

LAPORAN TUGAS PRAKTIKUM PROGRAM BERORIENTASI OBJEK

“Enkapsulasi Setter Getter menggunakan bahasa pemrograman Java”



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/Enkapsulasi_Setter_Getter

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS DARUSSALAM GONTOR SIMAN
1444 H / 2023 M**

Daftar Isi

BAB I : Pendahuluan

1.1 Latar Belakang.....	1
1.2 Tujuan.....	1

BAB II : Teori Dasar

2.1 Enkapsulasi.....	2
2.2 Setter.....	3
2.3 Getter.....	3

BAB III : Pembahasan

2.1 Deklarasi Enkapsulasi.....	4
2.2 Implementasi Getter dan Setter.....	5

BAB IV : Penutup

2.1 Kesimpulan.....	12
2.2 Saran.....	12

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 pemrograman 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 Ebkapsulasi sebagai berikut:

2.1. Memahami konsep Enkapsulasi.

2.2. Iplementasi Enkapsulasi 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.

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.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package pesertaujian;

/**
 *
 * @author Apipz
 */
public class PesertaUjian {
    private String nomorPeserta;
    private String nama;
    private String prodi;

    //objek
    public PesertaUjian(String nomorPeserta, String nama, String prodi) {
        this.nomorPeserta = nomorPeserta;
        this.nama = nama;
        this.prodi = prodi;
    }

    //Getter untuk nomorPeserta
    public String getNomorPeserta() {
        return nomorPeserta;
    }

    //setter untuk mengubah nomorpeserta
    public void setNomorPeserta(String nomorPeserta) {
        this.nomorPeserta = nomorPeserta;
    }

    // Getter untuk nama
    public String getName() {
        return nama;
    }

    // Setter untuk nama
    public void setName(String nama) {
        this.nama = nama;
    }

    // Getter untuk prodi
    public String getProdi() {
        return prodi;
    }

    //setter
    public void setProdi(String prodi) {
        this.prodi = prodi;
    }

    public static void main(String[] args) {
        // Membuat objek peserta ujian
        PesertaUjian pesertal = new PesertaUjian("432022611016", "Fuazan Afif Lutfiansah", "Teknik
        Informatika");

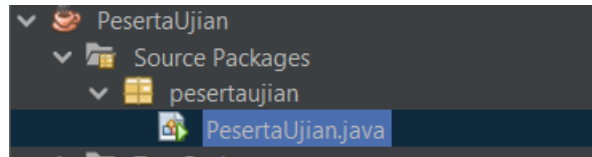
        // Menggunakan getter untuk mengakses informasi peserta ujian
        System.out.println("Nomor Peserta: " + pesertal.getNomorPeserta());
        System.out.println("Nama Peserta: " + pesertal.getName());
        System.out.println("Program Studi: " + pesertal.getProdi());

        // Menggunakan setter untuk mengubah informasi peserta ujian
        pesertal.setName("Natasya Willonteng");
        pesertal.setProdi("Teknik Informatika");

        System.out.println("\nData Peserta (setelah perubahan):");
        System.out.println("Nomor Peserta: " + pesertal.getNomorPeserta());
        System.out.println("Nama Peserta: " + pesertal.getName());
        System.out.println("Program Studi: " + pesertal.getProdi());
    }
}
```

2.2 Implementasi Enkapsulasi dalam Java

Kita membuat project Peserta Ujian dengan package dan kelas PesertaUjian menggunakan Bahasa pemrograman java.



A. Deklarasi Atribut Private (private String nama;, private String nomorUjian;,,

Private String prodi;), Ini adalah deklarasi kelas **PesertaUjian** yang akan digunakan untuk mewakili data PesertaUjian dan Deklarasi Atribut Private (private String nama;, private String nomorUjian;, private String prodi;):

```
public class PesertaUjian {  
    private String nomorPeserta;  
    private String nama;  
    private String prodi;  
  
    //objek  
    public PesertaUjian(String nomorPeserta, String nama, String prodi)  
    {  
        this.nomorPeserta = nomorPeserta;  
        this.nama = nama;  
        this.prodi = prodi;  
    }  
}
```

B. Konstruktor (public PesertaUjian(String nama, String nomorUjian, String prodi) { ... })

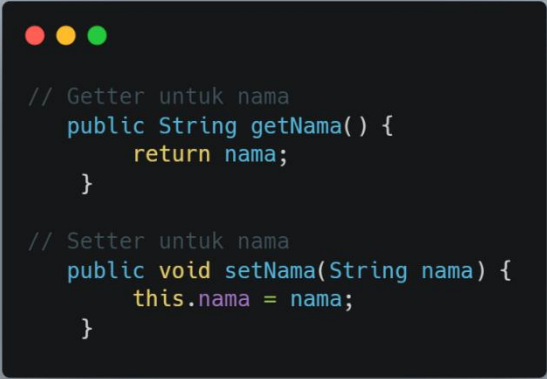
Ini adalah konstruktor kelas **PesertaUjian** yang digunakan untuk membuat objek **PesertaUjian**. Ini menerima tiga parameter: **nama**, **nomorUjian**, dan **prodi**, dan menginisialisasi atribut-atribut kelas dengan nilai-nilai yang diberikan.



```
public PesertaUjian(String nomorPeserta, String nama, String prodi)
{
    this.nomorPeserta = nomorPeserta;
    this.nama = nama;
    this.prodi = prodi;
}
```

C. Getter dan Setter untuk Nama (getNama(), setNama(String nama)):

Ini adalah metode getter dan setter untuk atribut **nama**. Getter (**getNama()**) digunakan untuk mengambil nilai atribut **nama**, sementara setter (**setNama(String nama)**) digunakan untuk mengubah nilai atribut **nama**.



```
// Getter untuk nama
public String getNama() {
    return nama;
}

// Setter untuk nama
public void setNama(String nama) {
    this.nama = nama;
}
```

D. Getter dan Setter untuk Program Studi (getProdi(), setProdi(String prodi)):

Ini adalah metode getter dan setter untuk atribut **prodi**. Getter (**getProdi()**) digunakan untuk mengambil nilai atribut **prodi**, sementara setter (**setProdi(String prodi)**) digunakan untuk mengubah nilai atribut **prodi**.

```

// Getter untuk prodi
public String getProdi() {
    return prodi;
}

//setter
public void setProdi(String prodi)
{
    this.prodi = prodi;
}

```

E. Main Java.

Fungsi main adalah fungsi khusus dalam bahasa pemrograman Java yang merupakan titik masuk utama ke program Java. Fungsi ini harus ada di setiap program Java karena bertanggung jawab untuk menjalankan program tersebut. Fungsi main memiliki sintaks yang sama untuk semua program Java dan harus didefinisikan sebagai berikut:

```

public static void main(String[] args) {
    // Membuat objek peserta ujian
    PesertaUjian peserta1 = new PesertaUjian("432022611016", "Fuazan Afif Lutfiansah", "Teknik
Informatika");

    // Menggunakan getter untuk mengakses informasi peserta ujian
    System.out.println("Nomor Peserta: " + peserta1.getNomorPeserta());
    System.out.println("Nama Peserta: " + peserta1.getNama());
    System.out.println("Program Studi: " + peserta1.getProdi());

    // Menggunakan setter untuk mengubah informasi peserta ujian
    peserta1.setNama("Natasya Willonteng");
    peserta1.setProdi("Teknik Informatika");

    System.out.println("\nData Peserta (setelah perubahan):");
    System.out.println("Nomor Peserta: " + peserta1.getNomorPeserta());
    System.out.println("Nama Peserta: " + peserta1.getNama());
    System.out.println("Program Studi: " + peserta1.getProdi());
}

```


BAB IV

Penutup

2.1 Kesimpulan

Dalam tugas ini, kami berhasil memahami dan mengimplementasikan konsep enkapsulasi dalam bahasa pemrograman Java. 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. Kami membuat kelas dengan atribut-atribut privat yang mewakili data yang perlu dienkapsulasi. Kemudian, kami menggunakan metode setter untuk mengatur nilai atribut-atribut tersebut dan metode getter untuk mengambil nilai mereka. Dengan cara ini, kami dapat memastikan bahwa akses ke data terkontrol dan terlindungi, sehingga meningkatkan keamanan dan keandalan program.

Selain itu, kami juga memahami manfaat dari enkapsulasi dalam hal meminimalkan risiko kesalahan dalam kode, karena data hanya dapat dimanipulasi melalui metode-metode yang telah didefinisikan, dan kita dapat menerapkan validasi atau logika khusus dalam metode setter jika diperlukan.

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 kritik dan Saranya yang bersifat membangun saya selaku penulis dan mungkin pembaca-pembacanya karna manusia tak luput 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 pembelajaran kami , cukup sekian dan trimakasih.

Wassalamualaikum warahmatullahi wabarakatuhu