

VERSION 1.2  
FEBRUARI, 2023



# PEMROGRAMAN BERORIENTASI OBJEK

MODUL 3 – ENCAPSULATION

DISUSUN OLEH:

Muhammad Nizar Zulmi Rohmatulloh

Jody Yuantoro

DIAUDIT OLEH:

Aminudin, S.Kom., M.Cs.

PRESENTED BY: TIM LAB. IT

UNIVERSITAS MUHAMMADIYAH MALANG

## PEMROGRAMAN BERORIENTASI OBJEK

---

### TUJUAN

1. Mahasiswa mampu membangun aplikasi sederhana menggunakan paradigma pemrograman terstruktur
  2. Mahasiswa mampu membangun aplikasi sederhana menggunakan paradigma object oriented.
- 

### TARGET MODUL

1. Mahasiswa mampu memahami konsep Encapsulation
  2. Mahasiswa mampu memahami dan menerapkan Object Oriented Programming.
- 

### PERSIAPAN

1. Java Development Kit.
  2. Text Editor / IDE (Visual Studio Code, Netbeans, IntelliJ IDEA, atau yang lainnya).
- 

### KEYWORDS

- Pemrograman Terstruktur
- Pemrograman Berorientasi Objek
- Encapsulation
- OOP
- Access Modifier

## TEORI

### • Konsep Encapsulation

Encapsulation adalah pembungkus, encapsulation pada object oriented maksudnya adalah membungkus class dan menjaga apa apa saja yang ada di dalam class tersebut, baik method ataupun atribut, agar tidak dapat diakses oleh class lainnya. Untuk menjaga hal tersebut dalam Encapsulation dikenal nama Hak Akses Modifier yang terdiri dari Private. Public dan Protected.

Fungsi dari access modifier pada Java adalah untuk membatasi scope dari sebuah class, constructor, variabel, method, atau anggota data lainnya yang terdapat dalam suatu program Java.

Modifier	Class	Package	Subclass	World
public	True	True	True	True
protected	True	True	True	False
no modifier	True	True	False	False
private	True	True	False	False

#### a. Public

Memberikan hak akses kepada atribut atau method agar bisa diakses oleh siapapun (property atau class lain diluar class yang bersangkutan), artinya method atau atribut yang ada di class A dapat diakses oleh siapapun baik itu class A, class B dan seterusnya.

#### b. Protected

Memberikan hak akses pada class itu sendiri dan class hasil turunannya (inheritance), artinya apa saja yang di class A hanya bisa diakses oleh class A sendiri dan class yang meng extends class A. Namun harus dipahami class lain yang berada dalam satu package dengan class A mampu mengakses tipe data protected, sedangkan yang tidak mampu mengakses adalah class - class yang berada diluar

package class A. Untuk dapat mengaksesnya, class yang berada diluar package class A harus meng extends class A.

c. Private

Memberikan hak akses hanya pada class itu sendiri, artinya apa-apa saja yang ada di dalam class A baik itu method apapun atribut hanya bisa diakses oleh class A saja, class lain tidak bisa mengaksesnya.

- **Konsep Pemrograman Terstruktur**

Pemrograman Terstruktur merupakan suatu tindakan untuk membuat program yang berisi instruksi-instruksi dalam bahasa komputer yang disusun secara logis dan sistematis supaya mudah dimengerti, mudah dites, dan mudah dimodifikasi. Prinsip dari pemrograman terstruktur adalah Jika suatu proses telah sampai pada suatu titik ataupun langkah tertentu, maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya ataupun kembali ke baris sebelumnya, kecuali pada langkah - langkah untuk proses pengulangan atau berulang (Loop).

Contoh Program dengan konsep Pemrograman Terstruktur:

```

no usages
1 ► public class Modul3 {
    no usages
2 ►     public static void main(String[] args) {
3
4         int sisi = 10;
5         int luas = sisi * sisi;
6         int keliling = sisi * 4;
7
8         System.out.println("Luas Persegi: " + luas);
9         System.out.println("Keliling Persegi: " + keliling);
10    }
11 }
12

```

- **Konsep Pemrograman Berorientasi Objek**

Pemrograman Berorientasi Objek (PBO) atau dalam bahasa inggris disebut Object Oriented Programming (OOP) merupakan sebuah paradigma pemrograman yang

berorientasikan kepada objek. Semua data dan fungsi di dalamnya dibungkus dalam suatu kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Contoh Program dengan konsep Pemrograman Berorientasi Objek:

```

1  ► 2 usages
    public class Modul300P {
        2 usages
2      ... private int sisi, luas, keliling;
        no usages
3  ►  ... public static void main(String[] args) {
4      ...     Modul300P mainObj = new Modul300P();
5      ...     mainObj.setSisi(10);
6      ...     mainObj.setLuas(mainObj.getSisi());
7      ...     mainObj.setKeliling(mainObj.getSisi());
8      ...     System.out.println("Luas Persegi: " + mainObj.getLuas());
9      ...     System.out.println("Keliling Persegi: " + mainObj.getKeliling());
10     ... }

        2 usages
11     ... public int getSisi() {
12     ...     return sisi;
13     ... }

        1 usage
14     ... public void setSisi(int sisi) {
15     ...     this.sisi = sisi;
16     ... }

        1 usage
17     ... public int getLuas() {
18     ...     return luas;
19     ... }

        1 usage
20     ... public void setLuas(int sisi) {
21     ...     this.luas = sisi * sisi;
22     ... }

        1 usage
23     ... public int getKeliling() {
24     ...     return keliling;
25     ... }

        1 usage
26     ... public void setKeliling(int sisi) {
27     ...     this.keliling = sisi * 4;
28     ... }
29 }
30

```

- **Perbedaan Pemrograman Terstruktur dan Pemrograman Berorientasi Objek**

Perbedaan mendasar antara Pemrograman Berorientasi Objek dan Pemrograman Terstruktur adalah dengan menggunakan Pemrograman Berorientasi Objek maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sedangkan untuk Pemrograman Terstruktur, menggunakan prosedur/tata cara yang teratur untuk mengoperasikan data struktur.

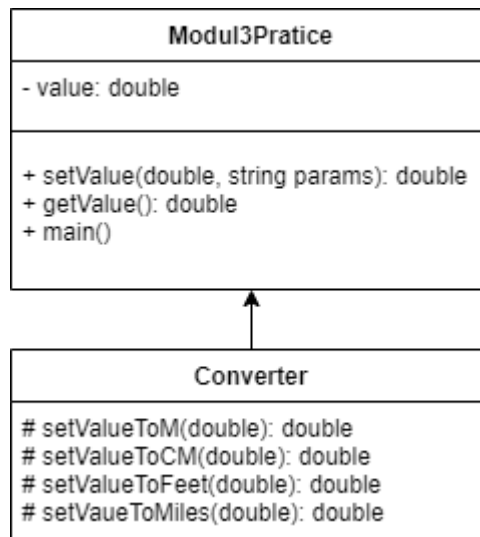
Untuk program yang simpel/ sederhana biasanya menggunakan pemrograman terstruktur karena masih mudah dan tidak banyak dilakukan perubahan yang berarti, sedangkan untuk line lebih dari 100 atau bisa dikatakan rumit, maka digunakan pemrograman berorientasi objek. Pemrograman Terstruktur terdiri dari pemecahan masalah yang besar menjadi masalah yang lebih kecil dan seterusnya, sedangkan untuk pemrograman berorientasi objek terdiri dari pengelompokan kode dengan data yang mana setiap objek berfungsi secara independen sehingga untuk setiap perubahan kode tidak tergantung pada kode yang lainnya, atau lebih dikenal dengan modular. Terdapat juga perbedaan secara spesifik antara Pemrograman Berorientasi Objek dengan Pemrograman Terstruktur, yaitu pada kelas dan objek. Pada Pemrograman Terstruktur tidak terdapat kelas dan objek.

Pada Terstruktur terdapat "function", di OOP terdapat "method". Kalau di Terstruktur terdapat "modules", di OOP terdapat "objects". Pada di Terstruktur terdapat "argument", di OOP terdapat "message". Begitu juga dengan "variabel" yang terdapat di Terstruktur, di OOP lebih dikenal dengan nama "atribut".

## CODELAB

### a. Konversi Satuan Imperial ke Metrik

Buatlah program konversi satuan imperial ke metrik sederhana dengan menerapkan konsep pemrograman OOP yang sesuai dengan diagram dibawah ini, mulai dari enkapsulasi class, penamaan method dan atributnya.



### Contoh program:

```

Run
C:\Users\xyzuan\.jdk\openjdk-19.0.1\bin\java.exe -javaagent:C:\Users\xyzuan\AppData\Local\JetBrains\Toolbox\app
Yard Converter to m, cm, feet & miles
1. Yard to M
2. Yard to CM
3. Yard to Feet
4. Yard to Miles
Pilih menu: 2
Masukkan Yard: 10
Hasil konversi: 914.4
Process finished with exit code 0
  
```

## TUGAS

Buatlah sebuah kelas *public* bernama User yang akan mempresentasikan seorang user di dalam sistem informasi akademik. Sebuah User harus menyimpan informasi tentang `namaUser(String)`, `passwordUser(String)`, `roleUser(int)`, dan `statusUser(boolean)`. Seluruh atribut tersebut bersifat *private*. Buatlah juga setter getter untuk atribut tersebut yang bersifat *public*.

Pada class User akan terdapat method sebagai berikut:

- **login()**, akan digunakan sebagai laman tampilan otoritas login sederhana yang membutuhkan inputan `namaUser` dan `passwordUser` untuk divalidasi, yang kemudian *redirect* ke dalam method **dashboard()** sesuai dengan role user tersebut. Apabila `roleUser` bernilai 0, maka user tersebut adalah sebagai admin, jika `roleUser` bernilai 1 maka user tersebut adalah sebagai mahasiswa. Default User Admin adalah : admin, admin, dan default password mahasiswa adalah UMM\_a2022.
- **dashboard()**, method ini akan digunakan sebagai laman dashboard utama yang menampilkan berbagai menu sesuai dengan `roleUser`. Apabila `roleUser` adalah Admin, maka akan terdapat menu update status mahasiswa, update password mahasiswa dan logout. Jika `roleUser` adalah sebagai mahasiswa, maka cukup menampilkan informasi user dan menu untuk mengubah password dan logout.
- **updateStatus()**, method ini digunakan untuk mengupdate status mahasiswa berdasarkan `namaUser` yang hanya boleh digunakan oleh admin.
- **updatePassword()**, method ini digunakan untuk mengubah password user boleh digunakan oleh admin maupun mahasiswa.
- **isPassSymbols(String)**, method ini digunakan untuk mengecek inputan password yang terdapat symbols, memiliki parameter string untuk argumen inputan password dan method ini akan mengembalikan nilai dalam bentuk boolean.



- **isPassDigit(String)**, method ini digunakan untuk mengecek inputan password terdapat angka numerik, memiliki parameter string untuk argumen inputan password dan method ini akan mengembalikan nilai dalam bentuk boolean.
- **isPassUpperCase(String)**, method ini digunakan untuk mengecek inputan password terdapat karakter Uppercase / huruf besar, memiliki parameter string untuk argumen inputan password dan method ini akan mengembalikan nilai dalam bentuk boolean.
- **isPassLowerCase(String)**, method ini digunakan untuk mengecek inputan password terdapat karakter LowerCase / huruf kecil, memiliki parameter string untuk argumen inputan password dan method ini akan mengembalikan nilai dalam bentuk boolean.

Ketentuan dari Atribut Data User Mahasiswa adalah sebagai berikut :

- database User yang mencakup atribut namaUser, passwordUser, roleUser, dan statusUser disimpan dalam bentuk object of array multidimensi.

	<b>0 - namaUser</b>	<b>1 - passwordUser</b>	<b>2 - roleUser</b>	<b>3 - statusUser</b>
<b>0</b>	admin	admin	0	true
<b>1</b>	202110370311147	UMM_a2021	1	true
<b>2</b>	202110370311145	UMM_a2021	1	false

- **passwordUser**, harus terdapat minimal 1 UpperCase 1 LowerCase, 1 Simbol, dan Kombinasi Angka dan Huruf.

**Catatan :**

- Improvisasi program boleh dilakukan asal sesuai dan tidak mengurangi requirement program yang telah ditentukan.

**Diagram:**

User
- admin : boolean - userIndex : int - userDB : Object[][]
+ login() : void + dashboard() : void + updateStatus() : void + updatePassword() : void + isPassSymbols(String) : boolean + isPassDigit(String) : boolean + isPassUpperCase(String) : boolean + isPassLowerCase(String) : boolean + isAdmin() : boolean + setAdmin(boolean) : void + getUserIndex() : int + setUserIndex(int) : void + getUserDB() : Object[][] + setUserDB(Object[][]) : void

**Contoh Output:**

```

Run
C:\Users\xyzuan\.jdk\openjdk-19.0.1\bin\java.exe -javaagent:C:\Users\xyzuan\AppData\Local\JetBrains\Toolbo
Masukkan username: 202110370311147
Masukkan password: UMM_a2022
Sistem Akademik UMM | Mahasiswa
Username : 202110370311147
Status Mahasiswa : true
1. Update Password
0. Logout
Pilih menu: 1
Input Password Baru: uU
Password harus kombinasi uppercase lowercase, terdapat angka, dan symbols
Sistem Akademik UMM | Mahasiswa
Username : 202110370311147
Status Mahasiswa : true
1. Update Password
0. Logout
Pilih menu: 1
Input Password Baru: uUmbek2_
Password berhasil diubah
Sistem Akademik UMM | Mahasiswa
Username : 202110370311147
Status Mahasiswa : true
1. Update Password
0. Logout
Pilih menu: 0
Masukkan username: admin
Masukkan password: admin
Sistem Akademik UMM | Admin
1. Update Status Mahasiswa
2. Update Password Mahasiswa
0. Logout
Pilih menu: 1
Masukkan namaUser: 202110370311147
Ganti Status Mahasiswa:
1. Aktif
2. Tidak Aktif
Pilih menu: 2

Sistem Akademik UMM | Admin
1. Update Status Mahasiswa
2. Update Password Mahasiswa
0. Logout
Pilih menu: 0
Masukkan username: 202110370311147
Masukkan password: uUmbek2_
Sistem Akademik UMM | Mahasiswa
Username : 202110370311147
Status Mahasiswa : false
1. Update Password
0. Logout
Pilih menu:

```

## RUBRIK PENILAIAN

Aspek Penilaian	Poin
Codelab	20
Tugas Modul	35
Pemahaman	45
<b>Total</b>	<b>100</b>