

Jurusan Teknik Informatika
- [Lab TIA]

[Praktikum PBO]

[Modul 2]

2020 Ganjil,
Yoga Dwitya Pramudita

Konsep State dan Behavior

Dalam konsep pemrograman berorientasi objek, karakteristik dari suatu objek ditentukan berdasarkan state dan behavior yang dimilikinya.

Definitions

State

"the particular condition that someone or something is in at a specific time"
(kondisi tertentu dari seseorang atau sesuatu pada suatu saat tertentu)

State menunjukkan kondisi objek, menunjukkan informasi yang disimpan oleh sebuah objek.

Behavior

"the way in which one acts or conducts oneself, especially toward others"
(cara dalam bertindak atau menunjukkan dirinya, terutama kepada pihak lainnya)

Behavior adalah cara suatu objek memberikan respon kepada objek lainnya. Untuk memberikan respon, di implementasikan dalam bentuk method. Method bisa dalam bentuk fungsi atau prosedur.

Contoh

Objek: Rekening

State: jumlah saldo

Behavior:

- Bila saldo == 0, maka penarikan tunai oleh Nasabah pemilik rekening akan ditolak
- Bila saldo > 0, maka penarikan tunai oleh Nasabah pemilik rekening dapat dilakukan selama tidak melebihi saldo

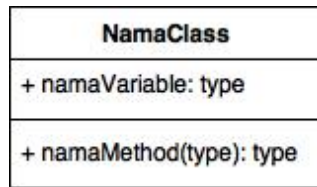
Disain dan implementasi

Disain (UML diagram)

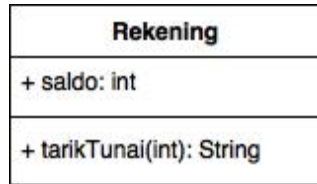
Dalam PBO, terutama Java

- **State** suatu objek disimpan sebagai **instance variable (field)**.

- **Behavior** suatu objek dispesifikasikan sebagai **instance method**.



Gambar 1 Notasi class dengan instance variable dan instance method



Gambar 2 Contoh class dengan instance variable dan instance method

Dalam UML class diagram, instance variable dan instance method dapat disertakan pada diagram melalui notasi seperti pada [Gambar 1](#). Setiap class dapat memiliki 0..n instance variable, dan 0..n instance method.

Setiap instance variable harus memiliki nama dan tipe data. Instance variable juga dapat memiliki nilai awal. Contoh:

- `saldo: int = 2000`

Setiap instance method harus memiliki nama. Bila memiliki parameter maka perlu disebutkan tipe data parameter-parameternya. Bila memiliki return value maka perlu disebutkan tipe data dari return value-nya. Contoh:

- `tarikTunai(): String` // tdk memiliki parameter
- `tarikTunai()` // tdk memiliki parameter dan return value

Tanda + adalah access modifier, penjelasan tentang access modifier ada pada materi berikutnya.

Pada disain tahap awal, diperkenankan untuk hanya menyebutkan nama dari instance variable dan instance method.

Implementasi

Instance variable dan instance method, adalah bagian dari class. Oleh karena itu, deklarasinya harus ada dalam lingkup class. Syntax deklarasi-nya adalah sebagai berikut:

```

public NamaClass {

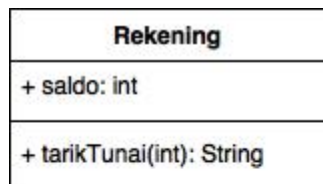
    // deklarasi instance variable
    accessModifier tipeData namaVariable;

    // deklarasi instance variable dg nilai awal
    accessModifier tipeData namaVariable = nilaiAwal;

    // deklarasi instance method tanpa parameter dan tanpa
    // return value
    accessModifier void namaMethod() {
        // isi method
    }

    // deklarasi instance method dengan parameter dan dengan
    // return value
    accessModifier tipeReturnValue namaMethod(tipeParam namaParam, tipeParam namaParam) {
        // isi method
    }
}

```



```

public class Rekening {

    public int saldo;

    public String tarikTunai(int jml) {
        // isi method
    }

}

```

Gambar 3 Contoh implementasi class diagram

Pada [Gambar 3](#), ditunjukkan contoh implementasi dari class diagram menjadi kode Java. Tanda + pada diagram adalah simbol dari access modifier **public**.

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