

Class and Object

class and object

Class

- Merupakan **RANCANGAN** cetakan atau blueprint dari sebuah object
- Memuat semua data/atribut dan method/fungsi yang bersifat umum

Deklarasi Class

```
class NamaClass {  
    tipe_data namaVariabel1;  
    .  
    .  
    tipe_data namaVariabelN;  
  
    tipe_data namaMethod1(parameter) {  
        // implementasi Method1  
    }  
    .  
    .  
    tipe_data namaMethodN(parameter) {  
        // implementasi MethodN  
    }  
}
```

Objek

- Adalah **HASIL NYATA**
- Inisialisasi atau instance dari sebuah class
- Punya 2 karakteristik :
 - *State* atau atribut
 - *Behaviour* atau perilaku
- Inisialisasi sebuah objek ditandai dengan keyword *new*



- Misalnya :

- Kelas : Rumah

- Objek : Rumah A, Rumah B, Rumah C

Atau

- Kelas : Dosen

- Objek : Ramos, Marcell, Joko

Contoh...

```
public class Rumah {  
    int noRumah;  
    String alamat;  
  
    public void diBersihkan() {  
        System.out.println("Disapu dulu ahhhhh.....");  
    }  
}
```

- `int noRumah` dan `String alamat` merupakan sebuah atribut.
- `public void dibersihkan()` adalah sebuah method.



```
public class Objek {  
    public static void main(String[] args) {  
        Rumah obj = new Rumah();  
        obj.alamat = "Perum Kota Baru 86";  
        obj.noRumah = 86;  
  
        System.out.println("Alamat : "+obj.alamat+"\nNo : "+obj.noRumah);  
        obj.diBersihkan();  
    }  
}
```




- ***Rumah obj = new Rumah();***
- Maksudnya, kita membuat sebuah objek dari kelas ***Rumah*** dengan nama ***obj***.

Attribute

Attribute merupakan karakteristik yang menjelaskan keadaan / kondisi suatu object.

Attribute memiliki nama, nilai dan tipe yang spesifik.

Misalkan :

Manusia : tangan, telinga, mata, kaki, dll...

Mahasiswa : Nama, NIM, Alamat, dll...

Constructor

- Konstruktor merupakan method khusus yang digunakan untuk menginisialisasi obyek saat pertama kali dibentuk.
- konstruktor harus memiliki nama yang sama dengan nama classnya dan tidak memiliki nilai kembalian (tipe-data)
- Konstruktor dijalankan pada saat sebuah object diinisialisasi (menggunakan kata new)

Overloading

- **Salah satu implementasi dari Polimorfisme.**
- **Method dengan nama yang sama dalam satu kelas.**
- **Nama boleh sama tetapi daftar parameter harus beda.**

```
public class Buku {  
  
    String pengarang;  
    String judul;  
  
    public void setNilai(String param1) {  
        judul = param1;  
        pengarang = "Tidak dikenal";  
    }  
  
    public void setNilai(String param1, String param2) {  
        judul = param1;  
        pengarang = param2;  
    }  
  
    public void cetak() {  
        System.out.println("Judul = " + this.judul +  
            "\nPengarang = " + this.pengarang);  
    }  
}
```