

- Berorientasi Obyek / Object Oriented adalah paradigma yang memandang sistem sebagai kumpulan obyek-obyek diskrit yang saling berinteraksi satu sama lain.
- Obyek adalah segala sesuatu/benda, baik yang secara fisik dapat diketahui keberadaannya maupun yang bersifat khayal (virtual).

Misalkan : mobil, truk, sepeda, kucing, harimau, singa, kamera, dll.

 Setiap obyek mempunyai 2 pengenal, yaitu Informasi (atribut) tentang obyek tersebut dan Perilaku (behavior) yang mengaturnya.



Attribute:

Topi, Baju, Jaket, Tas Punggung, Tangan, Kaki, Mata

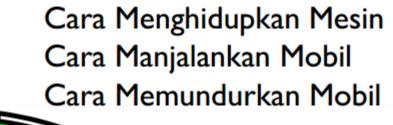
Behavior:

Cara Jalan ke Depan Cara Jalan Mundur Cara Belok ke Kiri Cara Memanjat

Attribute (State):

Ban, Stir, Pedal Rem, Pedal Gas, Warna, Tahun Produksi

Behavior:



Attribute → Variable(Member)
Behavior → Method(Fungsi)

 Obyek-obyek mempunyai atribut dan operasi yang sama dapat dikelompokkan dalam sebuah kategori, yang disebut dengan Kelas atau Class.

Contohnya :

Mobil, truk, sepeda, truk, becak adalah kategori kendaraan Kucing, harimau, singa, ayam adalah kategori binatang Kamera, HP, televisi, radio adalah kategori barang elektronik

Sejauh mana kategori sebuah kelas, bergantung pada semesta pembicaraan.

- Perbedaan Kelas dan Obyek :
 - ✓ Kelas: konsep dan deskripsi dari sesuatu Kelas mendeklarasikan method/fungsi yang dapat digunakan/dipanggil oleh obyek.
 - ✓ Obyek : bentuk (contoh) nyata dari Kelas Obyek memiliki sifat independen dan dapat digunakan untuk memanggil method.
- Contoh Kelas dan Obyek :

Kelas: mobil

Obyek: mobil Joko, mobilnya P. Bondan

Pemrograman Berorientasi Obyek

Pemrograman berorientasi obyek memungkinkan pengelolaan data dan fungsi dalam satu kesatuan yang disebut sebagai Obyek.

Menciptakan Kelas & Obyek

Deklarasi kelas:

```
class nama_kelas
{
    atribut;
    ......
    behavior/method;
}
```

*Semua atribut dari obyek didefinisikan dalam sebuah kelas

Setelah kelas dideklarasikan, kelas tersebut digunakan untuk mendefinisikan /instansiasi sebuah obyek.

Deklarasi obyek :

```
nama_kelas nama_obyek = new nama_kelas();
```

Menciptakan Kelas & Obyek

Jika obyek hendak mengakses atribut maupun method yang sudah dideklarasikan dalam kelas, maka gunakan sintak berikut:

```
nama_obyek • atribut
    atau

nama obyek • method
```

Menciptakan Kelas & Obyek

Contoh listing program_1:

```
class lingkaran {
     double r; //atribut
     public double keliling() { //method
       return 2 * 3.14 * r;
     public double luas() { //method
       return 3.14 * r * r ;
public class coba {
   public static void main(String [] args) {
      lingkaran hitung = new lingkaran;
      hitung.r = 10;
      System.out.println("Kell Lingkaran = "+hitung.keliling());
      double luasling = hitung.luas();
      System.out.println("Luas Lingkaran = "+luasling);
```

Modifier - Acces Modifier

public

Kelas, method maupun atribut yang mempunyai access modifier public dapat di akses dari manapun dan oleh kelas apapun.

protected

Atribut dan method yang mempunyai access modifier protected, artinya bahwa atribut dan method tersebut dapat diakses oleh kelas yang sama, package yang sama, dan kelas turunannya (*subclass*).

default / no access

Atribut dan method yang mempunyai access modifier default, artinya bahwa atribut dan method tersebut dapat diakses oleh kelas yang sama dan package yang sama.

private

Atribut dan method yang mempunyai access modifier default, artinya bahwa atribut dan method tersebut dapat diakses oleh kelas yang sama dan package yang sama.

