

**TUGAS PERTAMA**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**“ Object Oriented Programming (OOP) ”**



**Disusun Oleh :**

Nama : Mubessirul Ummah  
NIM : 210411100140  
Mata Kuliah : Pemrograman Berorientasi Objek  
Dosen Pengampu : HERMAWAN, ST., M.Kom.

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TRUNOJOYO MADURA**  
**2022**

1. What is Class and Object in OOP....?

Jawaban :

a) Class

Class adalah prototype, atau blueprint, atau rancangan yang mendefinisikan variable dan method-methode pada seluruh objek tertentu. Class adalah wadah untuk menampung Object. Jadi class akan memiliki fungsi guna mengumpulkan prosedur/fungsi dan menampung isi dari program yang akan di jalankan dalam satu tempat. Penamaan Class menjadi representasi dari objek yang ingin dibuat maka nama yang dipilih harus sesuai dengan apa yang akan dibuat, hal ini akan memudahkan kita ketika ingin menghubungkan beberapa Class. Adapun terdapat aturan dalam penulisan nama Class yaitu setiap nama wajib ditulis dengan huruf kapital dan apabila terdapat beberapa kata yang menggambarkan suatu objek tidak diperbolehkan menggunakan spasi.

Lalu di dalam class sendiri terdapat field (atribut atau type data) dan method untuk menjalankan suatu program. Field merupakan tipe data yang didefinisikan oleh class, sementara method merupakan operasi.

Contoh :

```
class kucing{
    var mood;
    var energi;

    lari()
    tidur()
    makan()
    loncat()
}

class makanan{
    var nama;
    var rasa;

    hide()
}
```

Dengan class, kita bisa menentukan mana variabel dan prosedur untuk kucing dan makanan.

b) Object

Objek adalah sebuah variabel yang merupakan instance atau perwujudan dari Class. Variabel dan method dalam obyek Java diketahui sebagai variabel instance dan method instance. Hal ini dilakukan untuk membedakan dari variable class dan method class.

Jadi obyek adalah sebuah perangkat lunak yang berisi sekumpulan variable dan method yang berhubungan. Object secara sederhana bisa dimaknai sebagai suatu permasalahan yang akan diselesaikan atau hal yang ingin dibuat. Objek ini nanti bisa diklasifikasikan menjadi objek-objek yang lebih detail lagi.

Contoh :

```
public class Kendaraan{
    // Konstruktor Dengan Parameter
    public Kendaraan(String nama){
        System.out.println("Nama Kendaraannya Adalah "+ nama);
    }
    public static void main(String[] args){
        // Perintah untuk membuat objek jenis
        Kendaraan jenis = new Kendaraan("Pesawat Terbang");
    }
}
```

## 2. What is different of Class and Object....?

Jawaban :

Jadi perbedaan dari class dan object sendiri adalah Class sebenarnya bertugas untuk mengumpulkan prosedur/fungsi dan variabel dalam satu tempat. Sedangkan objek adalah sebuah variabel yang merupakan instance atau perwujudan dari Class.

Berikut penjelasannya :

- Class merupakan kumpulan dari suatu fungsi-fungsi yang dibuat untuk suatu tujuan tertentu. Misalnya kita akan membuat sebuah Mobil. Maka yang perlu kita buat adalah beberapa class yang didalamnya berisi fungsi-fungsi yang dapat membuat Mobil itu dapat berjalan sebagaimana mestinya, misalnya class Roda, class Mesin, class Kerangka Mobil, class Body Mobil dan lain lain. Class masih merupakan sebuah konsep yang belum siap digunakan.
- Sedangkan Object adalah pemanfaatan dari class yang siap untuk digunakan, misalnya kita sudah membuat class Roda, yang di dalamnya itu berisi konsep bagaimana Roda itu bisa berjalan, bagaimana Roda itu bisa berputar dan lain-lain, jika semua konsep Roda sudah oke, barulah kita akan dirubah menjadi sebuah Roda yang siap untuk dipakai. Nah itulah yang dinamakan Object Roda.

Berikut adalah contoh yang menjelaskan perbedaan Object dan Class.

a) Object: Manusia

Class: Manusia dapat berjalan, melihat, berlari, memiliki kaki, mata, rambut, hidung, dapat berbicara dan lain-lain.

b) Object: Komputer

Class: Komputer memiliki layar, prosesor, ram, bisa dimatikan, dihidupkan, bisa memutar musik dan lain-lain.

Tabel perbandingan perbedaan :

Tidak	Kelas	Obyek
1	Kelas digunakan sebagai template untuk mendeklarasikan dan membuat objek.	Objek adalah turunan dari kelas.
2	Ketika sebuah kelas dibuat, tidak ada memori yang dialokasikan.	Objek dialokasikan ruang memori setiap kali mereka dibuat.
3	Kelas harus dideklarasikan hanya sekali.	Sebuah objek dibuat berkali-kali sesuai kebutuhan.
4	Kelas tidak dapat dimanipulasi karena tidak tersedia dalam memori.	Objek dapat dimanipulasi.
5	Kelas adalah entitas logis.	Objek adalah entitas fisik.
6	Itu dideklarasikan dengan kata kunci class	Itu dibuat dengan nama kelas di C++ dan dengan kata kunci baru di Jawa.
7	Kelas tidak berisi nilai apa pun yang dapat dikaitkan dengan bidang.	Setiap objek memiliki nilainya sendiri, yang terkait dengannya.
8	Kelas digunakan untuk mengikat data serta metode bersama-sama sebagai satu unit.	Objek seperti variabel dari kelas.

### 3. How to create a new Object from a Class in Java OOP....?

Jawaban :

- Agar bisa membuat object, di dalam konsep OOP ada, maka kita harus membuat class terlebih dahulu, dalam bahasa java proses instalasi dilakukan dengan menulis nama class, lalu diikuti nama variable yang akan menampung object tersebut, tanda sama dengan "=", keyword new dan kembali nama class dengan tambahan tanda kurung "()".
- Hasil cetakan class akan disimpan dalam variable untuk selanjutnya dipakai selama penulisan kode program. (Object)

```
class Laptop {

    // buat 3 property untuk class Laptop
    String pemilik;
    String merk;
    double ukuranLayar;

    // buat 2 method untuk class Laptop
    String hidupkanLaptop() {
        return "Hidupkan Laptop";
    }

    String matikanLaptop() {
        return "Matikan Laptop";
    }

};
```

Di antara baris 1-16, terdapat perintah untuk membuat class Laptop lengkap dengan 3 property dan 2 method.

```
class BelajarJava {

    public static void main(String args[]){
        Laptop laptopmubes = new Laptop();
    }
}
```

```
}
```

```
}
```

Kemudian di baris 18 – 24 terdapat perintah untuk membuat class BelajarJava yang akan meng-instansiasi object laptopmubes dari class Laptop.

- c) Agar kita bisa menampilkan dan mengakses isi sebuah object, object tersebut harus di buat dari dalam method public static void main().
- d) Selanjutnya save kode yang telah dibuat saya beri nama “laptopmubes.java”, dan compile maka di dalam folder tempat kita menyimpan kodenya akan muncul sebuah java byte code dengan nama laptopmubes.class

```
D:\folderjava>javac.exe laptopmubes.java
```

Laptop.class	07/09/2022 01.49	File CLASS	1 KB
laptopmubes	07/09/2022 01.37	Java Source File	1 KB

- e) Selanjutnya masuk pada bagian mengakses object di dalam java nya, yakni untuk mengakses isis dari sebuah object, yakni property dan method

```
class Laptop {  
  
    // buat 3 property untuk class Laptop  
    String pemilik;  
    String merk;  
    double ukuranLayar;  
  
    // buat 2 method untuk class Laptop  
    String hidupkanLaptop() {  
        return "Hidupkan Laptop";  
    }  
  
    String matikanLaptop() {  
        return "Matikan Laptop";  
    }  
};  
  
class BelajarJava {  
  
    public static void main(String args[]){  
  
        // buat object dari class Laptop (instansiasi)  
        Laptop laptopmubes = new Laptop();  
  
        // set nilai property  
        laptopmubes.pemilik = "Mubessirul Ummah";  
        laptopmubes.merk = "Lenovo";  
        laptopmubes.ukuranLayar = 14;  
  
        // tampilkan property object Laptop  
        System.out.println(laptopmubes.pemilik);  
        System.out.println(laptopmubes.merk);  
    }  
}
```

```

System.out.println(laptopmubes.ukuranLayar);

// tampilkan hasil method object Laptop
System.out.println(laptopmubes.hidupkanLaptop());
System.out.println(laptopmubes.matikanLaptop() );
}
}

```

Penjelasan code :

- Pembuatan class laptop

```

class Laptop {
    String pemilik;
    String merk;
    double ukuranLayar;
}

```

Kurung kurawal menandakan awal dari class. Setelah itu terdapat pendefinisian 3 variabel class atau dikenal dengan istilah property. Property ini sebenarnya tidak lain dari variabel yang berada di dalam class.

- Method

```

String hidupkanLaptop() {
    return "Hidupkan Laptop";
}

String matikanLaptop() {
    return "Matikan Laptop";
}

```

Dua buah fungsi ini adalah method dari class Laptop. Ini tidak lain hanya function yang mengembalikan string.

- Method main(), Fungsi main() adalah fungsi utama dalam program Java. Semua kode yang kita tulis di dalamnya, akan langsung dieksekusi.

```

class BelajarJava {
    public static void main(String args[]){
        Laptop laptopmubes = new Laptop();
    }
}

```

- Proses instansiasi, yakni perintah untuk membuat object dari class laptop.

```

laptopmubes.pemilik = "Mubessirul Ummah";
laptopmubes.merk = "Lenovo";
laptopmubes.ukuranLayar = 14;

```

Ini adalah cara mengisi nilai ke dalam property dari object laptopmubes. Cara pengisiannya sama seperti mengisi nilai ke variabel biasa, yakni dari kanan ke kiri dengan operasi assignment (tanda sama dengan "="). Perhatikan tanda titik untuk mengakses property dari object. Tanda titik ini adalah operator khusus yang dikenal dengan istilah Object Operator.

- ```
System.out.println(laptopmubes.pemilik);
System.out.println(laptopmubes.merk);
System.out.println(laptopmubes.ukuranLayar);
```

- Pemanggilan method dari object

```
System.out.println(laptopmubes.hidupkanLaptop());  
System.out.println(laptopmubes.matikanLaptop());
```

- Hasil kode program

```
PS D:\folderjava> & 'C:\Program Files\Lenovo\Lenovo\Lenovo.exe' -cp 'C:\Users\Lenovo\AppData\Local\Temp\Lenovo\Lenovo.exe'
Mubessirul Ummah
Lenovo
14.0
Hidupkan Laptop
Matikan Laptop
PS D:\folderjava>
```

Jawaban :

Metode Accessor adalah metode yang digunakan untuk membaca nilai variabel pada suatu class, baik berupa instance maupun static. Metode ini digunakan untuk mengembalikan nilai dari sebuah private field. Skema ini biasanya ditandai dengan awalan nama "get". Metode accesor selalu mengembalikan value/nilai dengan tipe data yang sama sesuai korespondensi private fieldnya.

Metode Mutator merupakan sebuah metode yang digunakan untuk menetapkan nilai dari sebuah private field. Skema ini biasanya ditandai dengan awalan nama "set". Metode ini tidak memiliki nilai balik (return), sebab metode ini merupakan pengaturan parameter yang akan digunakan oleh koresponden private fieldnya, sama dengan Enkapsulasi.

Jawaban :

```
import java.util.Scanner;  
class menghitung {  
    /*kubus*/  
    static int panjangkubus;
```

```

static int volumekubus;

/*balok*/
static int panjangbalok;
static int lebarbalok;
static int tinggibalok;
static int volumebalok;

/*segitiga*/
static int tinggisegitiga;
static int alassegitiga;
static int luassegitiga;

private static Scanner hitung;

public static void main(String[] args){

    hitung = new Scanner(System.in);

    /*Kubus*/
    System.out.println("Masukan Panjang Sisi Kubus :");
    panjangkubus=hitung.nextInt();

    volumekubus = (panjangkubus)*3;
    System.out.println("\nHasil dari volume kubus = " +volumekubus);

    System.out.print("\n");

    /*Balok*/
    System.out.println("Masukan Panjang Balok :");
    panjangbalok=hitung.nextInt();
    System.out.println("Masukan Lebar Balok :");
    lebarbalok=hitung.nextInt();
    System.out.println("Masukan Tinggi Balok :");
    tinggibalok=hitung.nextInt();

    volumebalok = (panjangbalok*lebarbalok*tinggibalok);
    System.out.println("\nHasil dari volume balok = " +volumebalok);

    System.out.print("\n");

    /*segitiga*/
    System.out.println("Masukan Tinggi Segitiga :");
    tinggisegitiga=hitung.nextInt();
    System.out.println("Masukan Panjang Alas Segitiga :");
    alassegitiga=hitung.nextInt();

    luassegitiga = (alassegitiga*tinggisegitiga)/2;

```



```
System.out.println("\nHasil dari luas segitiga = " +luassegitiga);
}}
```

Hasil running program :

```
PS D:\folderjava> & 'C:\Users\Lenovo\AppData\Local\Microsoft\Windows\Workspaces\Code\User\workspace
:\Users\Lenovo\AppData\Roaming\Code\User\workspace
Masukan Panjang Sisi Kubus :
3
Hasil dari volume kubus = 9
Masukan Panjang Balok :
5
Masukan Lebar Balok :
6
Masukan Tinggi Balok :
7
Hasil dari volume balok = 210
Masukan Tinggi Segitiga :
2
Masukan Panjang Alas Segitiga :
5
Hasil dari luas segitiga = 5
PS D:\folderjava> █
```