LAPORAN PRAKTIKUM TEKPRO MINGGU – 3

Penyusun:

Zaidan Zulkaisi Setiaji

1A – D4 Teknik Informatika

DAFTAR ISI

KASUS 1

Pertanyaan:

1. Amati desain setiap class, Apakah desain class tersebut sudah memenuhi konsep OOP yang benar? Jika tidak, coba anda perbaiki dengan mengacu pada Design Hint di Buku Chapter 4.10. Setiap perubahan yang dibuat harus dibubuhi penjelasan serta argumentasiyang jelas.
2. Ada kebutuhan untuk mengembangkan aplikasi tersebut, dengan menambah fitur pemesanan dan mengurangi setiap stok yang ada. Apakah dengan desain program yang ada dapat dikembangkan? Jika Sulit kemukakan alasannya dan bandingkan dengan desain class hasil modifikasi anda.

Jawaban:

1. Setelah mengamati desain setiap class dan membandingkannya dengan mengacu pada Design Hint di Buku Chapter 4.10. Terdapat beberapa kesalahan yang perlu diperbaikin agar desain class dapat memenuhi konsep OOP yang benar. Pertama adalah tidak adanya penggunaan enkapsulasi data (*always keep data private*), khususnya Pada kelas Restaurant, variabel nama_makanan, harga_makanan, stok, dan id dideklarasikan sebagai public. Ini melanggar prinsip enkapsulasi karena data seharusnya diprivate untuk mencegah akses langsung dari luar kelas. Kedua adalah tanggung jawab kelas yang terlalu banyak. Kelas Restaurant bertanggung jawab untuk menambah menu, menampilkan menu, dan memeriksa stok. Ini bisa dipecah menjadi kelas yang lebih kecil untuk memisahkan tanggung jawab. Dan terakhir adalah kelas Restaurant tidak immutable karena data dapat diubah setelah objek dibuat. Ini tidak selalu buruk, tetapi perlu dipertimbangkan jika ada kebutuhan untuk membuat kelas immutable.
2. Desain program yang ada saat ini akan sulit untuk diimplementasikan fitur pemesanan dikarenakan alasan – alasan berikut:

- Data tersebar di beberapa array.
- Tidak ada encapsulation.
- Tidak ada kelas untuk mewakili makanan.
- Tidak ada mekanisme pemesanan.

Hal yang perlu diubah:

- Menggunakan ArrayList dan kelas Menu untuk modularitas.
- Menerapkan encapsulation dengan membuat atribut private dan menyediakan getter/setter.

Desain program setelah dimodifikasi :

- Restaurant.java

```
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Scanner;

public class RestaurantFix {
    private MenuManager menuManager;

    public RestaurantFix(){
        menuManager = new MenuManager();
    }

    public void tambahMenuMakanan(String nama, double harga, int
stok) {
        menuManager.tambahMenuMakanan(nama, harga, stok);
```

```
    }

    public void tampilMenuMakanan() {
        menuManager.tampilMenuMakanan();
    }

    public void pesanMakanan(){
        menuManager.PesanMakanan();
    }

}

class Menu{
    private final String nama_makanan;
    private final double harga_makanan;
    private int stok;

    public Menu(String nama, double harga, int stok){
        this.nama_makanan = nama;
        this.harga_makanan = harga;
        this.stok = stok;
    }

    public String getNama_makanan(){
        return nama_makanan;
    }

    public double getHarga_makanan() {
        return harga_makanan;
    }

    public int getStok() {
        return stok;
    }

    public void setStok(int stok) {
        this.stok = stok;
    }
}

class MenuManager{
    private ArrayList<Menu> menu;
    public MenuManager(){
        menu = new ArrayList<>();
    }

    public void tambahMenuMakanan(String nama,double harga, int
stok){
        menu.add(new Menu(nama, harga, stok));
    }

    public void tampilMenuMakanan(){
        int i = 1;
        for(Menu makanan : menu){
            if(!isOutOfStock(makanan)){
                System.out.println(i+ ". " +
makanan.getNama_makanan() + "[" + makanan.getStok() + "]" + "\tRp. "
+ makanan.getHarga_makanan());
                i++;
            }
        }
```

```
    }

    private boolean isOutOfStock(Menu makanan) {
        return makanan.getStok() == 0;
    }

    public void PesanMakanan() {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            tampilMenuMakanan();
            System.out.println("Pilih nomor makanan (Ketik 0 jika
ingin keluar): ");
            int pilihan = scanner.nextInt();

            if (pilihan == 0) {
                System.out.println("Terima kasih telah memesan!");
                break;
            }
            if (pilihan < 1 || pilihan > menu.size()) {
                System.out.println("Pilihan tidak valid. Silahkan
coba lagi.");
                continue;
            }

            Menu makanan = menu.get(pilihan - 1);
            if (isOutOfStock(makanan)) {
                System.out.println("Maaf, stok " +
makanan.getNama_makanan() + "habis.");
                continue;
            }

            System.out.println("Masukkan jumlah yang ingin dipesan:
");
            int jumlah = scanner.nextInt();

            if (jumlah <= 0) {
                System.out.println("Jumlah tidak valid. Silahkan coba
lagi.");
                continue;
            }
            if (jumlah > makanan.getStok()) {
                System.out.println("Maaf, jumlah stok " +
makanan.getNama_makanan() + " tidak mencukupi.");
                continue;
            }

            makanan.setStok(makanan.getStok() - jumlah);
            System.out.println("Anda telah memesan " + jumlah + " " +
makanan.getNama_makanan() + ".");
            System.out.println("Stok tersisa: " + makanan.getStok());
        }
    }
}
```

- RestaurantMain.java

```
import java.util.Scanner;

public class RestaurantMainFix {
    public static void main(String[] args){
        RestaurantFix menu = new RestaurantFix();
        menu.tambahMenuMakanan("Bala-bala", 1_000, 20);
```

```
        menu.tambahMenuMakanan("Gehu", 1_000, 20);
        menu.tambahMenuMakanan("Tahu", 1_000, 0);
        menu.tambahMenuMakanan("Molen", 1_000, 20);
        menu.tambahMenuMakanan("Bubur Memek", 15_000, 20);

        Scanner scanner = new Scanner(System.in);
        boolean loop = true;
        while (loop == true){
            System.out.println("\n1. Tampilkan Menu");
            System.out.println("2. Pesan Makanan");
            System.out.println("3. Keluar");
            System.out.println("Pilih opsi: ");
            int opsi = scanner.nextInt();

            switch (opsi){
                case 1:
                    menu.tampilMenuMakanan();
                    break;

                case 2:
                    menu.pesanMakanan();
                    break;

                case 3:
                    loop = false;
                    System.out.println("Terima kasih telah
menggunakan layanan kami!");
                    break;
            }


        }
    }
}
```

KASUS 2

Pertanyaan:

1. Identifikasi Kelas, attribute, method
   - Film
     Atribut: (String judul_film, String Genre, double Durasi)
     Metode: Getter, Constructor
   - Ticket
     Atribut: (Film judulFilm, String Jadwal, double Harga)
     Metode: Getter, Constructor
   - Pelanggan
     Atribut: (String nama, String email)
     Metode: Getter, Constructor
   - Pemesanan
     Atribut: (Pelanggan pelanggan, Ticket tiket, int Jumlah)
     Metode: Getter, Constructor

2. Implementasi program
   - Film

```
class Film {
    private final String judul_film;
    private final String Genre;
    private final double Durasi;

    public Film(String judulFilm, String genre, double durasi) {
        this.judul_film = judulFilm;
        this.Genre = genre;
        this.Durasi = durasi;
    }

    public String getJudul_film() {
        return judul_film;
    }

    public String getGenre() {
        return Genre;
    }

    public double getDurasi() {
        return Durasi;
    }
}
```

   - Ticket

```
class Ticket {
    private final Film judulFilm;
    private final String Jadwal;
    private final double Harga;

    public Ticket(Film film, String jadwal, double harga) {
        this.judulFilm = film;
        this.Harga = harga;
        this.Jadwal = jadwal;
    }

    public String getFilm() {
        return judulFilm.getJudul_film();
```

```
    }

    public double getHarga() {
        return Harga;
    }

    public String getJadwal() {
        return Jadwal;
    }
}
```

- Pelanggan

```
class Pelanggan {
    private final String nama;
    private final String email;

    public Pelanggan(String nama, String email) {
        this.email = email;
        this.nama = nama;
    }

    public String getEmail() {
        return email;
    }

    public String getNama() {
        return nama;
    }
}
```

- Pemesanan

```
class Pemesanan {
    private final Ticket tiket;
    private final Pelanggan pelanggan;
    private final int jumlah;

    public Pemesanan(Ticket tiket, Pelanggan pelanggan, int jumlah) {
        this.tiket = tiket;
        this.pelanggan = pelanggan;
        this.jumlah = jumlah;
    }

    public Pelanggan getPelanggan() {
        return pelanggan;
    }

    public Ticket getTiket() {
        return tiket;
    }

    public int getJumlah() {
        return jumlah;
    }

    public double TotalHarga() {
        return jumlah * tiket.getHarga();
    }

}
```