

# MENGENAL OBJEK & CLASS

## 1. Paradigma Objek

- Paradigma adalah suatu cara pandang atau cara berpikir
- Paradigma objek adalah cara pandang yang memandang SEGALA SESUATU sebagai OBJEK
- Semua aspek dalam Java programming dapat dianggap sebagai objek, -kecuali TIPE DATA PRIMITIF-, karena semua library dan objek dalam Java memiliki akar awal class `java.lang.Object`
- Berbagai benda di sekitar kita adalah objek nyata yang dapat dilihat, seperti : kucing, meja, rumah, orang , dll

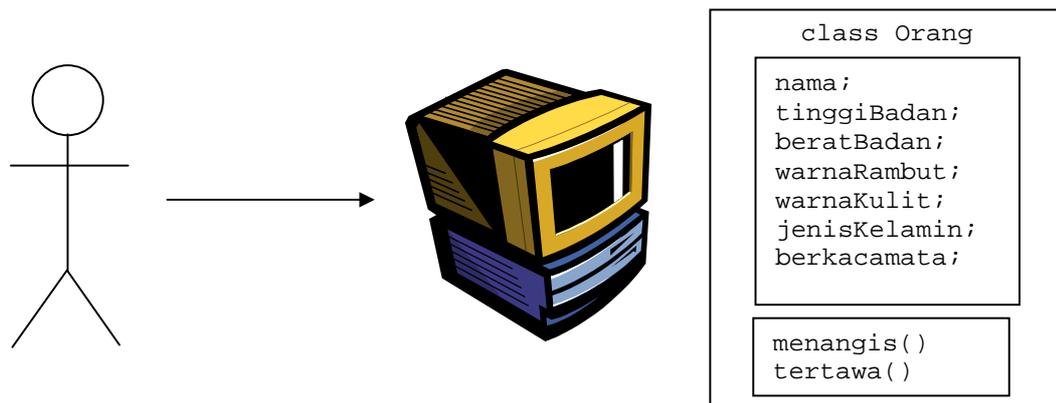
## 2. Konsep Objek & Class

- Java adalah merupakan OOP sehingga konsep objek dan class menjadi penting untuk dipahami
- OOP memiliki banyak kelebihan dibandingkan dengan bahasa prosedural, di antaranya :
  - Reusabilitas
  - Pembangunan program lebih cepat
  - Fleksibilitas lebih tinggi
  - Ekstensibilitas
  - Less maintenance
- Persoalannya, bagaimana memindahkan pemikiran objek di dunia nyata menjadi objek di dunia software atau pemrograman, khususnya Java
- Ambil contoh objek nyata yang akan dipindahkan adalah objek orang

## 3. Data Member

- Setiap objek yang dinamakan 'orang' pasti memiliki : nama, tinggi badan, berat badan, warna rambut, warna kulit, jenis kelamin, menggunakan kacamata, dll
- Ciri-ciri tersebut dapat dipindahkan menjadi variabel-variabel dari class yang sering disebut sebagai : data member
- Contoh pemisalan objek orang nyata menjadi kode program dalam class Orang :

```
class Orang {
    String nama;           //nama orang
    int tinggiBadan;      //dalam cm
    int beratBadan;       //dmlm kg
    String warnaRambut;   //hitam, pirang, coklat
    String warnaKulit;    //sawoMatang, hitam, putih
    String jenisKelamin;  //pria atau wanita
    boolean berkacamata;  //bila berkacamata berarti true
}
```



Gambar 1. Memindahkan orang dari dunia nyata menjadi class Orang

- Class dapat diumpamakan seperti spesifikasi atau blueprint. Dalam hal ini, Tuhan menciptakan manusia dengan spesifikasi tertentu, sehingga kita mengenal istilah SPESIES manusia
- Jadi dapat diumpamakan bahwa Tuhan memiliki class Orang yang kemudian membuat banyak objek dari class Orang tsb, dan contoh objek tersebut adalah Anda sendiri.
- Objek dalam pemrograman adalah objek yang dibuat dari class tertentu.
- Dari definisi class Orang di atas, kita bisa membuat objek-objek berdasar class tersebut.
- Objek-objek yang dibuat perlu disimpan dalam variabel yang akan menyimpan referensi/address dari objek yang dibuat.
- Proses pembuatan objek sering disebut sebagai instansiasi class, sedangkan objeknya disebut sebagai instance dari class
- Jadi, apa beda class dengan objek ??

#### 4. Method

- Selain memiliki atribut atau STATE yang diimplementasikan sebagai data member di atas, manusia juga dapat melakukan suatu aksi atau pekerjaan tertentu (BEHAVIOR)
- Contoh aksi/behavior yang umum adalah menangis dan tertawa
- Kedua behavior tsb bisa dipindahkan ke dalam bahasa pemrograman menjadi method sbb :

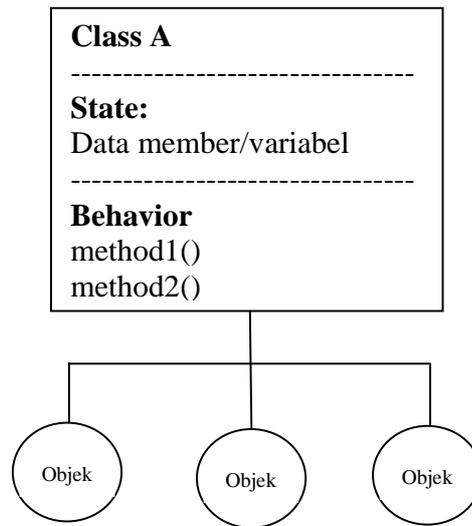
```

void menangis() {
    System.out.println("hik..hikk..hik...");
}

void tertawa() {
    System.out.println("ha..ha..ha..ha..");
}

```

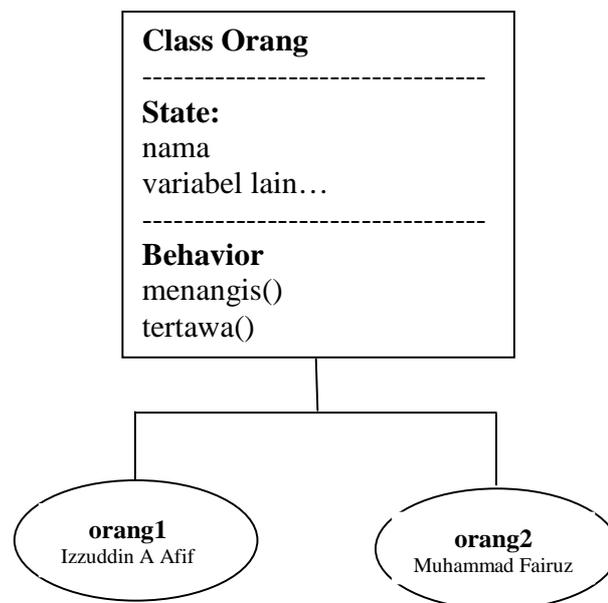
- Method merupakan perwujudan aksi atau tindakan dari dunia nyata di dalam pemrograman komputer.
- Method dalam dunia pemrograman juga “pasti melakukan sesuatu aksi”, misalnya menampilkan String di konsol → **do just one thing**



Gambar 2. Ilustrasi perbedaan antara class dan objek

- Dari gambar di atas dapat dipahami bahwa suatu class dapat memiliki banyak objek, dan setiap objek akan mewarisi data member dan method yang sama dari class
- Untuk membuat objek Orang dari class Orang, gunakan keyword **new** sbb :  

```
Orang orang1 = new Orang("Izzuddin A Afif");
Orang orang2 = new Orang("Muhammad Fairuz");
```
- Kesimpulannya : objek bertipe Orang dapat dibuat dari class Orang dan tiap objek perlu disimpan dalam variabel untuk menyimpan referensi/address dari lokasi di mana sebenarnya objek disimpan
- Semua objek memiliki data member yang sama tetapi setiap objek dapat memiliki state atau nilai data member yang berbeda (hanya nama dan tipe variabel yang sama)



Gambar 3. Ilustrasi pembuatan objek dari class