

# MODUL

# PEMOGRAMAN JAVA

Oleh:

Chalifa Chazar

# Pertemuan 7

## Object

### Tujuan:

- » Mahasiswa mampu memahami konsep pemograman berorientasi object pada Java
- » Mahasiswa mampu menggunakan konsep pemograman berorientasi pada pemograman Java

### Pustaka:

- » Bambang Hariyanto (2014): **Esensi-Esensi Bahasa Pemograman Java (Revisi Keempat)**. Informatika. Bandung.

Masih ingat ada pembahasan sebelumnya, pada konsep OOP (Object Oriented Programming), fungsi dan variabel dibungkus dalam sebuah object atau class yang dapat saling berinteraksi, sehingga membentuk sebuah program.

### 1. Pengertian Object

Object secara lugas dapat diartikan sebagai instansiasi atau hasil ciptaan dari suatu class. Asumsikan bahwa cetakan kue ada sebuah kelas, maka kue yang dihasilkan dari cetakan tersebut merupakan object dari kelas cetakan kue.

Sebuah object dapat terdiri dari object-object lain, atau dapat juga merupakan turunan dari object lain, sehingga mewarisi sifat-sifat induknya dan memiliki sifat tambahan. Contoh object dalam dunia nyata misalnya mobil, sapi, mahasiswa, meja, pohon, dsb. Semua object di dunia nyata yang memiliki sifat dan tingkah laku, dapat direpresentasikan ke dalam kode.

Object berisi data dan fungsi. Variabel dalam object akan menyimpan data dari object. Sedangkan fungsi akan menentukan operasinya.

## 2. Penciptaan Object

Untuk melakukan instansiasi/menciptakan sebuah object baru, digunakan keyword **new**. Perhatikan contoh berikut ini.

```
Manusia Mahasiswa = new Manusia();
```

Contoh di atas memiliki arti bahwa kita meng-instansiasi kelas manusia ke dalam object baru bernama mahasiswa.

Setelah membuat object, kita dapat mengakses atribut dan metode dari object tersebut. Perhatikan contoh berikut ini.

```
Mahasiswa.perwalian();  
Mahasiswa.nilai();
```

Tanda titik (.) berfungsi untuk mengakses atribut dan metode.

### Keyword This

Di dalam Java terdapat suatu besaran referensi khusus yang disebut **this**, yang digunakan di dalam method yang dirujuk untuk objek yang sedang berlaku. Nilai **this** merujuk pada objek di mana method yang sedang berjalan dipanggil.

## 3. Enkapsulasi

Enkapsulasi merupakan proses pemaketan objek beserta methodnya untuk menyembunyikan rincian implementasi dari pemakai/objek lainnya. Inti dari enkapsulasi atau pengkapsulan adalah ketidaktahuan apa yang ada dalam suatu objek dan bagaimana pengimplementasiannya. Yang dibutuhkan hanyalah apa kegunaan, bagaimana cara memakainya dan apa yang akan terjadi.

Dengan enkapsulasi, maka programmer tidak dapat mengakses suatu atribut yang dimiliki oleh suatu class. Kemampuan ini ditujukan untuk mendapatkan desain suatu software yang baik dan untuk keamanan software itu sendiri. Segala yang tidak perlu diketahui oleh yang lain, tidak perlu di-publish.

Salah satu implementasi dari enkapsulasi adalah adanya setter dan getter untuk suatu atribut dalam suatu kelas. Jika pada suatu kelas terdapat atribut a dan b, maka terdapat method setA-getA dan setB-getB. Bentuk lain dari enkapsulasi adalah memasukkan nilai atribut dengan menggunakan konstruktor.

### Latihan

1. Buatlah sebuah program untuk menentukan apakah sebuah bilangan termasuk ke dalam salah satu kriteria berikut:

- a. Positif Genap
- b. Negatif Genap
- c. Positif Ganjil
- d. Negatif Ganjil

Gunakan prinsip class & object dengan sebuah konstruktor untuk mengeset bilangan.

2. Buatlah program untuk meng-generate deret dengan ketentuan terdapat suatu:

- a. Nilai awal
- b. Beda
- c. Jumlah kemunculan angka pada deret

Program akan menghitung nilai rata-rata dari deret tersebut.

Contoh:

Masukkan jumlah kemunculan deret: 4

Deret: 2 5 8 11

Gunakan prinsip class & object dengan sebuah konstruktor untuk mengeset hal yang diketahui untuk menghasilkan deret.

----- **GOOD LUCK** -----