

Jurusan Teknik Informatika
- [Lab TIA]

[Praktikum PBO]

[Modul 2]

2018 Ganjil,
Moch. Kautsar Sophan

Object dan Class

Konsep Objek dan Class

Object dan Class adalah 2 hal yang berbeda. Object bukan Class, dan Class bukanlah Object. Object dibuat dari sebuah Class. Dari 1 buah Class, bisa dibuat banyak object.

Dalam disain dan pemrograman berorientasi objek, semua entitas dibentuk menggunakan konsep objek. Sebagai contoh dalam aplikasi sistem informasi akademik, ada objek mahasiswa, dosen, mata kuliah, dll. Dalam pemrograman berorientasi objek, yang diprogram adalah bagaimana objek saling berkomunikasi, dan melakukan pekerjaannya.

Sebagai contoh, pada pabrik mobil yang memproduksi mobil jenis X, Y, dan Z. Bila bagian produksi diperintahkan untuk memproduksi setiap jenis tersebut, maka pastinya mereka telah memiliki spesifikasi detail setiap jenis mobil tersebut barulah bisa memproduksi mobil sebenarnya. Jika dianalogikan pada konsep objek, maka spesifikasi jenis mobil adalah *class*, dan mobil yang diproduksi adalah *object* dari *class* tersebut. Atau dengan istilah sederhana, bisa diasumsikan bahwa *class* adalah cetakannya dan *object* adalah hasil cetakan yang dibuat.

Rancangan Object dan Class

Dalam mata kuliah ini, berbagai model rancangan berorientasi objek akan disajikan sebagai diagram UML. Terdapat beberapa diagram yang biasa digunakan dalam rancangan berbasis objek:

- Struktur *class* direpresentasikan sebagai *class diagram*.
- Perilaku *object* direpresentasikan sebagai *activity diagram*.
- Interaksi antar object direpresentasikan sebagai *sequence diagram*.

Dalam bagian ini akan diajarkan bagaimana menerjemahkan rancangan class ke dalam bahasa pemrograman.

Struktur Class

Seperti yang telah disebutkan, struktur class dapat direpresentasikan sebagai *class diagram*.



Gambar 1 Contoh class diagram

Pada [Gambar 1](#) ditunjukkan contoh *class diagram* yang berisi dua buah *class* yaitu *Nasabah* dan *Rekening*.

Implementasi Rancangan

Seperti yang telah dijelaskan pada konsep *object* dan *class*, sebelum membuat sebuah *object*, harus dibuat *class* (cetakannya) terlebih dahulu. Pada bahasa pemrograman Java ada beberapa aturan dalam pembuatan *class*:

- Nama:
 - Nama setiap class harus diawali dengan huruf besar
 - Karakter pertama harus huruf, contoh yang salah: 8Nasabah, _Nasabah
 - Tidak boleh ada spasi, bila lebih dari satu kata sebaiknya menggunakan camelCase
 - Contoh yang salah: Nasabah Bank
 - Contoh yang benar: NasabahBank
 - Contoh yang benar tapi tidak dianjurkan: Nasabahbank
- Setiap class harus disimpan pada sebuah file dengan nama yang bersesuaian (ada beberapa pengecualian yang akan dijelaskan kemudian), contoh: class *Nasabah* harus disimpan pada file dengan nama *Nasabah.java*.

Syntax untuk membuat class pada Java adalah sebagai berikut:

```
public class NamaClass {
    // ...
}
```

Untuk contoh pada [Gambar 1](#), *class* dapat dibuat sebagai berikut:

- Pada file *Nasabah.java*

```
public class Nasabah {
    // ...
}
```

- Pada file *Rekening.java*

```
public class Rekening {
    // ...
}
```

```
}
```

Dari *class* tersebut barulah dapat dibuat *object*-nya. *Object* dalam Java adalah sebuah variable yang mempunyai tipe data berupa *class* yang bersangkutan. Syntax untuk membuat object adalah sebagai berikut:

```
NamaClass namaVariabelObject = new NamaClass();
```

Contoh bila ingin dibuat *object* dari *class* Nasabah dengan nama variable x:

```
Nasabah x = new Nasabah();
```

Dimanakah suatu objek dapat dibuat? Akan dibahas pada pertemuan tentang Method.

Tugas Praktikum

1. Buat project dengan nama Modul2.
2. Dalam project Modul2 buat dua buah class dengan nama Nasabah dan Rekening.
3. Dalam class Nasabah, terdiri dari properti Nama, Alamat, No Telepon, dan Rekening. Pada class Rekening, terdapat properti NoRekening, dan Saldo
4. Pada class rekening, terdapat metode untuk tambah_rekening, dan ambil_rekening.
5. Simulasikan,
 - a. Buat 1 object nasabah, lakukan tambah rekening, cek saldo, ambil rekening, cek saldo
 - b. Buat 1 object nasabah lain, lakukan tambah rekening, cek saldo, ambil rekening, cek saldo
 - c. Tampilkan data 2 nasabah tsb (nama, alamat, noTelepon, saldo)

Referensi

- Objects First with Java - A Practical Introduction using BlueJ - , David Barnes & Michael Kolling, Prentice Hall, 2002
- Head First JAVA, 2nd Edition, Kathy Sierra & Bert Bates, O'Reilly
- Java 6 in 21 Days, Rogers Cadenhead & Laura Lemay, SAMS, 2007