



Modul 1 - Pengenalan PBO (Konsep Class, Object, dan Method)

Durasi Modul = 5 menit

Capaian pembelajaran:

1. Mampu memahami konsep pemrograman berorientasi objek.
2. Mampu memahami konsep object, class, dan method.
3. Mampu membedakan apa itu object, class, dan method.

Deskripsi Pembelajaran :

Dalam modul ini, kita akan mempelajari konsep dasar Pemrograman Berorientasi Objek (PBO) atau Object-Oriented Programming (OOP). Kita akan membahas tiga elemen kunci dalam PBO:

1. *Object*: Kita akan memahami apa itu objek dalam konteks pemrograman, bagaimana objek merepresentasikan entitas dunia nyata dalam kode, dan bagaimana objek berinteraksi satu sama lain.
2. *Class*: Kita akan mempelajari konsep class sebagai blueprint atau cetakan untuk membuat objek. Kita akan memahami bagaimana class mendefinisikan struktur dan perilaku objek.
3. *Method*: Kita akan membahas method sebagai fungsi atau perilaku yang dimiliki oleh objek. Kita akan mempelajari bagaimana method digunakan untuk memanipulasi data dan melakukan aksi dalam konteks PBO.

Melalui penjelasan, contoh, dan latihan, kita akan membangun pemahaman yang kuat tentang konsep-konsep ini dan bagaimana mereka bekerja bersama dalam paradigma pemrograman berorientasi objek.

TENTANG OOP, OBJECT, CLASS, METHOD

Istilah *Object Oriented Programming* (OOP) pertama kali digunakan oleh **Xerox PARC** di luar bahasa pemrograman lainnya. Istilah ini digunakan untuk merujuk kepada proses yang menggunakan object sebagai dasar untuk penghitungan.



Pemrograman Berorientasi Objek atau *Object Oriented Programming* (OOP) adalah suatu metode pemrograman yang berorientasi kepada objek.



Contoh, pesawat sebagai **objek** yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, dimana objek-objek tersebut saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain

▼ Kelebihan dari metode PBO sendiri ;

1. Maintenance program lebih mudah dan program yang dibuat dapat lebih mudah dibaca dan dipahami.
2. Mudah dalam pengubahan program, bisa berupa penambahan atau penghapusan fitur atau objek tertentu. Contoh pengubahan yang bisa dilakukan

antara lain

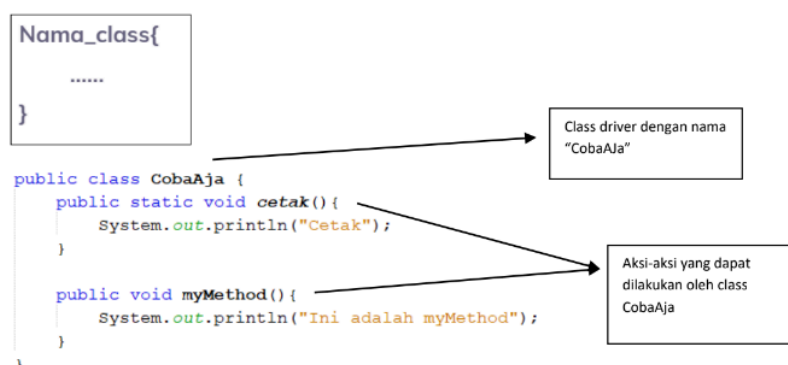
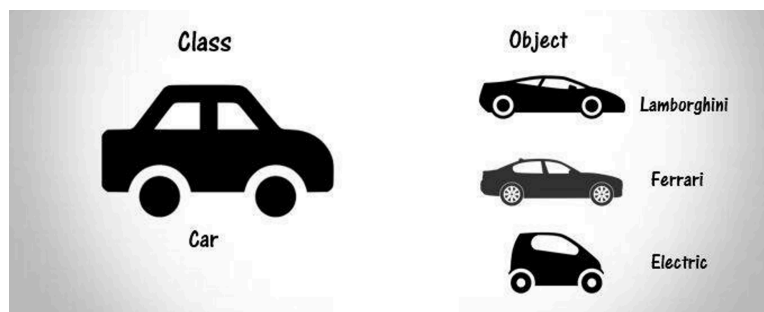
penambahan dan penghapusan data di dalam suatu database.

- Objek-objek di dalam program dapat digunakan sesering mungkin oleh programmer untuk menyimpan objek-objek yang dirancang ke dalam sebuah module, yang dapat disisipkan ke dalam baris kode baru.



Jadi setiap bagian dari suatu permasalahan adalah **objek**, sedangkan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Contohnya, **Pesawat** sebagai sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dan lain-lain.

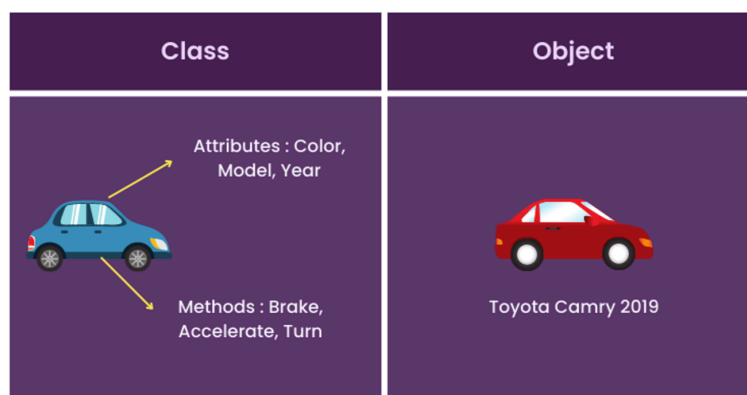
1.2. Istilah-Istilah dalam PBO





1.2.1. Dalam OOP, **class** digunakan untuk mendefinisikan dan mengelola *object*. Class merupakan *blueprint* dari sebuah objek atau cetakan untuk membuat objek.

Sebagai contoh sederhana, kita dapat mempertimbangkan class '**Mobil**' dalam konteks pemrograman. Class ini mungkin memiliki *attribute* seperti '**merk**', '**model**', dan '**tahun**', serta *method* seperti '**start()**' dan '**stop()**'.



NamaClass namaobject = new NamaClass();

```
class Kucing{  
    void lari(){  
    }  
    void makan_ikan(){  
    }  
}
```

→ Kucing anggora = new Kucing();



1.2.2. **Object** adalah instansi dari *class* ini, seperti '**mobilFerrari**' yang merupakan *object* dari *class* **Mobil**. Selain itu, bisa juga kita ibaratkan dengan 'Anggora' yang merupakan object dari class **Kucing** seperti pada kode program di atas.

```
class MyClass{
    public static void main(String [] args){
        Kucing anggora = new Kucing();
        anggora.lari();
    }
}
```



1.2.3. **Method** adalah perilaku dari sebuah class. Bisa juga disebut sebagai tindakan yang bisa dilakukan oleh suatu class. Lalu *method* dalam *class* mendefinisikan aksi yang dapat dilakukan oleh *object*, seperti '**mobilFerrari.start()**', sementara *attribute* menyimpan data spesifik tentang *object*, seperti '**mobilFerrari.merk = "Ferrari"**'.

Bisa juga kita ibaratkan dengan aksi yang dilakukan oleh object 'Anggora' ialah seperti 'anggora.lari()' yang menandakan perilaku yang dilakukan anggora ialah lari.

▼ Aturan Penamaan Method



Dalam memberikan nama untuk method mengikuti aturan sebagai berikut:

- a). Merupakan kata kerja atau frase kata kerja (terdiri dari beberapa kata)
- b). Kata pertama menggunakan huruf kecil dan sisanya dari kata-kata awal dikapitalisasi
- c). Dimulai dengan abjad (a,b,c,d,..) atau karakter underscore (_) atau tanda dolar (\$). Tidak dapat dimulai dengan bilangan. Setelah karakter pertama, dapat diikuti kombinasi abjad, underscore, dolar atau bilangan.
- d). Tidak boleh menggunakan reserved keyword dalam Java (public, class, Boolean, ...)
- e). Tidak boleh menggunakan operator (+,-,=)



REFERENSI

Aziefah, & Nurmasari, J. (2013). Pemrograman Berorientasi Objek. Pekanbaru: Politeknik Caltex Riau.

Barclay, K., & Savage, J. (2004). Object Oriented Design with UML and Java. Elsevier.

Deitel, P., & Deitel, H. (2012). Java: How to Program (9 ed.). US: Prentice Hall.