LAPORAN TUGAS PRAKTIKUM PROGRAM BERORIENTASI OBJEK

"Study Kasus Menghitung Luas Diarsir Menggunakan Enkapsulasi Java **' GENAP'** "

Nama : Fauzan Afif Lutfiansah

NIM : 432022611016

PRODI ; Teknik Informatika/3

Dosen : Ustadzah Hanifatus Sa'diah Widihasaniputri S.kom,M.kom

Github project :

https://github.com/apipz123456/Studi_Kasus_Menghitung_Luas_DiArsir

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS DARUSSALAM GONTOR SIMAN

1444 H / 2023 M

Daftar Isi

BAB I

Pendahuluan

1. Latar belakang

Dengan berkembangnya era digital saat ini dan semakin majunya peradaban dunia akan system informasi manusia tak bisa jauh dengan hal hal yang berbau digital/system informasi karna dalam system informasi seperti aplikasi dan lain-lain di buat untuk mempermudah pekerjaan manusia seperti penerapan ilmu Object Oriented Programing yang di rancang untuk pengonsepan model di dunia nyata dan membuat system yang mempermudah pekerjaan manusia seperti pembuatan aplikasi penghitung dan lain-lain.

Objek Oriented Programming (OOP) adalah paradigma pemrograman yang fokus pada Bahasa pemerograman yang mendukung OOP yang berfokuskan pada konsep objek, yang memungkinkan pemodelan dunia nyata dalam bentuk program computer. Salah satu konsep penting OOP adalah "Enkapsulasi", yang mengacu pada kemampuan suatu objek untuk berperilaku dengan cara yang berbeda beda tergantung pada konteks yang di inginkan oleh programmer tersebut.

Enkapsulasi adalah salah satu konsep dasar dalam pemrograman berorientasi objek (OOP) yang digunakan dalam bahasa pemrograman Java. Konsep enkapsulasi mengacu pada pembungkusan atau pengemasan data bersama dengan metode yang mengoperasikannya dalam sebuah kelas. Dalam konteks Java, enkapsulasi biasanya melibatkan penggunaan atribut privat (private fields) dan metode publik (public methods) untuk mengakses dan memanipulasi atribut tersebut.

2. Tujuan

Adapun tujuan dalam laporan pembelajarn Tugas Objek Oriented Programing dengan bertemakan Enkapsulasi sebagai berikut:

2.1. Memahami konsep Enkapsulasi.

2.2 Memahami Studi Kasus.

2.2. Iplementasi Setter Dan Getter.

BAB II

Teori Dasar

2.1 Enkapsulasi

Enkapsulasi adalah salah satu konsep dasar dalam pemrograman berorientasi objek (OOP) yang digunakan dalam bahasa pemrograman Java. Konsep enkapsulasi mengacu pada pembungkusan atau pengemasan data bersama dengan metode yang mengoperasikannya dalam sebuah kelas. Dalam konteks Java, enkapsulasi biasanya melibatkan penggunaan atribut privat (private fields) dan metode publik (public methods) untuk mengakses dan memanipulasi atribut tersebut.

2.2 Getter

Getter adalah metode dalam pemrograman berorientasi objek yang digunakan untuk mengambil atau mendapatkan nilai dari atribut (variabel) privat suatu objek. Dalam konteks Java atau bahasa pemrograman lain yang mendukung konsep enkapsulasi, getter digunakan untuk mengakses nilai atribut yang memiliki aksesibilitas private atau protected. Fungsi utama dari getter adalah untuk memberikan akses yang terkontrol dan aman ke atribut privat dalam sebuah kelas. Dengan cara ini, Anda dapat melindungi data dalam objek dari perubahan yang tidak diinginkan dan memberikan kontrol atas bagaimana data dapat diakses dan digunakan.

Pada dasarnya, getter adalah metode yang hanya mengembalikan nilai atribut dan tidak melakukan perubahan pada data tersebut. Getter biasanya memiliki tipe data yang sesuai dengan tipe data atribut yang akan diambil.

2.3 Setter

Setter adalah metode dalam pemrograman berorientasi objek yang digunakan untuk mengubah atau mengatur nilai dari atribut (variabel) suatu objek. Dalam konteks Java atau bahasa pemrograman lain yang mendukung konsep enkapsulasi, setter digunakan untuk mengatur nilai atribut yang memiliki aksesibilitas private atau protected. Fungsi utama dari setter adalah untuk memberikan cara yang terkontrol untuk mengubah data dalam sebuah objek. Dengan menggunakan setter, Anda dapat memvalidasi data yang akan diatur, menjalankan logika tambahan jika diperlukan, dan mengendalikan cara data dapat diubah.

Setter biasanya memiliki nama yang dimulai dengan kata "set" diikuti dengan nama atribut yang akan diatur, dan menerima parameter yang sesuai dengan tipe data atribut yang akan diubah.

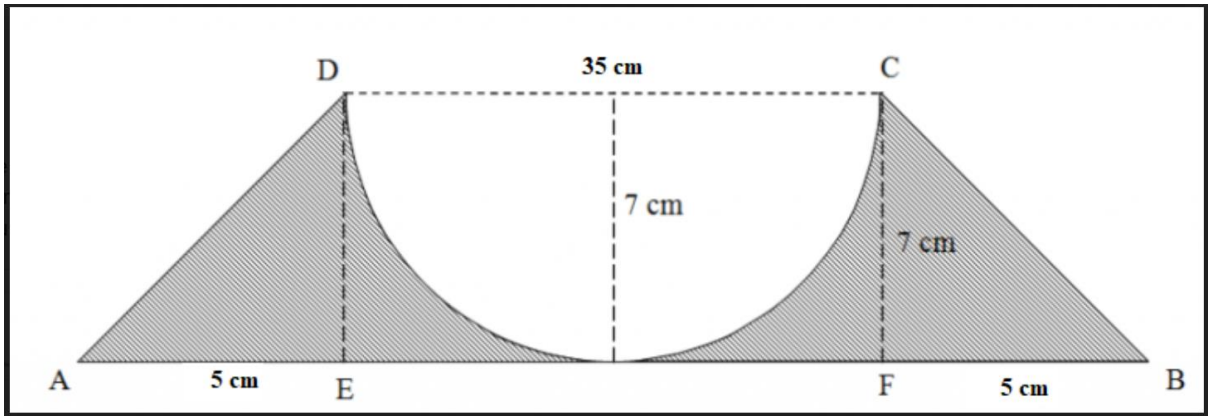
BAB III

Pembahasan

2.1 Enkapsulasi

Enkapsulasi adalah salah satu konsep dasar dalam pemrograman berorientasi objek (OOP) yang digunakan dalam bahasa pemrograman Java. Konsep enkapsulasi mengacu pada pembungkusan atau pengemasan data bersama dengan metode yang mengoperasikannya dalam sebuah kelas. Dalam konteks Java, enkapsulasi biasanya melibatkan penggunaan atribut privat (private fields) dan metode publik (public methods) untuk mengakses dan memanipulasi atribut tersebut.

2.2 Memahami Studi Kasus



1. Dalam studi kasus ini kita dapat menyimpulkan bahwa terdapat 4 komponen bangun ruang di dalamnya yaitu :

   - Segitiga ADE

   Dengan Panjang Alas 5 cm , dan Tinggi 7 cm

   - Segitiga BCF

   Dengan Panjang Alas 5 cm , adan tinggi 7 cm

   - Persegi Panjang CDEF

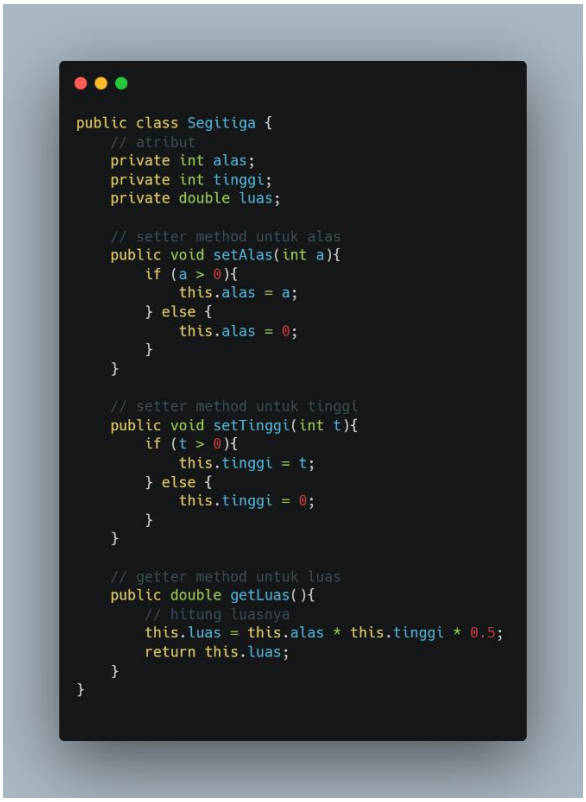
   Dengan Panjang Persegi Panjang 35 cm, dan Lebar 7 cm

   - Setengah Lingkaran

   Dengan Diameter 35 cm / jari2 = 17,5 cm

2. Dalam penyelesaia studi kasus kali ini 4 komponen bangun ruang tersebut kita implementasikan kedalam sebuah kelas masing masing untuk mempermudah compailer dalam memahami pembahasan study kasus yang kita kerjakan , Adapun kelas sebagai berikut

- Segitiga ADE



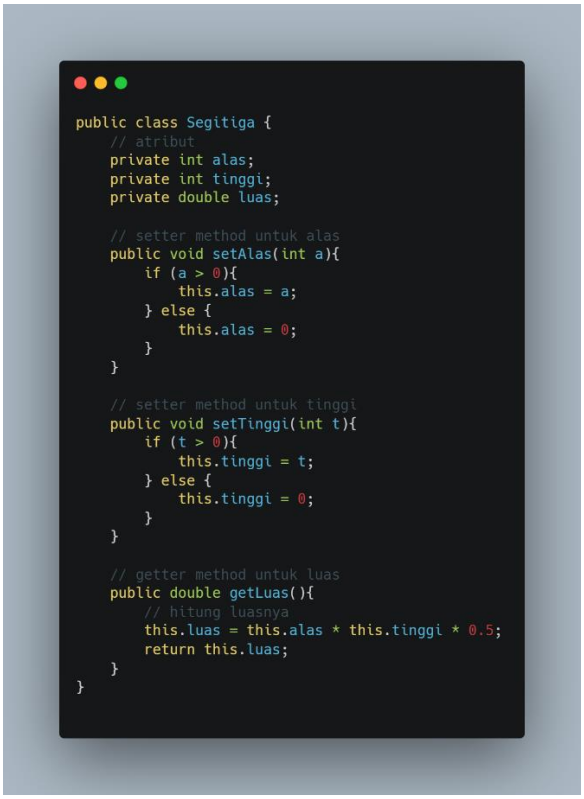
```
public class Segitiga {
    // atribut
    private int alas;
    private int tinggi;
    private double luas;

    // setter method untuk alas
    public void setAlas(int a){
        if (a > 0){
            this.alas = a;
        } else {
            this.alas = 0;
        }
    }

    // setter method untuk tinggi
    public void setTinggi(int t){
        if (t > 0){
            this.tinggi = t;
        } else {
            this.tinggi = 0;
        }
    }

    // getter method untuk luas
    public double getLuas(){
        // hitung luasnya
        this.luas = this.alas * this.tinggi * 0.5;
        return this.luas;
    }
}
```

- Class Segitiga BCF



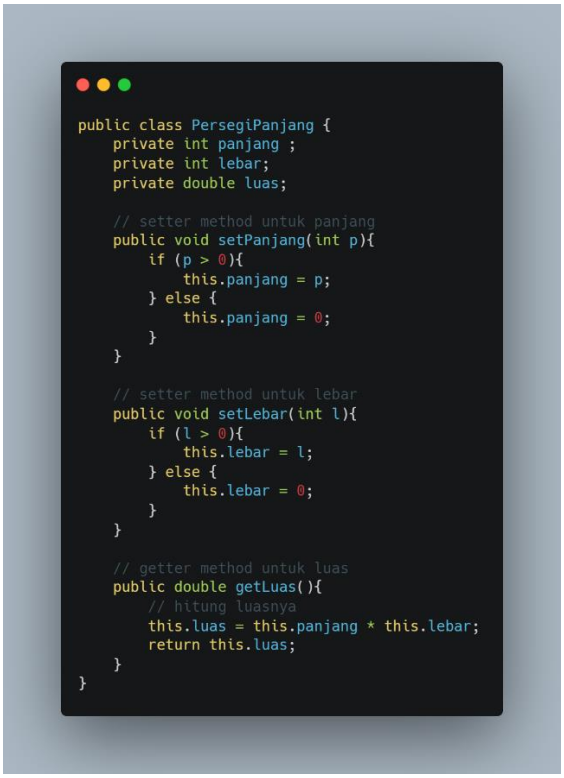
```
public class Segitiga {
    // atribut
    private int alas;
    private int tinggi;
    private double luas;

    // setter method untuk alas
    public void setAlas(int a){
        if (a > 0){
            this.alas = a;
        } else {
            this.alas = 0;
        }
    }

    // setter method untuk tinggi
    public void setTinggi(int t){
        if (t > 0){
            this.tinggi = t;
        } else {
            this.tinggi = 0;
        }
    }

    // getter method untuk luas
    public double getLuas(){
        // hitung luasnya
        this.luas = this.alas * this.tinggi * 0.5;
        return this.luas;
    }
}
```

- Class Persegi Panjang CDEF



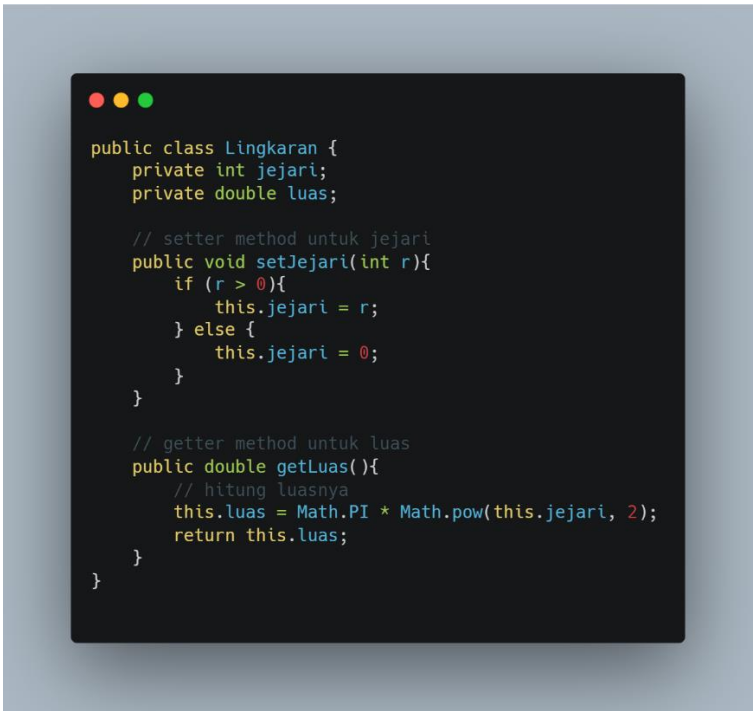
```java
public class PersegiPanjang {
    private int panjang ;
    private int lebar;
    private double luas;

    // setter method untuk panjang
    public void setPanjang(int p){
        if (p > 0){
            this.panjang = p;
        } else {
            this.panjang = 0;
        }
    }

    // setter method untuk lebar
    public void setLebar(int l){
        if (l > 0){
            this.lebar = l;
        } else {
            this.lebar = 0;
        }
    }

    // getter method untuk luas
    public double getLuas(){
        // hitung luasnya
        this.luas = this.panjang * this.lebar;
        return this.luas;
    }
}
```

- Class Setengah Lingkaran



```java
public class Lingkaran {
    private int jejari;
    private double luas;

    // setter method untuk jejari
    public void setJejari(int r){
        if (r > 0){
            this.jejari = r;
        } else {
            this.jejari = 0;
        }
    }

    // getter method untuk luas
    public double getLuas(){
        // hitung luasnya
        this.luas = Math.PI * Math.pow(this.jejari, 2);
        return this.luas;
    }
}
```

- Main Class

  Pada class Segitiga, method setAlas() dan setTinggi() merupakan *setter method* yang nantinya digunakan untuk men-set atau meng-*assign* nilai alas dan tinggi dari segitiganya. Adapun getLuas() adalah *getter method* digunakan untuk mendapatkan nilai luas setelah proses perhitungan.
  - Dengan Panjang Alas segitiga ADE 5 cm , dan Tinggi 7 cm
  - Dengan Panjang Alas BCF 5 cm , adan tinggi 7 cm
  - Dengan Panjang Persegi Panjang CDEF 35 cm, dan Lebar 7 cm
  - Dengan Stengah Lingkaran Diameter 35 cm / jari2 = 17,5 cm

```
public class MENGHITUNG_LUASARSIR {

        public static void main(String[] args) {
//segitiga ADE
        Segitiga st1 = new Segitiga();
        st1.setAlas(5);
        st1.setTinggi(7);
        double luasADE = st1.getLuas();

//segitiga CBF
        Segitiga st2 = new Segitiga();
        st2.setAlas(5);
        st2.setTinggi(7);
        double luasCBF = st2.getLuas();

//persegipanjang CDEF
        PersegiPanjang pp1 = new PersegiPanjang();
        pp1.setPanjang(35);
        pp1.setLebar((int) 17.5);
        double luasCDEF = pp1.getLuas();

//setengah lingkaran
        Lingkaran l1 = new Lingkaran();
        l1.setJejari(7);
        double SetengahLingkaran = 0.5 * l1.getLuas();

//hitung luas daerah diarsir
        double luasArsir = luasADE + luasCBF + luasCDEF - SetengahLingkaran;
        System.out.println("Luas Segitiga CBF: " + luasCBF + " cm2");
        System.out.println("Luas Segitiga ADE : " + luasADE + " cm2");
        System.out.println("Luas Persegi Panjang CDEF: " + luasCDEF + " cm2");
        System.out.println("Luas Setengah Lingkaran : " + SetengahLingkaran + " cm2");
        System.out.println("================================================");
        System.out.println("Luas daerah diarsir: " + luasArsir + " cm2");
    }
    }
```

3. Hasil Runing

```
Output - MENGHITUNG_LUAS-ARSIR (run)
    run:
    Luas Segitiga CBF: 17.5 cm2
    Luas Segitiga ADE : 17.5 cm2
    Luas Persegi Panjang CDEF: 595.0 cm2
    Luas Setengah Lingkaran : 76.96902001294993 cm2
    ================================================
    Luas daerah diarsir: 553.03097998705 cm2
    BUILD SUCCESSFUL (total time: 0 seconds)
```

BAB IV

Penutup

2.1 Kesimpulan

Dalam praktikum ini, kami berhasil mengimplementasikan konsep enkapsulasi dengan menggunakan setter dan getter dalam bahasa pemrograman Jawa untuk menghitung luas diarsis. Enkapsulasi adalah salah satu prinsip dalam pemrograman berorientasi objek yang memungkinkan kita untuk melindungi data dari akses langsung dan memastikan bahwa data tersebut hanya dapat diakses dan dimanipulasi melalui metode yang telah ditentukan.

2.2 Saran

Dan pastinya saya selaku penulis mengakui masih banyak kesalahan dan kekurangan dalam segi penulisan makalah laporan tugas OOP(Object Oriented Programing) Polymorphism kali ini maka dari itu saya selaku penulis berharap keritik dan Saranya yang bersifat membangun saya selaku penulis dan mungkin pembaca-pembacanya karna manusia tak tuput dari salah dan lupa, perlu kita ketahui keilmuan itu sangatlah luas sekali seperti mana contohnya dalam laporan tugas saya kali ini yang membahas tentang Polymorphim dan terdapat beberapa percabangan di dalamnya, maka dari itu kami siap menerima kritik dan saran yang bersifat membangun dalam proses pambelajaran kami , cukup sekian dan trimakasih.

Wassalamualaikum warahmatullahi wabarakatuhu