

## **PERTEMUAN KE – 1**

### ***Kelas dan Obyek***

#### **A. TUJUAN**

1. Dapat membaca class diagram sederhana
2. Dapat menyebutkan bagian-bagian kelas
3. Dapat menjelaskan sintak kelas
4. Dapat membuat kelas sederhana
5. Dapat membedakan antara kelas dan obyek
6. Dapat menciptakan Obyek

#### **B. TEORI SINGKAT**

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.

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.

Tabel berikut contoh ilustrasi kelas dan objek.

*Tabel : Contoh class car dan object-object nya*

Kelas Mobil		Objek Mobil A	Objek Mobil B
Instan Variabel	nomor Plat	AB 3313 SY	AB 1234 AT
	Warna	Biru	Merah
	Manufaktur	Mitsubishi	Toyota
	Kecepatan	50 km/h	100 km/h
Instan Metode	method akselerasi		
	method belok		
	method rem		

## Membuat Class dan Obyek dalam Java

❑ Elemen-elemen dasar dalam mendefinisikan kelas :

1. Field (variabel) : menyimpan data untuk setiap objek (implementasi dari atribut)
2. Constructor : setup objek di awal
3. Method : implementasi perilaku objek

```
class NamaKelas
{
    Fields
    Constructor
    Methods
}
```

Untuk membuat obyek digunakan kata kunci new :

**NamaKelas NamaObyek = new NamaKelas();**

Untuk memanggil method dapat digunakan sintaks sebagai berikut:

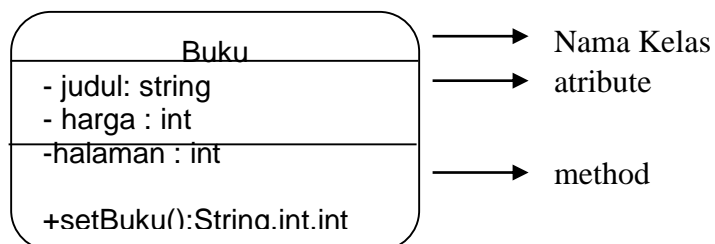
**namaObyek.nama\_method( [parameter] );**

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.

## **C. PRAKTIK**

### Praktik 1 . Membuat kelas

Berikut ini adalah contoh class diagram untuk kelas Buku



Kode program:

```

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);
    }
}

```

## Praktik 2. Membuat dan Menggunakan Obyek

Kelas Buku di atas hanya merupakan deklarasi dari obyek buku. Agar dapat digunakan maka Kelas Buku tersebut harus diinstantiasi. Dari satu buah kelas dapat dibuat banyak obyek. Obyek-obyek tersebut dapat dibedakan dengan suatu pengenal yang berupa nama.

```

public class ObyekBuku
{
    public static void main(String[] args)
    {
        Buku bukul=new Buku();
        bukul.setBuku("PBO",60000,100);
        bukul.tampil();
    }
}

```

## Praktik 3. Kelas dan Obyek dengan menggunakan Konstruktor

```

public class Pegawai
{
    String NamaPegawai;
    int IdPegawai;
    String PosisiPegawai;

    Pegawai (String nama,int Id, String posisi) // konstruktor
    {
        NamaPegawai = nama;
        IdPegawai = Id;
        PosisiPegawai = posisi;
    }
    Pegawai () //konstruktor
    {

```

```

       >NamaPegawai = "Azkiya";
        IdPegawai = 2514;
        PosisiPegawai = "Staf pengajar";
    }
    void Show()
    {
        System.out.println("Informasi Pegawai");
        System.out.println("Nama      : "+NamaPegawai);
        System.out.println("Id       : "+IdPegawai);
        System.out.println("Posisi    : "+PosisiPegawai);
    }
    public static void main(String args[])
    {

        Pegawai pegawai1 = new Pegawai();
        Pegawai pegawai2 = new Pegawai("Zahwa", 3313, "Staf Akademik");
        pegawai1.Show();
        pegawai2.Show();

    }
}

```

#### D. LATIHAN

- *Latihan diberikan oleh dosen pengampu pada saat praktikum.*
- *Dikerjakan di laboratorium pada jam praktikum.*

#### E. TUGAS

- *Tugas diberikan oleh dosen pengampu pada akhir praktikum.*
- *Dikerjakan di rumah dan dilampirkan pada laporan.*