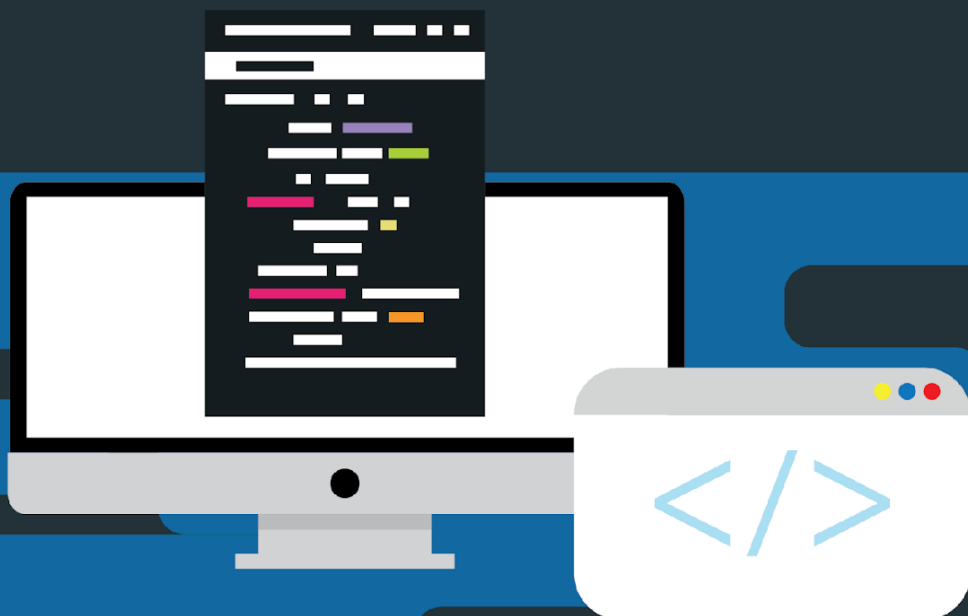


MODUL PRAKTIKUM
DESAIN DAN PEMROGRAMAN BERORIENTASI OBJEK

IMPLEMENTASI KELAS
&
ENKAPSULASI



TIM ASISTEN PEMROGRAMAN
ANGKATAN 11
ILMU KOMPUTER FPMIPA UPI

1. Pendahuluan

Pada kesempatan kali ini, kita akan mengimplementasikan penggunaan kelas pada 4 bahasa pemrograman, yaitu C++, PHP, Python, dan Java. Sebelum melakukan implementasi kelas, kita perlu mengetahui beberapa istilah yang ada dalam kelas(class) pada OOP.

Istilah	Keterangan
Class	Realisasi dari suatu objek yang dibuat, yang memiliki atribut dan fungsi
Property/Atribut	Beberapa nilai yang dimiliki oleh suatu class Misal : Class Manusia memiliki atribut yaitu nama manusia dan gender nya
Method/Fungsi	Beberapa aksi yang bisa dilakukan oleh suatu objek/class
Constructor	Metode khusus yang dijalankan otomatis ketika instansiasi objek dilakukan <ul style="list-style-type: none">• Dalam Java dan C++, nama constructor harus sama dengan nama kelasnya• Dalam PHP, nama constructor didefinisikan sebagai method bernama <code>__construct</code>• Dalam Python, nama constructor didefinisikan sebagai method bernama <code>__init</code>
Destructor	Metode khusus yang berjalan otomatis ketika suatu objek akan dihapus dari memori <ul style="list-style-type: none">• Dalam C++, destructor didefinisikan sama dengan nama kelasnya, namun ditambahkan karakter '~' di depan nama destructor nya• Diluar C++, destructor tidak perlu didefinisikan secara explicit, karena telah mendukung <i>garbage collection</i>
Inner Class	Class yang didefinisikan di dalam suatu class lain. Kasus ini biasanya terjadi bila suatu class memiliki satu kesatuan dengan kelas lainnya, seperti class Calculator dan Operator dari kalkulator itu sendiri
Enkapsulasi	Proses membungkus atribut dan fungsi dari suatu class agar tidak dapat diakses secara sembarangan oleh program lain di luar class tersebut

2. Contoh Implementasi Kelas & Enkapsulasi

Kita akan mulai implementasi pada bahasa C++ untuk membuat class bernama PersegiPanjang yang memiliki atribut panjang dan lebar, serta memiliki method/fungsi untuk membuat PersegiPanjang seperti menentukan setiap panjang dan lebar nya, serta fungsi untuk menghitung keliling dan luasnya. Berikut adalah contoh implementasinya

- PersegiPanjang.cpp

```
class PersegiPanjang
{
private:
    /*ini namanya atribut*/
    int panjang;
    int lebar;
public:
    /*ini namanya constructor*/
    PersegiPanjang(){
        this->panjang = this->lebar = 0;
    }
    /*sama ajjh, cuman ditambahin parameter*/
    PersegiPanjang(int panjang, int lebar){
        /*kodingan dibawah ini artinya panjang dan lebar dari class
        PersegiPanjang (yang ada this nya) diisi oleh panjang dan lebar
        inputan dari parameter*/
        this->panjang = panjang;
        this->lebar = lebar;
    }
    /*4 prosedur/fungsi dibawah ini adalah getter/setter*/
    void setPanjang(int panjang){
        this->panjang = panjang;
    }

    int getPanjang(){
        return this->panjang;
    }

    void setLebar(int lebar){
        this->lebar = lebar;
    }
}
```

```

        int getLebar(){
            return this->lebar;
        }

        int keliling(){
            return 2*(this->panjang + this->lebar);
        }

        int luas(){
            return this->panjang * this->lebar;
        }

        /*ini namanya destructor*/
        ~PersegiPanjang(){}
    };

```

- main.cpp

```

#include<bits/stdc++.h>
#include "PersegiPanjang.cpp"
using namespace std;

int main()
{
    //cara instansiasi objek persegi panjang (cara 1: gunakan setter dari objek)
    PersegiPanjang rectangle1;
    rectangle1.setPanjang(6);
    rectangle1.setLebar(3);

    //cara instansiasi objek persegi panjang (cara 2: gunakan constructor dari objek)
    PersegiPanjang rectangle2(10, 5);

    //coba deh pake inputan
    int n = 0, i = 0;
    cin>>n;

    //bikin array objek
    PersegiPanjang arrRectangle[n];

    for (i = 0; i < n; i++)
    {
        int p, l;
        cin>>p>>l;
        //gini caranya pake array objek
        arrRectangle[i].setPanjang(p);
        arrRectangle[i].setLebar(l);
    }
}

```

```

//get nilai keliling dan luas setiap objeknya
cout<<"Keliling Persegi Panjang 1 : "<< rectangle1.keliling() << endl;
cout<<"Luas Persegi Panjang 1 : "<< rectangle1.luas() << endl;
cout<<"Keliling Persegi Panjang 2 : "<< rectangle2.keliling() << endl;
cout<<"Luas Persegi Panjang 2 : "<< rectangle2.luas() << endl;

//kalo array caranya pake perulangan
for (i = 0; i < n; i++)
{
    cout<<"Keliling Persegi Panjang : "<< arrRectangle[i].keliling() << endl;
    cout<<"Luas Persegi Panjang : "<< arrRectangle[i].luas() << endl;
}

return 0;
}

```

Bagaimana jika objek Persegi Panjang ini diimplementasikan dalam bahasa pemrograman PHP? Kita mulai dengan membuat sebuah folder di dalam folder htdocs xampp kalian (bebas nama folder nya apa juga), kemudian buatlah 2 file berikut

- PersegiPanjang.php

```
?php

class PersegiPanjang
{
    private $panjang = 0;
    private $lebar = 0;

    public function __construct($panjang = 0, $lebar = 0) {
        $this->panjang = $panjang;
        $this->lebar = $lebar;
    }

    public function setPanjang($panjang)
    {
        $this->panjang = $panjang;
    }

    public function getPanjang()
    {
        return $this->panjang;
    }

    public function setLebar($lebar)
    {
        $this->lebar = $lebar;
    }

    public function getLebar()
    {
        return $this->lebar;
    }

    public function keliling()
    {
        return 2 * ($this->panjang + $this->lebar);
    }

    public function luas()
    {
        return $this->panjang * $this->lebar;
    }
}
```

- index.php

```
<?php
include "PersegiPanjang.php";

$rectangle1 = new PersegiPanjang();
$rectangle1->setPanjang(6);
$rectangle1->setLebar(3);

$rectangle2 = new PersegiPanjang(10, 5);

echo "Keliling Persegi Panjang 1 : ". $rectangle1->keliling() . "<br>";
echo "Luas Persegi Panjang 1 : ". $rectangle1->luas() . "<br>";
echo "Keliling Persegi Panjang 2 : ". $rectangle2->keliling() . "<br>";
echo "Luas Persegi Panjang 2 : ". $rectangle2->luas() . "<br>";
```

Note: untuk bahasa PHP, kita hanya bisa menjalankan programnya melalui browser, dengan cara ketik **localhost/nama folder yang kalian buat tadi/index.php**

Sekarang kita maju ke bahasa berikutnya, yaitu Python. Kita akan mengimplementasikan kelas PersegiPanjang juga, yang didefinisikan sebagai berikut

- PersegiPanjang.py

```
class PersegiPanjang():
    #private members
    __panjang = 0
    __lebar = 0

    def __init__(self, panjang=0, lebar=0):
        self.__panjang = panjang
        self.__lebar = lebar

    def setPanjang(self, panjang):
        self.__panjang = panjang

    def getPanjang(self):
        return self.__panjang

    def setLebar(self, lebar):
        self.__lebar = lebar

    def getLebar(self):
        return self.__lebar

    def keliling(self):
        return 2 * (self.__panjang + self.__lebar)

    def luas(self):
        return self.__panjang * self.__lebar
```


- main.py

```
from PersegiPanjang import PersegiPanjang

#pake setter
rectangle1 = PersegiPanjang()
rectangle1.setPanjang(6)
rectangle1.setLebar(3)
#pake constructor
rectangle2 = PersegiPanjang(10, 5)

n = int(input())
#kalo objeknya banyak, gini cara alokasi dan sekaligus instansiasi objeknya
arrRectangle = [PersegiPanjang() for i in range(n)]
i = 0

for i in range(n):
    #cara input banyak angka dalam satu baris di python
    p, l = map(int, input().split())
    #cara tampung nilai panjang dan lebar nya pake setter
    arrRectangle[i].setPanjang(p)
    arrRectangle[i].setLebar(l)

#ambil nilainya dengan getter
print("Keliling Persegi Panjang 1 : " + str(rectangle1.keliling()))
print("Luas Persegi Panjang 1 : " + str(rectangle1.luas()))
print("Keliling Persegi Panjang 2 : " + str(rectangle2.keliling()))
print("Luas Persegi Panjang 2 : " + str(rectangle2.luas()))

#kalo banyak, pake perulangan
i = 0
for i in range(n):
    print("Keliling Persegi Panjang : " + str(arrRectangle[i].keliling()))
    print("Luas Persegi Panjang : " + str(arrRectangle[i].luas()))
```

Yang terakhir adalah implementasi menggunakan bahasa pemrograman Java. Khusus Java, penamaan class dengan nama file **harus disesuaikan**, misalnya nama file nya Main, maka nama class nya juga harus Main. Selain itu, huruf pertama dari file harus kapital, yang artinya huruf pertama dari nama class juga harus kapital. Contohnya pada program PersegiPanjang berikut.

- PersegiPanjang.java

```
public class PersegiPanjang {  
    private int panjang;  
    private int lebar;  
  
    public PersegiPanjang(int panjang, int lebar) {  
        this.panjang = panjang;  
        this.lebar = lebar;  
    }  
  
    public PersegiPanjang() {  
    }  
  
    public int getPanjang() {  
        return panjang;  
    }  
  
    public void setPanjang(int panjang) {  
        this.panjang = panjang;  
    }  
  
    public int getLebar() {  
        return lebar;  
    }  
  
    public void setLebar(int lebar) {  
        this.lebar = lebar;  
    }  
}
```

```
    public int keliling(){  
        return 2 * (this.panjang + this.lebar);  
    }  
  
    public int luas(){  
        return this.panjang * this.lebar;  
    }  
}
```

- Main.java

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        int n = 0;

        //instansiasi kalo di java (cara 1: pake setter)
        PersegiPanjang rectangle1 = new PersegiPanjang();
        rectangle1.setPanjang(6);
        rectangle1.setLebar(3);

        //instansiasi kalo di java (cara 2: pake constructor)
        PersegiPanjang rectangle2 = new PersegiPanjang(10, 5);

        //cara bikin inputan di java
        Scanner sc = new Scanner(System.in);
        try{
            n = sc.nextInt();
        }catch(Exception e){}

        //kalo objeknya banyak, alokasikan objek sebanyak n di array
        PersegiPanjang[] arrRectangle = new PersegiPanjang[n];
    }
}
```

```
int i = 0, p = 0, l = 0;
for(i=0; i < n; i++){
    //input panjang dan lebar
    try{
        p = sc.nextInt();
    }catch(Exception e){}

    try{
        l = sc.nextInt();
    }catch(Exception e){}

    //baru instansiasi setiap objeknya disini
    arrRectangle[i] = new PersegiPanjang(p, l);
}

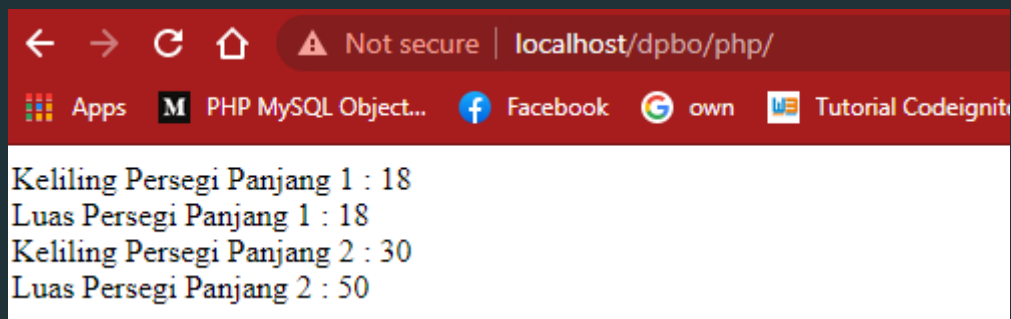
System.out.println("Keliling Persegi Panjang 1 : " + rectangle1.keliling());
System.out.println("Luas Persegi Panjang 1 : " + rectangle1.luas());
System.out.println("Keliling Persegi Panjang 2 : " + rectangle2.keliling());
System.out.println("Luas Persegi Panjang 2 : " + rectangle2.luas());

for(i=0; i < n; i++){
    System.out.println("Keliling Persegi Panjang : " + arrRectangle[i].keliling());
    System.out.println("Luas Persegi Panjang : " + arrRectangle[i].luas());
}
}
```

Ketika bahasa Java, Python dan C++ menjalankan program PersegiPanjang yang telah dibuat, maka hasilnya adalah seperti berikut

```
1
8 3
Keliling Persegi Panjang 1 : 18
Luas Persegi Panjang 1 : 18
Keliling Persegi Panjang 2 : 30
Luas Persegi Panjang 2 : 50
Keliling Persegi Panjang : 22
Luas Persegi Panjang : 24
```

Sementara, karena bahasa pemrograman PHP tidak bisa membuat program menerima inputan command line, sehingga pada program yang dibuat, tidak ada data inputan dari program karena perlu membuat tampilan form terlebih dahulu. Oleh karena itu, beginilah hasil program PersegiPanjang di bahasa PHP.



Sekilas tentang Enkapsulasi

Program OOP sangat disarankan (atau bahkan harus) menggunakan konsep enkapsulasi ini, dimana dengan tujuan agar data yang disimpan dalam kelas tidak dapat diakses sembarangan oleh program lain di luar kelas tersebut. Pada beberapa bagian program, kita lihat ada pemberian atribut yang disimpan sebagai atribut dengan akses private, dan ada juga method/fungsi yang diberi sebagai akses public. Atribut diberi akses private supaya tidak ada manipulasi data dari suatu kelas secara bebas dari pihak luar yang mengaksesnya, sehingga bila orang lain ingin mengubah data dari kelas tersebut, perlu adanya mekanisme atau proses yang dilewati agar bisa mengubah data tersebut. Mekanisme tersebut digambarkan sebagai method/fungsi dari suatu kelas tersebut, karena mereka bisa diakses secara public.

KANG PUSING! MASIH BINGUNG MAKSUDNYA GIMANA! :’ (Kutipan dari suara hati mahasiswa)

Oke, kita imajinasikan konsep enkapsulasi ini dengan kehidupan nyatanya. Contohnya yaitu membandingkan mekanisme memesan makanan di warteg dan memesan makanan di restoran. Anggap warteg dan restoran tersebut adalah suatu kelas, sementara menu makanan di warteg atau di restoran sebagai atribut dari kelas tersebut.

Ketika kita memesan makanan di warteg, mekanisme nya adalah kita bisa dengan bebas mengambil menu mana saja yang akan kita pesan, dan kita bisa membayar seluruh pesanan kita setelah selesai makan. Sementara, ketika kita memesan makanan di restoran, kita tidak bisa seperti itu, tidak ada uang artinya tidak bisa memesan, lalu kita tidak bisa mengambil menu yang kita inginkan secara langsung, harus melalui kasir dari restoran itu sendiri.

Namun, ketika memesan makanan di restoran, bila uang yang kita miliki mencukupi untuk memesan makanan, makanan yang kita pesan akan langsung sampai ke tangan kita, terjamin kualitasnya karena adanya proses sterilisasi makanan, serta pastinya pesanan sesuai dengan yang kita pesan. Hal ini akan berbahaya bagi warteg bila mana ada yang DARMAJI (Dahar (makan) lima dibayar hiji (satu)), atau karena pelanggan bisa mengambil menu mana saja secara bebas, bisa saja dari menu tersebut (amit-amit) ada menu yang sebelumnya diambil pelanggan lain, lalu tak sengaja jatuh ke tempat yang kotor, dan pelanggan malah mengembalikan menu tersebut ke piring/wadah yang tersedia di etalase. Hal tersebut pastinya akan merugikan warteg mereka.

Dengan penjelasan yang tertera di atas, jelas bahwa enkapsulasi digunakan dengan alasan untuk menjaga keamanan data dari kelas, supaya tidak mudah dimanipulasi oleh akses yang tidak berwenang.

3. Latihan

- Buat program untuk membuat daftar mahasiswa yang memiliki atribut NIM, Nama, Jenis Kelamin, Program Studi, dan Semester, menggunakan bahasa PHP!
- Buat program untuk membuat daftar tim sepakbola yang memiliki atribut nama tim, negara asal tim, tahun berdiri tim, serta pemain, menggunakan setiap bahasa (C++, PHP, Python dan Java)!

Note:

- Untuk latihan poin b, kalian bisa menambahkan atribut baru, atau membuat suatu atribut menjadi lebih detail (misal dibuat jadi inner class), namun tidak menghilangkan atribut yang sudah didefinisikan
- Buat terlebih dahulu satu repository github dengan nama "LATIHAN1DPBO2022" (Jika belum punya akun github, register dulu ya ges ya)
- Pada repository yang sudah dibuat, tambahkan file Readme.md untuk menjelaskan desain dari program latihan poin b
- Simpan screenshot hasil program di dalam folder drive
- Link submission latihan nya [disini](#)

4. Penutup

Penulis mengucapkan terimakasih yang sebesar-besarnya kepada semua elemen yang telah mendukung berjalannya sesi praktikum pada mata kuliah Desain dan Pemrograman Berorientasi Objek kali ini. Semoga apa yang kita dapatkan kali ini bisa menjadi berkah bagi kita semua.

5. Referensi

Rosa Ariani Sukamto. (2021). Pengenalan Kelas dan Implementasinya. Desain dan Pemrograman Berorientasi Objek. Bandung, Jawa Barat, Indonesia

Enkapsulasi

<https://sis.binus.ac.id/2017/09/04/encapsulation/>

Python Tutorial

<https://www.geeksforgeeks.org/python-programming-language/?ref=shm>

<https://docs.python.org/3/>

Constructor & Destructor

<https://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-constructor-dan-destructor/>

Java Tutorial

<https://www.w3schools.com/java/>

C++ Tutorial

<https://www.cplusplus.com/doc/>

PHP Tutorial

<https://www.php.net/docs.php>

<https://www.w3schools.com/php/>

Panduan penggunaan Readme.md untuk memberikan deskripsi kode program

<https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>