

Nama : Hamka Arifani
NIM : 2410817110006
Mata Kuliah : Pemrograman II
Tugas : Pembuatan CLI

Untuk membuat program suatu menu menggunakan abstraction, interface, composition, dan tidak boleh menggunakan library diluar base dari bahasa Java. Disertai dengan spesifikasi menu harus bisa menerima input berupa pilihan pengguna, serta Main tidak boleh menjadi god class. Maka dalam pembuatan menu dengan tema toko es krim bernama YOYOMO, saya membuat total 6 class dan 1 interface sebagaimana berikut:

HavePrice.java (Interface)

<pre>package TugasCLIMenu; public interface HavePrice { int getPrice(); }</pre>

Size.java

<pre>package TugasCLIMenu; public class Size { private String nameSize; private int extraPrice; public Size(String nameSize, int extraPrice) { this.nameSize = nameSize; this.extraPrice = extraPrice; } public String getNameSize() { return nameSize; } public int getExtraPrice() { return extraPrice; } }</pre>

IceCream.java (Abstract Class)

<pre>package TugasCLIMenu; public abstract class IceCream{ private String flavour; protected Size size; public IceCream(String flavour, Size size) {</pre>

```
        this.flavour = flavour;
        this.size = size;
    }

    public String getFlavour() {
        return flavour;
    }

    public Size getSize() {
        return size;
    }

    public abstract String getType();
}
```

IceCreamCone.java

```
package TugasCLIMenu;

public class IceCreamCone extends IceCream implements HavePrice{
    public IceCreamCone(String flavour) {
        super(flavour, new Size("-", 0));
    }

    @Override
    public String getType() {
        return "Cone";
    }

    @Override
    public int getPrice() {
        return 13000;
    }
}
```

IceCreamCup.java

```
package TugasCLIMenu;

public class IceCreamCup extends IceCream implements HavePrice {
    public IceCreamCup(String flavour, Size size) {
        super(flavour, size);
    }

    @Override
    public String getType() {
        return "Cup";
    }

    @Override
    public int getPrice() {
        int base = 15000;
    }
}
```

```
        return base+size.getExtraPrice();
    }
}
```

YoyomoMenu.java

```
package TugasCLIMenu;
import java.util.Scanner;

public class YoyomoMenu {
    Scanner scan=new Scanner(System.in);

    public void start() {
        int choice;
        do {
            System.out.println("\n=== YOYOMO ICE CREAM ===");
            System.out.println("Pilih tipe es krim mu: ");
            System.out.println("1. Es Krim Cup");
            System.out.println("2. Es Krim Cone");
            System.out.println("0. Keluar");
            System.out.print("Pilihan Kamu: ");
            choice = scan.nextInt();

            switch (choice) {
                case 1 -> cupMenu();
                case 2 -> coneMenu();
                case 0 -> System.out.println("Terima kasih sudah
berkunjung ke YOYOMO!");
                default -> showInvalid();
            }
        } while (choice != 0);
    }

    private void cupMenu() {
        String flavour = getFlavourChoice();
        Size size = getSizeChoice();
        IceCreamCup cup = new IceCreamCup(flavour, size);

        System.out.println("\n=== PESANAN ANDA ===");
        System.out.println("Tipe      : " + cup.getType());
        System.out.println("Rasa      : " + cup.getFlavour());
        System.out.println("Ukuran    : " +
cup.getSize().getNameSize());
        System.out.println("Harga      : Rp" + cup.getPrice());

        System.out.println("\nTekan ENTER untuk kembali ke
menu...");
        scan.nextLine();
    }

    private void coneMenu() {
        String flavour = getFlavourChoice();
        IceCreamCone cone = new IceCreamCone(flavour);
    }
}
```

```

        System.out.println("\n=== PESANAN ANDA ===");
        System.out.println("Tipe      : " + cone.getType());
        System.out.println("Rasa      : " + cone.getFlavour());
        System.out.println("Harga      : Rp" + cone.getPrice());

        System.out.println("\nTekan ENTER untuk kembali ke
menu...");
        scan.nextLine();
    }

    private String getFlavourChoice() {
        int choice;
        do {
            System.out.println("\nPilih rasa favoritmu:");
            System.out.println("1. Vanilla");
            System.out.println("2. Cokelat");
            System.out.println("3. Stroberi");
            System.out.print("Pilihan Kamu: ");
            choice = scan.nextInt();
            scan.nextLine(); // flush newline

            switch (choice) {
                case 1 -> { return "Vanilla"; }
                case 2 -> { return "Cokelat"; }
                case 3 -> { return "Stroberi"; }
                default -> { showInvalid(); }
            }
        } while (true);
    }

    private Size getSizeChoice() {
        System.out.println("\nPilih ukuran:");
        System.out.println("1. Small");
        System.out.println("2. Medium (+Rp5000)");
        System.out.println("3. Large (+Rp10000)");
        System.out.print("Pilihan: ");
        int choice = scan.nextInt();
        scan.nextLine();

        return switch (choice) {
            case 1 -> new Size("Small", 0);
            case 2 -> new Size("Medium", 5000);
            case 3 -> new Size("Large", 10000);
            default -> new Size("Small", 0);
        };
    }

    private void showInvalid() {
        System.out.println("\nUps! Pilihan kamu tidak dikenali
oleh sistem.");
        System.out.println("Tekan ENTER untuk kembali...");
        scan.nextLine();
    }
}

```

YoyomoMain.java

```
package TugasCLIMenu;

public class YoyomoMain {
    public static void main(String[] args) {
        YoyomoMenu yoyomoMenu = new YoyomoMenu();
        yoyomoMenu.start();
    }
}
```

Alasan & Insight

Dalam penerapan abstract class disini saya memilih IceCream sebagai abstract, dan method getType() juga dijadikan abstract agar setiap jenis es krim mengembalikan nilai tipenya sendiri, sehingga pada logika menu hanya tinggal memanggil method getType() dari class IceCreamCone maupun IceCreamCup untuk menampilkan jenis es krim.

Pada program ini Size dibuat menjadi class tersendiri agar dapat diimplementasi menggunakan composition dikarenakan ini akan memudahkan jika misalnya terjadi penambahan menu kedepannya seperti milkshake yang membuat program harus diperbarui, karena milkshake itu juga akan memiliki hubungan mempunyai pilihan ukuran size dan perhitungan harganya bisa menjadi lebih mudah karena sudah ada Size yang menyimpan extraPrice untuk menghitung harga total. Alasan yang sama juga menjadi dasar kenapa HavePrice dijadikan interface meskipun sebenarnya bisa saja setiap class itu mengimplementasi sendiri getPrice apabila dibuat menjadi abstract.

Guna menghindari class Main menjadi god class maka tugas yang berat seperti penampilan menu saya fokuskan pada bagian menu dengan logikanya adalah akan terus menampilkan menu selama input bukan 0, kemudian baru menggunakan switch untuk mengatur apa yang akan dilakukan program untuk merespon pilihan pengguna. Dikarenakan setiap pilihan akan mempunyai menu baru yang akan ditampilkan, seperti jika telah memilih tipe cone maka lanjut menampilkan menu untuk pengguna memilih rasa, sementara jika tipe cup yang dipilih terlebih dahulu menampilkan menu untuk memilih ukuran, lalu dilanjutkan memilih rasa, maka agar simpel saya memutuskan membuat method yakni coneMenu() dan cupMenu() yang dibuat untuk memanggil menu kemudian menyimpan hasil yang dikembalikan menu, serta membuat deskripsi dari variasi es krim yang sudah dipilih pengguna. Sementara tampilan menu dan switchnya itu sendiri saya buat terpisah ke getFlavourChoice() dan getSizeChoice() untuk menyederhanakan kode dari coneMenu dan cupMenu().

Dari pembuatan program CLI menu ini saya belajar untuk membuat Main yang dumb class hanya membuat objek lalu memanggil method dan program bisa berjalan. Dalam proses pembuatan saya juga sempat mendapat pengetahuan baru kalau abstract class bisa mengimplentasikan interface sehingga setiap subclassnya otomatis mengimplementasi interface, walaupun saya tidak membuat programnya seperti itu karea menurut saya itu malah membuat konsep interface yang digunakan ketika jenisnya berbeda itu jadi kurang terasa.

Tautan Github

<https://github.com/HamkaArifani/Pemrograman-II>