

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK

PERTEMUAN KE-1



Disusun Oleh :

NAMA : Raden Isnawan Argi Aryasatya

NIM : 195410257

JURUSAN : Teknik Informatika

JENJANG : S1

KELAS : TI-5

Laboratorium Terpadu
Sekolah Tinggi Manajemen Informatika Komputer
AKAKOM
YOGYAKARTA

2021

PERTEMUAN KE-1 **(KELAS DAN OBJEK)**

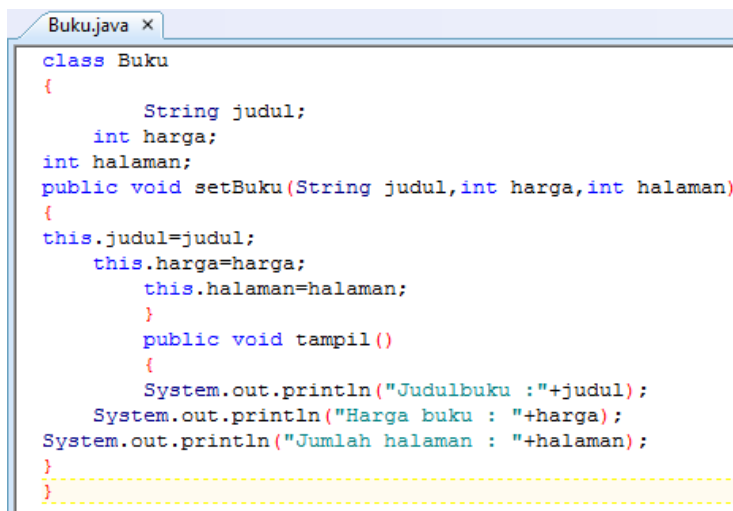
TUJUAN

Dapat membaca class diagram sederhana, Dapat menyebutkan bagian-bagian kelas, Dapat menjelaskan sintak kelas, Dapat membuat kelas sederhana, Dapat membedakan antara kelas dan objek, Dapat menciptakan Objek

DASAR TEORI

Sebuah sistem yang dibangun berdasarkan metoda berorientasi objek adalah sebuah sistem yang komponennya di enkapsulasi menjadi kelompok data dan fungsi, yang dapat mewarisi atribut dan sifat dari komponen lainnya, dan komponen-komponen tersebut saling berinteraksi satu sama lain. Bahasa Pemrograman yang berorientasi OBJEK memiliki kemampuan dalam pengelolaan program yang lebih diarahkan pada pembentukan objek. Dengan menerapkan konsep ini program akan lebih mudah untuk dikembangkan karena sifatnya yang lebih modular.

PRAKTIK 1

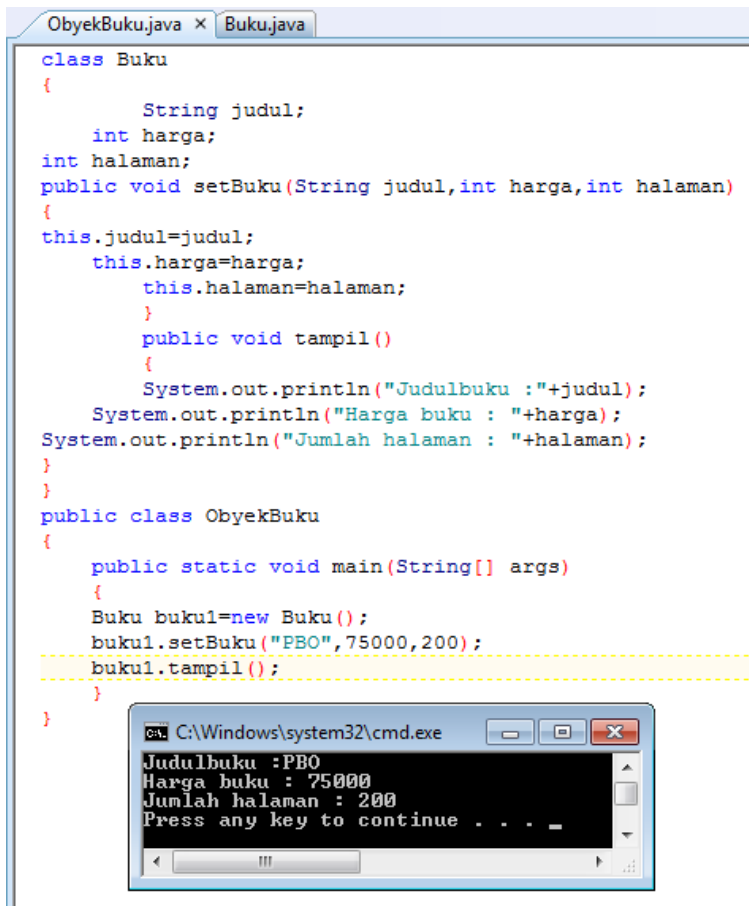


```
Buku.java x
class Buku
{
    String judul;
    int harga;
    int halaman;
    public void setBuku(String judul,int harga,int halaman)
    {
        this.judul=judul;
        this.harga=harga;
        this.halaman=halaman;
    }
    public void tampil()
    {
        System.out.println("Judulbuku :"+judul);
        System.out.println("Harga buku : "+harga);
        System.out.println("Jumlah halaman : "+halaman);
    }
}
```

penjelasan:

seperti yang sudah dijelaskan pada modul, praktik ke-1 ini adalah deklarasi dari class obyek buku. Pertama, ada setBuku yang mempunyai atribut berupa judul, harga, halaman. public void setBuku berfungsi untuk mendeklarasikan variabel. Hal tersebut dilakukan supaya variable bisa diakses di semua class. Kali ini program belum bisa mengeluarkan output karena belum ada method untuk menggunakan objek.

PRAKTIK 2



```
class Buku
{
    String judul;
    int harga;
    int halaman;
    public void setBuku(String judul,int harga,int halaman)
    {
        this.judul=judul;
        this.harga=harga;
        this.halaman=halaman;
    }
    public void tampil()
    {
        System.out.println("Judulbuku : "+judul);
        System.out.println("Harga buku : "+harga);
        System.out.println("Jumlah halaman : "+halaman);
    }
}

public class ObyekBuku
{
    public static void main(String[] args)
    {
        Buku buku1=new Buku();
        buku1.setBuku("PBO",75000,200);
        buku1.tampil();
    }
}
```

```
C:\Windows\system32\cmd.exe
Judulbuku :PBO
Harga buku : 75000
Jumlah halaman : 200
Press any key to continue . . .
```

penjelasan:

membuat dan menggunakan obyek dari class buku dengan **buku1.setBuku ("PBO",75000,200);**

lalu untuk menampilkan output menggunakan **buku1.tampil();**

PBO adalah variabel judul

60000 adalah variabel harga

100 adalah variabel halaman

LATIHAN

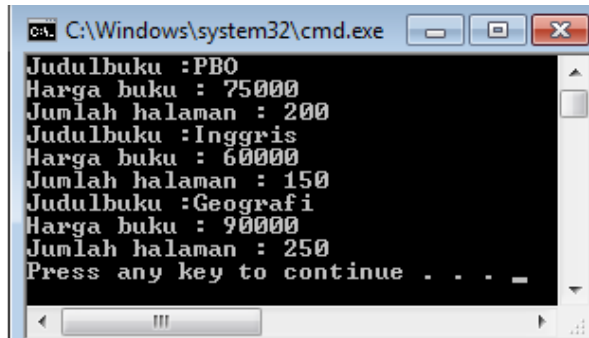
1. Tambahkan 2 buah objek dari kelas Buku !

```
public class ObyekBuku
{
    public static void main(String[] args)
    {
        Buku buku1=new Buku();
        buku1.setBuku("PBO",75000,200);
        buku1.tampil();

        Buku buku2=new Buku();
        buku2.setBuku("Inggris",60000,150);
        buku2.tampil();

        Buku buku3=new Buku();
        buku3.setBuku("Geografi",90000,250);
        buku3.tampil();
    }
}
```

2. Panggil method `setBuku()` dan `tampil()`

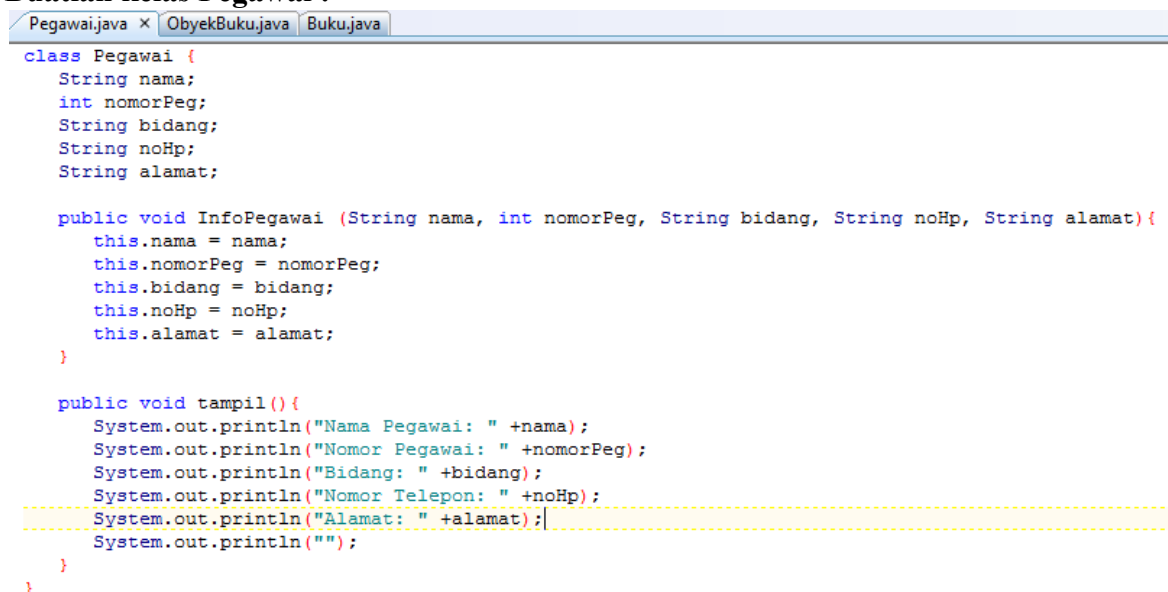


```
C:\Windows\system32\cmd.exe
Judulbuku :PBO
Harga buku : 75000
Jumlah halaman : 200
Judulbuku :Inggris
Harga buku : 60000
Jumlah halaman : 150
Judulbuku :Geografi
Harga buku : 90000
Jumlah halaman : 250
Press any key to continue . . .
```

Penjelasan (Latihan nomor 1 dan 2):

Prinsipnya sama seperti praktik 2. buatlah objek dengan nama **buku2** dan **buku3**. Di program tersebut, **buku2** dideklarasikan oleh **buku2.setBuku("Inggris",60000,150);** lalu ditampilkan dengan **buku2.tampil();** , hal tersebut menghasilkan Judul buku Inggris, Harga buku 60000, Jumlah halaman 150. Lalu **buku3** dideklarasikan oleh **buku3.setBuku("Geografi",90000,250);** lalu ditampilkan dengan **buku3.tampil();** , hal tersebut menghasilkan Judul buku Geografi, Harga buku 90000, Jumlah halaman 250.

3. Buatlah kelas Pegawai !



```
Pegawai.java x ObyekBuku.java Buku.java
class Pegawai {
    String nama;
    int nomorPeg;
    String bidang;
    String noHp;
    String alamat;

    public void InfoPegawai (String nama, int nomorPeg, String bidang, String noHp, String alamat){
        this.nama = nama;
        this.nomorPeg = nomorPeg;
        this.bidang = bidang;
        this.noHp = noHp;
        this.alamat = alamat;
    }

    public void tampil(){
        System.out.println("Nama Pegawai: " +nama);
        System.out.println("Nomor Pegawai: " +nomorPeg);
        System.out.println("Bidang: " +bidang);
        System.out.println("Nomor Telepon: " +noHp);
        System.out.println("Alamat: " +alamat);
        System.out.println("");
    }
}
```

Penjelasan:

class Pegawai berfungsi untuk mendeklarasikan tipe data class yang bertugas untuk mengumpulkan fungsi dan variabel di satu tempat. Di dalam class tersebut terdapat deklarasi variabel **nama**, **nomorPeg**, **Bidang**, **noHp**.

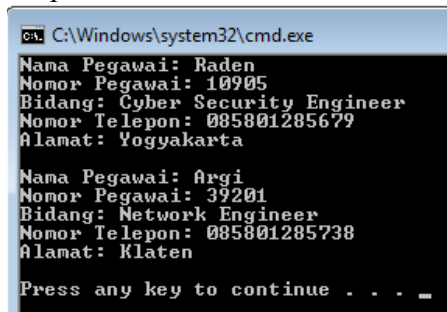
public void InfoPegawai(String nama,int nomorPeg,String bidang ,String noHp, String alamat) merupakan method yang memiliki parameter deklarasi variable **this.nama=nama; this.nomorPeg=nomorPeg; this.bidang=bidang; this.noHp=noHp; dan this.alamat=alamat;**

4. Buatlah 2 buah objek dari kelas pegawai!

```
public class ObyekPegawai {
    public static void main (String[] args){
        Pegawai pegawai1=new Pegawai();
        pegawai1.InfoPegawai("Raden",10905,"Cyber Security Engineer", "085801285679", "Yogyakarta");
        pegawai1.tampil();

        Pegawai pegawai2=new Pegawai();
        pegawai2.InfoPegawai("Argi",39201,"Network Engineer", "085801285738", "Klaten");
        pegawai2.tampil();
    }
}
```

Output:



```
C:\Windows\system32\cmd.exe
Nama Pegawai: Raden
Nomor Pegawai: 10905
Bidang: Cyber Security Engineer
Nomor Telepon: 085801285679
Alamat: Yogyakarta

Nama Pegawai: Argi
Nomor Pegawai: 39201
Bidang: Network Engineer
Nomor Telepon: 085801285738
Alamat: Klaten

Press any key to continue . . . _
```

penjelasan:

ObyekPegawai berfungsi untuk memberi nama pada class. public static void main(String[] args) berfungsi untuk memanggil method yang dituju pada suatu program untuk dijalankan. Pegawai pegawai1=new Pegawai() berfungsi untuk membuat objek baru pada Pegawai dengan nama pegawai1 dengan isi pegawai1. **InfoPegawai("Raden",10905,"Cyber Security Engineer","085801285679","Yogyakarta")** akan mengeluarkan hasil seperti di output atas. Konsep tersebut juga berlaku sama dengan pegawai2.

TUGAS

1. Implementasikan ke dalam program dari contoh Tabel 1

```
1 class Mobil {
2     String plat;
3     String warna;
4     String manufaktur;
5     int kecepatan;
6     String arah;
7     String rem;
8     int perubahan;
9
10    public void setMobil (String plat, String warna, String manufaktur, int kecepatan){
11        this.plat=plat;
12        this.warna=warna;
13        this.manufaktur=manufaktur;
14        this.kecepatan=kecepatan;
15    }
16    public void akselerasi(int perubahan){
17        this.perubahan = perubahan+kecepatan;
18    }
19    public void belok(String arah){
20        this.arah=arah;
21    }
22    public void rem(String rem){
23        this.rem=rem;
24    }
25    public void tampil(){
26        System.out.println("Nomor Plat \t: "+plat);
27        System.out.println("Warna \t: "+warna);
28        System.out.println("Manufaktur \t: "+manufaktur);
29        System.out.println("Kecepatan \t: "+kecepatan+" km/h");
30        System.out.println("Akselerasi \t: "+perubahan+" km/h");
31        System.out.println("Arah Belok \t: "+arah);
32        System.out.println("Kondisi Rem \t: "+rem);
33        System.out.println("");
34    }
35 }
```

penjelasan:

di dalam class Mobil ada variabel plat, warna, manufaktur, kecepatan, arah, rem, perubahan. Semua string kecuali kecepatan dan perubahan, 2 variabel itu merupakan integer. Di dalam **public void setMobil(String plat, String warna, String manufaktur, int kecepatan)** adalah method bernama setMobil yang didalamnya ada parameter yang mendefinisikan variabel yaitu

this.plat=plat;

this.warna=warna;

this.manufaktur=manufaktur;

this.kecepatan=kecepatan;

lalu setelahnya terdapat public static dengan nama method akselerasi, belok, rem, dan tampil. Semuanya mengandung parameter beserta variabelnya. Konsepnya sama persis.

ObyekMobil

```
37 public class ObyekMobil{
38 public static void main (String[] args){
39     Mobil mobilA = new Mobil();
40     mobilA.setMobil("DK 2473 KL", "Putih", "Honda", 60);
41     mobilA.akselerasi(30);
42     mobilA.belok("kanan/kiri");
43     mobilA.rem("Berfungsi");
44     mobilA.tampil();
45
46     Mobil mobilB = new Mobil();
47     mobilB.setMobil("DK 1958 KL", "Hitam", "Ford", 90);
48     mobilB.akselerasi(30);
49     mobilB.belok("kanan/kiri");
50     mobilB.rem("Berfungsi");
51     mobilB.tampil();
52 }
53 }
```

output:

```
CA: C:\Windows\system32\cmd.exe
Nomor Plat      : DK 2473 KL
Warna Mobil     : Putih
Manufaktur      : Honda
Kecepatan       : 60 km/h
Akselerasi      : 30 km/h
Arah Belok      : kanan/kiri
Kondisi Rem      : Berfungsi

Nomor Plat      : DK 1958 KL
Warna Mobil     : Hitam
Manufaktur      : Ford
Kecepatan       : 90 km/h
Akselerasi      : 120 km/h
Arah Belok      : kanan/kiri
Kondisi Rem      : Berfungsi

Press any key to continue . . .
```

2. Buatlah Kelas Handphone

```
class Handphone {
    String nama;
    String warna;
    String ram;
    String os;
    String harga;

    public void InfoHP (String nama, String warna, String ram, String os, String harga){
        this.nama = nama;
        this.warna = warna;
        this.ram = ram;
        this.os = os;
        this.harga = harga;
    }

    public void tampil(){
        System.out.println("Nama Handphone \t: " +nama);
        System.out.println("Warna Handphone \t: " +warna);
        System.out.println("Kapasitas RAM \t: " +ram);
        System.out.println("Operating System \t: " +os);
        System.out.println("Harga Handphone \t: " +harga);
        System.out.println("");
    }
}
```

penjelasan:

class Handphone memiliki variabel nama, warna, ram, os, harga. Lalu public void infoHP(String nama,String warna,String ram,String os, String harga) adalah method bernama infoHP yang didalamnya ada parameter yang mendaklarasikan variabel yaitu

this.nama=nama

this.warna=warna;

this.ram=ram

this.os=os

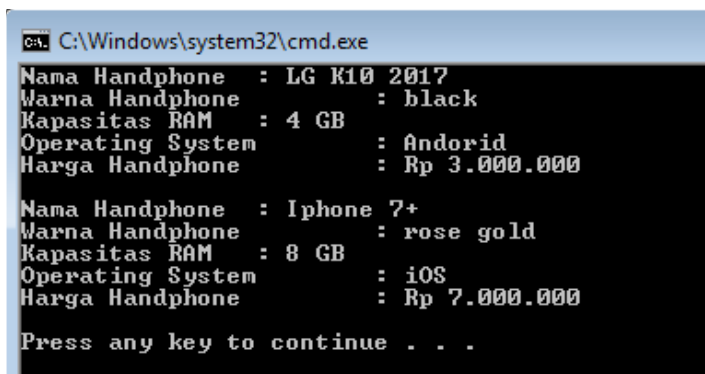
this.harga=harga

3. Buatlah 2 buah objek dari kelas handphone !

```
public class ObyekHandphone{
    public static void main (String[] args){
        Handphone hp1=new Handphone();
        hp1.InfoHP("LG K10 2017","black","4 GB","Andorid","Rp 3.000.000");
        hp1.tampil();

        Handphone hp2=new Handphone();
        hp2.InfoHP("Iphone 7+","rose gold","8 GB","iOS","Rp 7.000.000");
        hp2.tampil();
    }
}
```

Output:



```
C:\Windows\system32\cmd.exe
Nama Handphone : LG K10 2017
Warna Handphone : black
Kapasitas RAM : 4 GB
Operating System : Andorid
Harga Handphone : Rp 3.000.000

Nama Handphone : Iphone 7+
Warna Handphone : rose gold
Kapasitas RAM : 8 GB
Operating System : iOS
Harga Handphone : Rp 7.000.000
Press any key to continue . . .
```

penjelasan:

Handphone hp1=new Handphone() berfungsi untuk membuat objek pada class Handphone dengan nama hp1. Lalu **hp1.InfoHP("LG K10 2017","black","4 GB","Andorid","Rp 3.000.000");** akan mengeluarkan output seperti gambar diatas. Kemudian hp1.tampil(); akan menampilkan output. Konsep ini berlaku juga untuk hp2.

KESIMPULAN

Dalam konsep object oriented akan kita temukan kata object dan class, class merupakan pola / template yang menggambarkan kumpulan object yang mempunyai sifat yang sama, perilaku, atau disebut dengan himpunan object sejenis. Sementara object adalah implementasi dari class. Adapun proses pembuatan obyek dari kelas disebut instantiasi atau implementasi dari kelas.

Terdapat 3 konsep penting dalam OOP, yaitu : enkapsulasi, pewarisan, dan polimorfisme. Enkapsulasi bertujuan untuk menyembunyikan kerumitan implementasi dan menyediakan interface bagi pengguna. Selain itu enkapsulasi juga berguna untuk proteksi data (read/write). Pewarisan adalah salah satu cara untuk menggunakan ulang kode-kode yang telah dibuat sebelumnya. Sedangkan polimorfisme adalah kemampuan obyek untuk bereaksi secara berbeda terhadap “pesan yang sama”, tergantung obyek yang menerima pesan tersebut. Kemampuan ini berkat adanya konsep method overriding dan method overloading.