

VERSION 2.0

FEBRUARI, 2023



PEMROGRAMAN BERORIENTASI OBJEK

MODUL 1 – OBJECT ORIENTED PROGRAMMING, CLASS DAN OBJECT

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 dapat memahami konsep dasar dari class dan object.
 2. Mahasiswa dapat memahami konsep OOP.
-

TARGET MODUL

1. Mahasiswa dapat menulis aplikasi berdasarkan class diagram yang dibuat.
 2. Mahasiswa dapat membuat class dan object dengan benar.
-

PERSIAPAN

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

KEYWORDS

- Class Diagram
- Class dan Object
- Konsep OOP
- OOP Java

TEORI

- **Konsep OOP**

Object-oriented programming atau OOP adalah suatu metode pemrograman yang berorientasi pada objek. Program-program yang telah ada merupakan gabungan dari beberapa komponen-komponen kecil yang sudah ada sebelumnya. Hal itu dapat mempermudah pekerjaan seorang programmer dalam melakukan pengembangan program. Objek-objek yang saling berkaitan dan disusun kedalam satu kelompok ini disebut dengan class. Nantinya, objek-objek tersebut akan saling berinteraksi untuk menyelesaikan masalah program yang rumit.

- **Class**

Class adalah “blueprint” atau “cetakan” untuk menciptakan suatu object. Class biasa dipergunakan untuk membungkus berbagai atribut dan method yang saling berhubungan menjadi sebuah grup agar lebih terorganisir sebagai satu kesatuan.

Pembuatan class:

```
public class namaClass {  
  
    // Dapat berisi atribut / variabel.  
    // Dapat berisi method.  
  
}
```

Contoh class:

```
public class petAnimals {  
  
    // Dapat berisi atribut / variabel.  
    boolean isFlying;  
  
    // Dapat berisi method.
```

```

void petSound(){
    System.out.println("miawwww...");
}

}

```

- **Object**

Object adalah hasil instantiate dari class. Karena class berbentuk “cetakan”, maka untuk mengambil isi cetakan tersebut kita wajib buat objectnya. Class dan Object adalah kedua hal yang saling berhubungan satu sama lain.

Pembuatan object:

```

public class classPertama {
    namaClassKedua namaObj = new namaClassKedua();
}

```

Contoh object:

```

public class Dog {
    pet myPet = new pet();
}

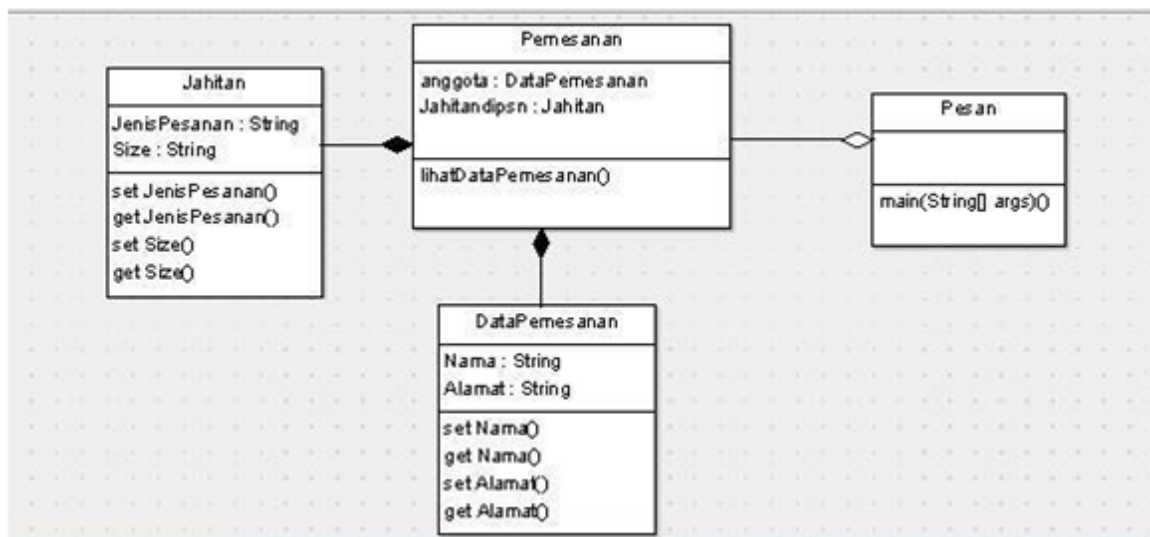
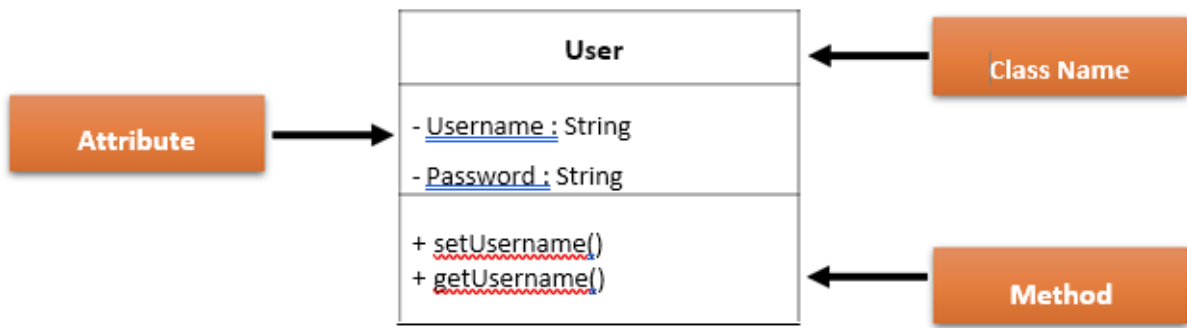
```

- **Class Diagram**

Class diagram merupakan salah satu diagram utama dari UML untuk menggambarkan class atau blueprint object pada sebuah sistem. Analisis pembentukan class diagram merupakan aktivitas inti yang sangat mempengaruhi arsitektur piranti lunak yang dirancang hingga ke tahap pengkodean. Class diagram ini terdiri atas 3 bagian dimulai dari kotak teratas sampai terbawah yaitu Nama Class, Properti atau Atribut, dan Method.

Komponen penyusun class Diagram:

Contoh class diagram:



- **Konvensi Penamaan Class, Method, dan Atribut**

Class menggunakan kata benda karena menunjukkan sebuah objek atau benda. Contoh: KTP, KTM, Data Penduduk, Data Mahasiswa, Manusia, Hewan, dll.

Attribute menunjukkan karakteristik yang dimiliki objek tersebut, sehingga biasanya menggunakan kata benda, kata adjektiva, kata adverbial dan kata nomina. Attribute juga bisa merupakan sebuah objek. Contoh attribute menggunakan kata benda: **nama** orang

pada data penduduk. Contoh attribute menggunakan kata adjektiva: **warna** kulit manusia. Contoh attribute menggunakan kata nomina: **tinggi** badan seseorang. Attribute biasanya memiliki modifier private. Attribute umumnya dirancang private agar tidak mengalami perubahan data tanpa melalui proses pengecekan. Bayangkan jika warna kulit tidak diisi warna, namun diisi kata sifat semisal: “Warna kulitnya gembira”

Method menggunakan kata kerja yang menunjukkan apa yang bisa dilakukan oleh atau terhadap objek tersebut, terutama terhadap atribut yang dimiliki objek tersebut. Contoh: Ubah data nama, tetapkan warna kulit menjadi coklat. Method biasanya memiliki modifier public. Semua perubahan pada nilai attribute sebaiknya dilewatkan melalui method. Method yang dilakukan untuk mengubah attribute biasanya disebut setter.

Library
- bookName : String - bookYear : int - bookAuthor : String
+ getBookName() + getBookYear() + getBookAuthor()

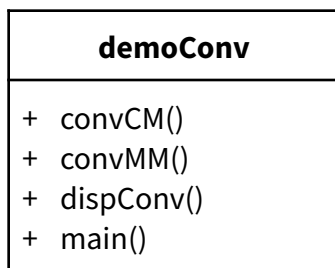
CODELAB

a. Membuat Class dan Object

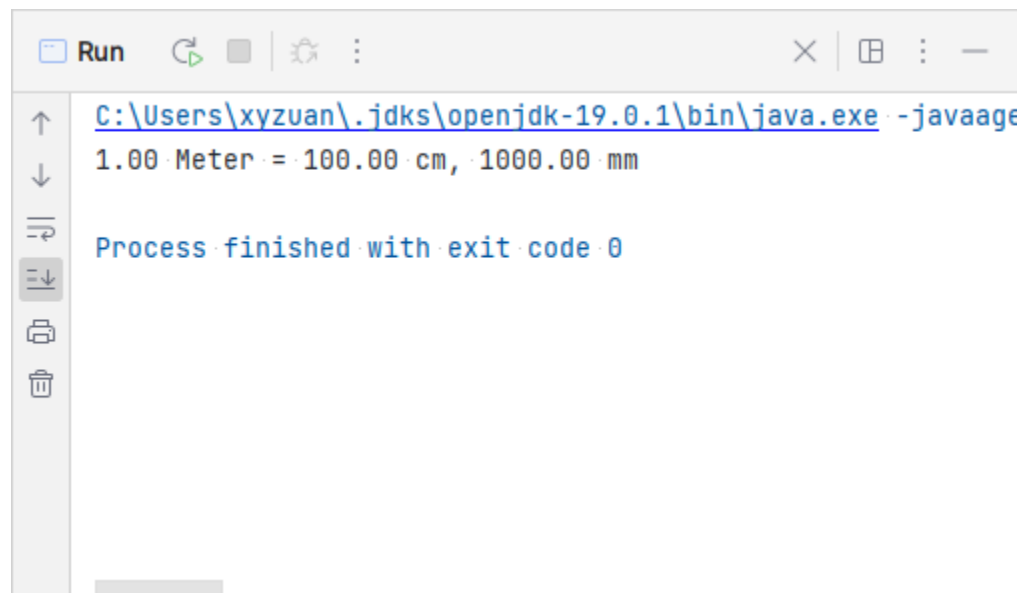
Buatlah sebuah class **demoConv** yang didalamnya memiliki method :

- **convCM** untuk mengubah nilai meter ke cm.
- **convMM** untuk mengubah nilai meter ke mm.
- **dispConv** untuk menampilkan hasil konversi nilai meter ke cm dan mm.

Diagram:



Contoh Output:



```
Run C:\Users\xyzuan\.jdk\openjdk-19.0.1\bin\java.exe -javaage
1.00 Meter = 100.00 cm, 1000.00 mm
Process finished with exit code 0
```

b. Program Menghitung Umur

Buatlah sebuah program penghitung umur berdasarkan tahun kelahiran, dengan memanfaatkan penggunaan library bawaan *java.time* & *java.util*.

Diagram:

demoCalcYearsOld
- yourBirthYear : int
+ getYourBirthYear() + setYourBirthYear() + main()

Contoh Output:

```
Run C:\Users\xyzuan\.jdk\openjdk-19.0.1\bin\java.exe -javaagent:C:\
Masukkan Tahun Kelahiran: 2002
Umur anda: 21
Process finished with exit code 0
```


TUGAS

KEGIATAN 1

Buatlah sebuah class **BookSystem** yang didalamnya memiliki atribut **nama**, **noTelp**, **alamat**, **email**, dan **room** yang dimana atribut tersebut bersifat private dan perlu terdapat method **Setter & Getter** untuk menginisialisasi dan mengaksesnya.

- **init()** digunakan untuk menginisialisasi attribut diatas dengan memanggil setter atributnya.

Inputan user terhadap atributnya memiliki ketentuan sebagai berikut:

- Input Nama harus berupa karakter / huruf
- Input No Telp harus sesuai format dengan kode negara diawal input, contoh **+6285155024678**
- Input Email harus disertai dengan domain yang valid, contoh (xyzuan@gmail.com, xyzuan@webmail.umm.ac.id)
- **dispPerson()** digunakan untuk menampilkan atribut identitas yang telah diinisialisasi dengan memanggil getter atributnya
- Atribut **room** dibuat array multidimensi untuk inisiasi ketersediaan kamar, no kamar dan harga, contoh :

	0 - available	1 - noRoom	2 - priceRoom
0	0 - true	101	1.000.000
1	1 - false	102	2.500.000

- **bookRoom()** digunakan untuk user memilih kamar yang telah diinisialisasi, kamar tidak bisa dipilih ketika kamar berstatus tidak tersedia (memiliki nilai 1 pada indeks kolom 0).
- **doPayment(int noRoom)** berisi tentang pembayaran dan informasi kamar yang akan dibayar oleh user pada method **bookRoom()**, method ini juga memiliki parameter yang

digunakan untuk menerima argument index **room** yang akan dipesan oleh user, dan ketika pembayaran berhasil maka status dari kamar yang dipesan harus berstatus tidak tersedia / 1.

Diagram :

BookSystem
<ul style="list-style-type: none"> - nama : String - noTelp : String - alamat : String - email : String - room[][] : int
<ul style="list-style-type: none"> + setNama(String nama) + getNama() + setNoTelp(String noTelp) + getNoTelp() + setAlamat(String alamat) + getAlamat() + setEmail(String email) + getEmail() + getRoom() + init() + dispPerson() + bookRoom() + doPayment(int noRoom) + main()

Catatan :

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

Contoh Output:

```

C:\Users\xyzuan\.jdk\openjdk-19.0.1\bin\java.exe
Registration
Masukkan Nama : Jody Yuantoro
Masukkan Email : xyzuan@webmail.umm.ac.id
Masukkan No Telp (+62) : +62851550252
Masukkan Alamat : Malang
Rayz Hotel
1. No. 101 (Tidak Tersedia)
2. No. 102
3. No. 103 (Tidak Tersedia)
Pilih kamar yang tersedia: 1
Mohon Maaf Kamar Tidak Tersedia
Ingin memesan kamar lagi? 0/1
0
Rayz Hotel
1. No. 101 (Tidak Tersedia)
2. No. 102
3. No. 103 (Tidak Tersedia)
Pilih kamar yang tersedia: 2
=====
Pembayaran Kamar
Nama : Jody Yuantoro
No Telp : +62851550252
Alamat : Malang
Email : xyzuan@webmail.umm.ac.id
No Kamar : 102
Harga Kamar : 2000000
=====
Masukkan Tunai : 2000000
Kamar berhasil dipesan
Ingin memesan kamar lagi? 0/1
0
Rayz Hotel
1. No. 101 (Tidak Tersedia)
2. No. 102 (Tidak Tersedia)
3. No. 103 (Tidak Tersedia)
Pilih kamar yang tersedia: 

```

KEGIATAN 2

Buatlah sebuah diagram class bebas yang memuat sebuah objek dan jelaskan kepada asisten mengenai class diagram class tersebut.

Catatan:

- Class diagram tidak boleh mengambil dari contoh yang sudah dibuat dari modul.
- Jika ada source code yang identik, maka akan ada pengurangan nilai.

RUBRIK PENILAIAN

Aspek Penilaian	Poin
Codelab 1	10%
Codelab 2	10%
Tugas Kegiatan 1	25%
Tugas Kegiatan 2	10%
Pemahaman	45%
Total	100%

```

class weather{

    public int[] temp;

    weather(int[] argS){
        this.temp = argS;
    }

    public void tampilWeather(){
        for (int i = 0; i < this.temp.length; i++) {
            if (this.temp[i] >= 30) {
                System.out.println("Suhu : " + this.temp[i] + " : " + "Panas");
            } else if (this.temp[i] >= 25 && this.temp[i]<30) {
                System.out.println("Suhu : " + this.temp[i] + " : " + "Hangat");
            } else if (this.temp[i] < 25) {
                System.out.println("Suhu : " + this.temp[i] + " : " + "Sejuk");
            }
        }
    }

}

}

public class Main {

    public static void main(String[] args) {
        int suhu[] = {28,32,25,30,22};
        weather objWeath = new weather(suhu);
        objWeath.tampilWeather();

    }

}

```

```

public class exName {

    String[] nama;


    public exName(String[] nama){

        this.nama = nama;

    }


    public void checkPanjangNama(){

        for(int i = 0; i < nama.length; i++){

            System.out.print(nama[i]);

            int panjang = 0;

            for(int j = 0; j < nama[i].length(); j++){

                panjang += 1;

            }

            if(panjang <= 5){

                System.out.println(" Memiliki panjang 5 karakter atau kurang");

            } else {

                System.out.println(" Memiliki panjang lebih dari 5 karakter");

            }

        }

    }

}

import java.io.File;

import java.util.Random;


public class Main {

    public static void main(String[] args) throws Exception {

        String[] kumpulanNama = {"Agus", "Martin", "Rudi", "Darmono"};


        exName classCheck = new exName(kumpulanNama);

        classCheck.checkPanjangNama();

    }

}

```