakar
2. Buat class baru dengan nama kendaraan, class kendaraan merupakan class yang akan mewarisi bahan bakar
3. Buat class baru dengan nama mobil
4. Buat class baru dengan nama pesawat
5. Tuliskan script berikut pada class kendaraan

```
package bhnbakar;

/**
 *
 * @author Admin
 */
public class kendaraan{
    public void bahanbakar () {
        System.out.println("Bahan Bakar Kendaraan :");
    }
}
```

6. Tuliskan script berikut pada class mobil

```
package bhnbakar;

/**
 *
 * @author Admin
 */
public class mobil extends kendaraan{
    public void bahanbakar () {
        System.out.println("Bahan Bakar MOBIL Bensin");
    }
}
```

## TEKNIK INFORMATIKA

7. Tuliskan script berikut pada class pesawat

```
package bhnbakar;

/**
 *
 * @author Admin
 */
public class pesawat extends kendaraan{
    public void bahanbakar(){
        System.out.println("Bahan Bakar Pesawat Bensol");
    }
}
}
```

8. Tuliskan script berikut pada class utama (bhnbakar)

```
public class Bhnbakar {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        kendaraan KENDARAAN = new kendaraan();
        mobil MOBIL = new mobil();
        pesawat PESAWAT = new pesawat();
        KENDARAAN.bahanbakar();
        MOBIL.bahanbakar();
        PESAWAT.bahanbakar();
    }
}
```

## TUGAS PRAKTIKUM

1. Dari latihan praktikum diatas tambahkan class kereta api, sepeda motor dan bus.
2. Buatlah class polymorphism dari class hewan yang mewarisi "suara" pada kuda, kucing, ayam



## PRAKTIKUM V INTERFACE

Study Case: makhluk hidup yang mewarisi data kepada manusia dan hewan, yang diwarisi adalah mendengar, melihat dan berfikir.

1. Buat class dengan nama dengar

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package pkginterface;  
  
/**  
 *  
 * @author Admin  
 */  
interface dengar {  
    public void mendengar();  
  
}
```

2. Buat class dengan nama fikir

```
*/  
package pkginterface;  
  
/**  
 *  
 * @author Admin  
 */  
interface fikir {  
    public void berfikir();  
  
}
```

3. Buat class dengan nama lihat

```
*/  
package pkginterface;  
  
/**  
 *  
 * @author Admin  
 */  
interface lihat {  
    public void melihat();  
  
}
```

## TEKNIK INFORMATIKA

4. Buat class makhluk hidup yang berfungsi sebagai class yang mewarisi

```
*  
* @author Admin  
*/  
public class MakhlukHidup {  
    String dengar = "Tidak Bisa Mendengar";  
    String lihat="Tidak Bisa Melihat";  
    String fikir="Tidak Bisa Berfikir";  
  
    public void setDengar(String dengar) {  
        this.dengar = dengar;  
    }  
  
    public void setLihat(String lihat) {  
        this.lihat = lihat;  
    }  
  
    public void setFikir(String fikir) {  
        this.fikir = fikir;  
    }  
  
    public void tampil(){  
        System.out.println("Makhluk ini adalah -----");  
        System.out.println(dengar);  
        System.out.println(lihat);  
        System.out.println(fikir);  
    }  
  
}
```

5. Buat class hewan yang akan mewarisi class makhluk hidup

```
package pkginterface;  
  
/**  
 *  
 * @author Admin  
 */  
public class Hewan extends MakhlukHidup  
implements dengar, lihat{  
    public void mendengar(){  
        setDengar ("Bisa Mendengar");  
    }  
  
    public void melihat(){  
        setLihat ("Bisa Melihat");  
    }  
  
}
```

## TEKNIK INFORMATIKA

6. Buat class manusia yang akan mewarisi class makhluk hidup

```
*/
package pkginterface;

/**
 *
 * @author Admin
 */
public class manusia extends MakhlukHidup
    implements dengar, lihat, fikir{

    public void mendengar (){
        setDengar ("Bisa Mendengar");

    }
    public void melihat(){
        setLihat ("Bisa Melihat");

    }

    public void berfikir(){
        setFikir ("Bisa Berfikir");

    }
}
}
```

7. Tuliskan code berikut di class main menu untuk memanggil seluruh class pewarisan

```
/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    // TODO code application logic here
    Hewan h = new Hewan();
    System.out.println("Hewan");
    h.melihat();
    h.mendengar();
    h.tampil();
    System.out.println("");

    manusia m = new manusia();
    m.melihat();
    m.mendengar();
    m.berfikir();
    System.out.println("");
    m.tampil();
}
}
```

Tugas Praktikum:

1. Tambahkan class diatas dengan class bermain, makan, bernafas
2. Tambahkan class yang mewarisi data tumbuhan