

## Laporan Praktikum

Nama : Ahmad Billal

NIM : 2410511116

Kelas : D

### A. Poin Bahasan

#### 1. Latihan1

```
class Tiket { 2 usages
    private String jenisTiket; 2 usages
    private double hargaTiket; 2 usages
    private int jumlahKursi; 4 usages

    public Tiket(String jenisTiket, double hargaTiket, int jumlahKursi){ 1 usage
        this.jenisTiket = jenisTiket;
        this.hargaTiket = hargaTiket;
        this.jumlahKursi = jumlahKursi;
    }

    >     public String getJenisTiket() { return jenisTiket; }

    >     public double getHargaTiket() { return hargaTiket; }

    >     public int getJumlahKursi() { return jumlahKursi; }

    public boolean pesanTiket (int jumlah){ 1 usage
        if(jumlah > 0 && jumlah <= jumlahKursi){
            jumlahKursi -= jumlah;
            return true;
        }
        return false;
    }
}

public class MainTiket {
    public static void main(String[] args){
        Tiket tiket = new Tiket( jenisTiket: "Kereta Api", hargaTiket: 125000, jumlahKursi: 162);
        boolean berhasil = tiket.pesanTiket( jumlah: 3);
        System.out.println("Pemesanan Berhasil: " + berhasil);
        System.out.println("Kursi yang Tersisa: " + tiket.getJumlahKursi());
    }
}
```

Enkapsulasi disini digunakan untuk membatasi hak akses user terhadap class untuk dapat mengakses objek yang terdapat pada Tiket dan dikembalikan dengan sebuah method untuk membungkus objek pada class. Pada Program ini terdapat pesanTiket yang dimana harus di cek terlebih dahulu jumlah kursi yang masih tersedia, ika sudah melebih

batas jumlah kursi, maka akan mengembalikan nilai false dan tidak dapat memesan tiket Kembali.

Hasil :

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Pemesanan Berhasil: true  
Kursi yang Tersisa: 159  
  
Process finished with exit code 0
```

## 2. Latihan2

```
class Produk { 2 usages  
    private String namaProduk; 2 usages  
    private double harga; 3 usages  
    private int stok; 3 usages  
    // Constructor  
    public Produk(String namaProduk, double harga, int stok){ 1 usage  
        this.namaProduk = namaProduk;  
        this.harga = harga;  
        this.stok = stok;  
    }  
    // Getter  
>     public String getNamaProduk() { return namaProduk; }  
  
>     public double getHarga() { return harga; }  
  
>     public int getStok() { return stok; }  
  
    // Setter untuk mengubah harga  
    public void setHarga(double hargaBaru){ 1 usage  
        if(hargaBaru > 0){  
            this.harga = hargaBaru;  
        }  
    }  
    // Menambah stok  
    public void tambahStok(int jumlah){ 1 usage  
        if (jumlah > 0){  
            this.stok += jumlah;  
        }  
    }  
}
```

```
import java.text.NumberFormat;
import java.util.Locale;

public class MainProduk {
    public static void main(String[] args){
        Produk produk1 = new Produk( namaProduk: "Iphone 14", harga: 12500000, stok: 25);

        // Format rupiah (Indonesia)
        NumberFormat nf = NumberFormat.getInstance(new Locale( language: "id", country: "ID"));

        System.out.println("Nama Produk: " + produk1.getNamaProduk());
        System.out.println("Harga: " + nf.format(produk1.getHarga()));
        System.out.println("Stok: " + produk1.getStok());

        produk1.tambahStok( jumlah: 10);
        produk1.setHarga(12000000);
        System.out.println("Setelah perubahan - Harga: " + nf.format(produk1.getHarga()));
        System.out.println("Setelah perubahan - Stok: " + produk1.getStok());
    }
}
```

Pada Enkapsulasi ini terdapat perbedaan dimana menggunakan implementasi setter, setter digunakan dalam bentuk sebuah method yang dimana membuat suatu perubahan pada objek tertentu namun tidak dengan merubah nilai variabel secara langsung. Akan tetapi, dengan menggunakan method yang dimana merubah objek secara langsung sehingga nilai variabel pada class tetap terjaga. Setter pada program ini digunakan untuk merubah harga dan menambah jumlah stok pada produk

Hasil :

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\
Nama Produk: Iphone 14
Harga: 12.500.000
Stok: 25
Setelah perubahan - Harga: 12.000.000
Setelah perubahan - Stok: 35

Process finished with exit code 0
```

### 3. Latihan3

```
public class AkunBank { 2 usages
    private double saldo; 5 usages

    // constructor
    >     public AkunBank(double saldoAwal) { this.saldo = saldoAwal; }

    // getter
    >     public double getSaldo() { return saldo; }
    // Metode untuk menambah saldo
    public void deposit (double jumlah){ 1 usage
        if(jumlah > 0){
            saldo += jumlah;
        }
    }

    // Metode untuk menarik saldo
    public void withdraw(double jumlah){ 1 usage
        if(jumlah > 0 && jumlah <= saldo){
            saldo -= jumlah;
        }
    }
}

import java.text.NumberFormat;
import java.util.Locale;

public class MainAkunBank {
    public static void main (String[] args){
        AkunBank akun1 = new AkunBank( saldoAwal: 5000000);
        // Format rupiah (Indonesia)
        NumberFormat nf = NumberFormat.getInstance(new Locale( language: "id", country: "ID"));

        System.out.println("Saldo awal: " + nf.format(akun1.getSaldo()));

        akun1.deposit( jumlah: 500000);
        System.out.println("Saldo Setelah deposit: " + nf.format(akun1.getSaldo()));

        akun1.withdraw( jumlah: 200000);
        System.out.println("Saldo setelah withdraw: " + nf.format(akun1.getSaldo()));
    }
}
```

Enkapsulasi pada program ini digunakan untuk memunculkan nilai saldo awal dan merubah saldo Ketika melakukan deposit dan Ketika melakukan penarikan dengan menggunakan implementasi Setter untuk mengatur isi dari variabel tanpa mengakses variabel tersebut, namun dengan menggunakan method yang langsung memanipulasi objek.

Hasil :

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Pro
Saldo awal: 5.000.000
Saldo Setelah deposit: 5.500.000
Saldo setelah withdraw: 5.300.000

Process finished with exit code 0
```

#### 4. Latihan4

```
class Siswa { 2 usages
    private String nama; 3 usages
    private int usia; 3 usages
    private double nilai; 3 usages
    // Constructor
    public Siswa (String nama, int usia, double nilai){ 1 usage
        this.nama = nama;
        this.usia = usia;
        this.nilai = nilai;
    }
    // Getter dan Setter
    >     public String getNama() { return nama; }

    >     public void setNama (String nama) { this.nama = nama; }

    >     public int getUsia() { return usia; }

    public void setusia (int usia){ no usages
        if (usia > 0){
            this.usia = usia;
        }
    }
    >     public double getNilai() { return nilai; }

    public void setNilai(double nilai){ 1 usage
        if(nilai >= 0 && nilai <= 100){
            this.nilai = nilai;
        }
    }
}

public class Mahasiswa {
    public static void main(String[] args){
        Siswa siswa = new Siswa( nama: "Billal", usia: 19, nilai: 92.8);

        //Mengakses data siswa melalui getter
        System.out.println("Nama: " + siswa.getNama());
        System.out.println("Usia: " + siswa.getUsia());
        System.out.println("Nilai: " + siswa.getNilai());

        // Mengubah nilai siswa melalui setter
        siswa.setNilai(98.2);
        System.out.println("Nilai setelah diubah: " + siswa.getNilai());
    }
}
```

Enkapsulasi disini digunakan untuk memunculkan data mahasiswa dan melakukan perubahan pada nilai dengan menggunakan setter. Jika nilai lebih dari sama dengan 0 dan kurang dari sama dengan 100, maka nilai dapat dirubah, Namun, jika tidak maka nilai tidak akan dirubah.

Hasil :

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program
Nama: Billal
Usia: 19
Nilai: 92.8
Nilai setelah diubah: 98.2

Process finished with exit code 0
```

*B. Penugasan*

Buatlah program dengan menerapkan konsep enkapsulasi dengan 2 buah class yaitu Restoran dan Makanan dengan detail sebagai berikut :

a. Buat nama variable dan jenis tipe data berikut :

-menu bersifat private

-harga bersifat private

-spesial bersifat private

b. Ketika di running maka gambaran hasilnya adalah sebagai berikut :

**Menu Makanan : Ayam Goreng**

**Harga Makanan : Rp. 170000**

**Menu Spesial : true**

```

class Makanan { 2 usages
    private String menu; 2 usages
    private double harga; 3 usages
    private boolean spesial; 3 usages

    public Makanan(String menu, double harga, boolean spesial){ 1 usage
        this.menu = menu;
        this.harga = harga;
        this.spesial = spesial;
    }

    public String getMenu(){ 2 usages
        return menu;
    }

    public double getHarga(){ 2 usages
        return harga;
    }

    public boolean isSpecial(){ 2 usages
        return spesial;
    }

    public void setHarga(double harga){ 1 usage
        if(harga >= 0){
            this.harga = harga;
        }
    }

    public void setSpesial(boolean spesial) { 1 usage
        this.spesial = spesial;
    }
}

```

```

import java.text.NumberFormat;
import java.util.Locale;
import java.util.Scanner;

> public class Restoran {
>     public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Input data makanan
        System.out.print("Masukkan nama Pemesan: ");
        String nama = input.nextLine();

        System.out.print("Masukkan nama menu makanan yang dipesan: ");
        String menu = input.nextLine();

        System.out.print("Masukkan harga: ");
        double harga = input.nextDouble();

        System.out.print("Apakah menu spesial? (true/false): ");
        boolean spesial = input.nextBoolean();
        input.nextLine(); // Menghapus ENter yang tersisa

        // buat objek makanan
        Makanan makanan1 = new Makanan(menu, harga, spesial);
        // Format rupiah (Indonesia)
        NumberFormat nf = NumberFormat.getInstance(new Locale("id", "ID"));

```

```

// tampilkan hasil
System.out.println("\n==== Data Makanan " + nama + " ===");
System.out.println("Menu Makanan : " + makanan1.getMenu());
System.out.println("Harga Makanan : Rp. " + nf.format(makanan1.getHarga()));
System.out.println("Menu Spesial : " + makanan1.isSpecial());
System.out.print("\nApakah ingin ada perubahan (Iya / Tidak)? ");
String pilihan = input.nextLine();

if (pilihan.equalsIgnoreCase("Iya")) {
    System.out.print("Masukkan harga baru: ");
    harga = input.nextDouble();

    System.out.print("Apakah menu spesial? (true/false): ");
    spesial = input.nextBoolean();

    // Update Objek Makanan
    makanan1.setHarga(harga);
    makanan1.setSpesial(spesial);

    // tampilkan hasil perubahan
    System.out.println("\n==== Data Makanan Setelah Perubahan ===");
    System.out.println("Menu Makanan : " + makanan1.getMenu());
    System.out.println("Harga Makanan : Rp. " + nf.format(makanan1.getHarga()));
    System.out.println("Menu Spesial : " + makanan1.isSpecial());
}

input.close();
}
}

```

Pada Program ini, digunakan enkapsulasi untuk meruunculkan data makanan, harga, dan special atau tidaknya dengan berupa inputan dinamis dari user. Menggunakan implementasi getter untuk menampilkan data dan setter untuk mengubah data dengan mengubah harga dan ini special atau tidak. Pada setter harga, dilakukan pengecekan terlebih dahulu agar harga tidak diinput minus dan data dimunculkan Kembali sebagai data terbaru setelah dilakukan perubahan.

### Hasil :

```
"C:\Program Files\Java\jdk-24\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\lib\idea_rt.jar=53144:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\bin" -Dfile.encoding=UTF-8 -classpath "C:\Users\ahmadbillal\IdeaProjects\Kelas 1\src\main\java\com\example\Kelas1\Kelas1Application.class" com.example.Kelas1.Kelas1Application
Masukkan nama Pemesan: Ahmad Billal
Masukkan nama menu makanan yang dipesan: Nasi Goreng
Masukkan harga: 17000
Apakah menu spesial? (true/false): true

==== Data Makanan Ahmad Billal ====
Menu Makanan : Nasi Goreng
Harga Makanan : Rp. 17.000
Menu Spesial : true

Apakah ingin ada perubahan (Iya / Tidak)? Iya
Masukkan harga baru: 15000
Apakah menu spesial? (true/false): false

==== Data Makanan Setelah Perubahan ====
Menu Makanan : Nasi Goreng
Harga Makanan : Rp. 15.000
Menu Spesial : false

Process finished with exit code 0
```