

VERSION 1.1
4 MARET, 2022



PEMROGRAMAN BERORIENTASI OBJEK

MEMBANGUN APLIKASI OBJECT ORIENTED SEDERHANA, ENCAPSULATION -MODUL 2

DISUSUN OLEH :

- Muhammad Rahadian Arya Saputra
- Riyan Putra Firjatullah

DIAUDIT OLEH :

- Ali Sofyan Kholimi, M.Kom.

PRESENTED BY: TIM LAB-IT

UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN BEROIRENTASI 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 & menerapkan Object Oriented Programming

PERSIAPAN SOFTWARE/APLIKASI

1. Compiler java (JDK), JRE.
2. Editor Java (NetBeans, Gel, Eclipse, Jcreator, dll).

KEYWORDS

Pemrograman Terstruktur

Pemrograman Berorientasi Objek

Encapsulation

OOP

PERSIAPAN MATERI

Encapsulation

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

Modifier	Same Class	Same Package	Subclass	Other Packages
Public	Y	Y	Y	Y
Protected	Y	Y	Y	N
No Modifier	Y	Y	N	N
Private	Y	N	N	N

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 siapaun baik itu class A, class B dan seterusnya.

b. Protected

Memberikan hak akses kepada class itu sendiri dan class hasil turunannya (inheritance), artinya apa apa saja yang ada 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 ataupun atribut hanya bisa diakses oleh class A saja, class lain tidak bisa mengaksesnya.

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).

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.

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 pengkelompokan 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".

KEGIATAN PERCOBAAN

PERCOBAAN 1

Encapsulation

Buat 3 buah class yang atributnya menggunakan akses modifier protected.

```
public class Cavalry {  
    protected String unitName = "Cavalry";  
    protected int unitNumber;  
  
    Cavalry (int unitNumber) {  
        this.unitNumber = unitNumber;  
    }  
  
    public void display () {  
        System.out.println("Unit Number " + this.unitNumber + " is a " + unitName);  
    }  
}
```

```
public class Archer {  
    protected String unitName = "Archer";  
    protected int unitNumber;  
  
    Archer (int unitNumber) {  
        this.unitNumber = unitNumber;  
    }  
  
    public void display () {  
        System.out.println("Unit Number " + this.unitNumber + " is a " + unitName);  
    }  
}
```

```
public class Swordsman {  
    protected String unitName = "Swordsman";  
    protected int unitNumber;  
  
    Swordsman (int unitNumber) {  
        this.unitNumber = unitNumber;  
    }  
  
    public void display () {  
        System.out.println("Unit Number " + this.unitNumber + " is a " + unitName);  
    }  
}
```

Buat class main Militer untuk mengakses 3 class di atas.

```
public class Militer {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int i=0, numOfMiliter=0;

        Swordsman[] swordsman = new Swordsman[3];
        Archer[] archer = new Archer[3];
        Cavalry[] cavalry = new Cavalry[3];

        for (i=0; i<3; i++) {
            swordsman[i] = new Swordsman(++numOfMiliter);
        }

        for (i=0; i<3; i++) {
            archer[i] = new Archer(++numOfMiliter);
        }

        for (i=0; i<3; i++) {
            cavalry[i] = new Cavalry(++numOfMiliter);
        }
    }
}
```

```
        for (i=0; i<3; i++) {
            swordsman[i].display();
        }

        for (i=0; i<3; i++) {
            archer[i].display();
        }

        for (i=0; i<3; i++) {
            cavalry[i].display();
        }
    }
}
```

Karena sudah dienkapsulasi, data yang bersifat Protected dan Private tidak bisa diubah dari luar.

Ada dua keunggulan dari enkapsulasi:

- data yang tidak boleh diubah, tidak akan bisa diubah
- data yang bisa diubah, namun dengan format tertentu, maka programmer akan dipaksa memasukkan dengan format tersebut

`swordsman[1].unitName = "Tukang Pedang";` tidak bisa diubah karena variabel/atribut bersifat protected

PERCOBAAN 2

Pemrograman Terstruktur

Berikut contoh program terstruktur dengan studi kasus konversi mata uang, dari Dollar ke Rupiah Indonesia, Dollar ke Ringgit Malaysia, dan Dollar ke Yen Jepang.

```
public class konversiMataUang{
    Run | Debug
    public static void main(String[] args){
        int dollar = 2;
        double rupiah;
        double ringgit;
        double yen;

        rupiah = dollar * 14.36690;
        ringgit = dollar * 4.19;
        yen = dollar * 115.00;

        System.out.println("Hasil Konversi Mata Uang");
        System.out.println(dollar + " dollar = " + rupiah + " Rupiah");
        System.out.println(dollar + " dollar = " + ringgit + " Ringgit");
        System.out.println(dollar + " dollar = " + yen + " Yen");
    }
}
```

Output:

```
Hasil Konversi Mata Uang
2 dollar = 28.7338 Rupiah
2 dollar = 8.38 Ringgit
2 dollar = 230.0 Yen
```

PERCOBAAN 3

Pemrograman Berorientasi Objek

Dengan studi kasus yang sama dirubah ke pemrograman berbasis objek, terdapat 4 class yaitu Rupiah, Ringgit, Yen, dan Main. Pada setiap class terdapat metod untuk menghitung konversi mata uang.

Pertama kita buat class dan juga method untuk memasukkan rumus konversi

```
class Converter {

    public int getDollInRup(int doll) {
        return (doll* 1436690);
    }

    public int getDollInRing(int doll) {
        return (doll* 419);
    }

    public int getDollInYen(int doll) {
        return (doll*11500);
    }
}
```

Buat method lagi untuk menampilkan hasil konversi pada console di class yang sama

```
public void printDollInRup(int doll) {
    System.out.println(doll + " Dollar setara dengan " + this.getDollInRup(doll) + " Rupiah");
}

public void printDollInRing(int doll) {
    System.out.println(doll + " Dollar setara dengan " + this.getDollInRing(doll) + " Ringgit");
}

public void printDollInYen(int doll) {
    System.out.println(doll + " Dollar setara dengan " + this.getDollInYen(doll) + " Yen");
}
}
```

Pada class main kita panggil method yang telah kita buat pada class converter.

```
public class MateriOpp {
    Run | Debug
    public static void main(String[] args) {

        Converter konverter = new Converter(); // Membuat objek konverter
        int[] doll = new int[3]; // simpan semua dalam Dollar

        doll[0] = 10;
        doll[1] = konverter.getDollInRing(68);
        doll[2] = konverter.getDollInYen(1620);

        konverter.printDollInRup(doll[0]);
        konverter.printDollInRing(doll[0]);
        konverter.printDollInYen(doll[0]);
    }
}
```

Berikut referensi untuk menambah pemahaman kalian mengenai OOP :

<https://www.studytonight.com/java/object-and-classes.php>

LEMBAR KERJA

KEGIATAN 1

Buatlah sebuah program untuk menghitung nilai rata-rata mata kuliah dan mengecek kelulusannya. Apabila nilai rata-rata lebih dari 70 maka status “LULUS” dan apabila nilai rata-rata kurang dari 70 status “GAGAL”. Sertakan juga enkapsulasinya pada setiap atribut dan methodnya. Berikut adalah atribut dan method yang diperlukan :

Perhitungan
- NilaiProgdas : int - NilaiKalkulus : int - NilaiOrkom : int - Hasil : double
+ setNilaiProgdas() + getNilaiProgdas() + setNilaiKalkulus() + getNilaiKalkulus() + setNilaiOrkom() + getNilaiOrkom() + hitungRata() + cekNilai() + main()

Output:

```

Nilai Progdas : 80
Nilai Orkom : 80
Nilai Kalkulus : 80

Nilai Anda :
Progdas : 80
Orkom : 80
Kalkulus : 80

Rata-rata : 80.0
Status : LULUS
  
```

KEGIATAN 2

Buat aplikasi sederhana dengan menggunakan paradigma pemrograman terstruktur dan paradigma object oriented dengan studi kasus sebagai berikut :

- A. Konversi waktu bulan ke- (detik, menit, jam, hari). – easy
- B. Kalkulator menghitung keliling / luas belah ketupat – hard
- Terdapat 4 Inputan (nilai diagonal1, diagonal2, sisi, dan inputMenu)
 - Tersedia pilihan menu untuk menentukan hasil keliling atau luas.

Pilih salah satu dari kedua studi kasus tersebut untuk dikerjakan.

Catatan : Aplikasi yang dibangun adalah aplikasi yang sama.

*Jika ada source code yang identik, maka akan dilakukan pengurangan nilai.

RUBRIK PENILAIAN

Aspek Penilaian	Poin
Kegiatan 1	15
Kegiatan 2 A	15
Kegiatan 2 B	25
Pemahaman dan penerapan OOP	30
Ketepatan dalam menjawab pertanyaan	15
Total	100