Nama : Siti Fauziah Wulandari

NIM : 2309106038

Kelas : A2

Praktikum PBO

POSTTEST 6

---

1. **Interface Baru (Pembayaran.java)**
2. **Penggunaan static (variabel dan method di PembayaranService.java dan Main.java)**
3. **Error handling (try-catch block di PelangganService.java dan Main.java)**

Pembayaran. Java

```java
package services;

import models.Pembayaran;
import java.util.Arrays;
import java.util.List;

public class PembayaranService implements Pembayaran {
    private static final List<String> METODE_PEMBAYARAN_TERSEDIA = Arrays.asList(
        "Ambil di Apotek",
        "Transfer",
        "QRIS",
        "Kartu Kredit"
    );

    private static double BIAYA_ADMIN = 2000.0;

    public static List<String> getMetodePembayaranTersedia() {
        return METODE_PEMBAYARAN_TERSEDIA;
    }

    public static void setBiayaAdmin(double biayaBaru) {
        if (biayaBaru >= 0) {
            BIAYA_ADMIN = biayaBaru;
        }
    }

    public static double getBiayaAdmin() {
        return BIAYA_ADMIN;
    }

    @Override
    public boolean cekPembayaran(String metodePembayaran) {
        return METODE_PEMBAYARAN_TERSEDIA.contains(metodePembayaran);
    }

    @Override
    public void prosesPembayaran(double totalHarga, String metodePembayaran) {
        if (!cekPembayaran(metodePembayaran)) {
            throw new IllegalArgumentException("Metode pembayaran tidak tersedia: " + metodePembayaran);
        }

        double biayaFinal = totalHarga;

        if (!metodePembayaran.equals("Ambil di Apotek")) {
            biayaFinal += BIAYA_ADMIN;
        }

        System.out.println("Memproses pembayaran sebesar Rp " + biayaFinal + " dengan metode " + metodePembayaran);
        System.out.println("Pembayaran berhasil!");
    }
}
```

Main.java

```java
import services.AdminService;
import services.PelangganService;
import services.PembayaranService;
import models.Obat;
import models.Transaksi;
import models.ResepDokter;
import models.Admin;
import models.Pelanggan;
import models.User;

import java.util.ArrayList;
import java.util.Scanner;

public class Main {
    // Static variable untuk version
    public static final String APP_VERSION = "1.0.0";
    public static final String APP_NAME = "APOTEK SEHAT SENTOSA";

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Obat> daftarObat = new ArrayList<>();
        ArrayList<Transaksi> daftarTransaksi = new ArrayList<>();
        ArrayList<ResepDokter> daftarResep = new ArrayList<>();
        ArrayList<User> daftarUser = new ArrayList<>();

        Admin admin = new Admin(1, "AdminDefault", "081234567890", "Farmasi");
        daftarUser.add(admin);

        PembayaranService.setBiayaAdmin(2500.0);

        tampilkanInfoAplikasi();

        AdminService adminService = new AdminService(daftarObat, daftarTransaksi, daftarResep, scanner);
        PelangganService pelangganService = new PelangganService(daftarObat, daftarTransaksi, daftarResep, scanner);

        while (true) {
            try {
                tampilkanMenuUtama();
                System.out.print("Pilih: ");
                int pilihan = inputAngka(scanner);

                switch (pilihan) {
                    case 1:
                        adminService.menuAdmin();
                        break;
                    case 2:
                        pelangganService.menuPelanggan();
                        break;
                    case 3:
                        System.out.println("Terima kasih! Program selesai.");
                        scanner.close();
                        return;
                    default:
                        System.out.println("Pilihan tidak valid.");
                }
            } catch (Exception e) {
                System.out.println("Terjadi kesalahan: " + e.getMessage());
            }
        }
    }

    public static void tampilkanMenuUtama() {
        System.out.println("\n=== SELAMAT DATANG DI " + APP_NAME + " ===");
        System.out.println("1. Admin");
        System.out.println("2. Pelanggan");
        System.out.println("3. Exit");
    }

    public static void tampilkanInfoAplikasi() {
        System.out.println("=====================================");
        System.out.println("      " + APP_NAME);
        System.out.println("      Version: " + APP_VERSION);
        System.out.println("=====================================");
    }

    private static int inputAngka(Scanner scanner) {
        while (true) {
            try {
                return Integer.parseInt(scanner.nextLine().trim());
            } catch (NumberFormatException e) {
                System.out.print("Masukkan harus berupa angka! Coba lagi: ");
            }
        }
    }
}
```

PelangganService.java

```java
package services;

import models.Obat;
import models.Pelanggan;
import models.ResepDokter;
import models.Transaksi;

import java.time.LocalDate;
import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.Scanner;

public class PelangganService {
    // Encapsulation
    private ArrayList<Obat> daftarObat = new ArrayList<>();
    private ArrayList<Transaksi> daftarTransaksi = new ArrayList<>();
    private ArrayList<ResepDokter> daftarResep = new ArrayList<>();
    private Scanner scanner = new Scanner(System.in);
    private int idPelanggan = 1;
    private int idResep = 1;

    private PembayaranService pembayaranService = new PembayaranService();

    public PelangganService(ArrayList<Obat> daftarObat, ArrayList<Transaksi> daftarTransaksi, ArrayList<ResepDokter> daftarResep, Scanner scanner) {
        this.daftarObat = daftarObat;
        this.daftarTransaksi = daftarTransaksi;
        this.daftarResep = daftarResep;
        this.scanner = scanner;
    }

    public void menuPelanggan() {
        while (true) {
            try {
                System.out.println("\n=== MENU PELANGGAN ===");
                System.out.println("1. Lihat Obat");
                System.out.println("2. Beli Obat");
                System.out.println("3. Lihat Riwayat Transaksi");
                System.out.println("4. Logout");
                System.out.print("Pilih: ");
                int pilihan = Integer.parseInt(scanner.nextLine().trim());

                switch (pilihan) {
                    case 1: lihatObat(); break;
                    case 2: beliObat(); break;
                    case 3: lihatRiwayatTransaksi(); break;
                    case 4: return;
                    default: System.out.println("Pilihan tidak valid.");
                }
            } catch (NumberFormatException e) {
                System.out.println("Error: Masukkan harus berupa angka! Silakan coba lagi.");
            } catch (Exception e) {
                System.out.println("Error: " + e.getMessage());
            }
        }
    }

    private void lihatObat() {
        if (daftarObat.isEmpty()) {
            System.out.println("Tidak ada obat yang tersedia.");
            return;
        }

        System.out.println("\n=== DAFTAR OBAT ===");
        for (Obat obat : daftarObat) {
            obat.tampilkanObat();
        }
    }

    private int inputUsia() {
        while (true) {
            try {
                System.out.print("Masukkan Usia: ");
                return Integer.parseInt(scanner.nextLine());
            } catch (NumberFormatException e) {
                System.out.println("Usia harus berupa angka. Silakan coba lagi.");
            }
        }
    }
```

Lanjutan PelangganService.java

```java
// Method final
public final void tambahPelanggan(Pelanggan pelanggan) {
    System.out.println("Pelanggan baru berhasil ditambahkan!");
    pelanggan.tampilkanInfo();
}

// protected
protected void beliObat() {
    try {
        System.out.println("=== BELI OBAT ===");
        lihatObat();

        if (daftarObat.isEmpty()) {
            System.out.println("Tidak ada obat yang tersedia untuk dibeli.");
            return;
        }

        System.out.print("Masukkan ID Obat yang ingin dibeli: ");
        int idObat = Integer.parseInt(scanner.nextLine().trim());

        Obat obatDibeli = null;
        for (Obat obat : daftarObat) {
            if (obat.getIdObat() == idObat) {
                obatDibeli = obat;
                break;
            }
        }

        if (obatDibeli == null) {
            throw new IllegalArgumentException("Obat dengan ID " + idObat + " tidak ditemukan!");
        }

        System.out.print("Masukkan Nama Anda: ");
        String namaPelanggan = scanner.nextLine();
        if (namaPelanggan.trim().isEmpty()) {
            throw new IllegalArgumentException("Nama tidak boleh kosong!");
        }

        int usia = inputUsia();
        if (usia <= 0 || usia > 120) {
            throw new IllegalArgumentException("Usia tidak valid! Harus antara 1-120 tahun.");
        }

        System.out.print("Masukkan Alamat: ");
        String alamat = scanner.nextLine();
        if (alamat.trim().isEmpty()) {
            throw new IllegalArgumentException("Alamat tidak boleh kosong!");
        }

        System.out.print("Masukkan Nomor Telepon: ");
        String noTelp = scanner.nextLine();
        if (noTelp.trim().isEmpty()) {
            throw new IllegalArgumentException("Nomor telepon tidak boleh kosong!");
        }

        System.out.print("Jumlah yang ingin dibeli: ");
        int jumlah = Integer.parseInt(scanner.nextLine().trim());
        if (jumlah <= 0) {
            throw new IllegalArgumentException("Jumlah pembelian harus lebih dari 0!");
        }

        if (jumlah > obatDibeli.getStok()) {
            throw new IllegalArgumentException("Stok tidak mencukupi. Stok tersedia: " + obatDibeli.getStok());
        }

        boolean butuhResep = obatDibeli.isResepDokter();
        if (butuhResep) {
            System.out.println("\n=== Obat ini memerlukan resep dokter. ===");
            System.out.print("Nama Dokter: ");
            String namaDokter = scanner.nextLine();
            if (namaDokter.trim().isEmpty()) {
                throw new IllegalArgumentException("Nama dokter tidak boleh kosong!");
            }

            System.out.print("Nama Klinik: ");
            String namaKlinik = scanner.nextLine();
            if (namaKlinik.trim().isEmpty()) {
                throw new IllegalArgumentException("Nama klinik tidak boleh kosong!");
            }

            LocalDate tanggalResep = LocalDate.now();

            ResepDokter resep = new ResepDokter(idResep++, idPelanggan, idObat, namaDokter, namaKlinik, tanggalResep, false);
            daftarResep.add(resep);
            System.out.println("Resep berhasil diunggah, menunggu verifikasi Admin.");
        } else {
            System.out.println("\nObat ini **tidak** memerlukan resep dokter. Langsung ke pembayaran.");
        }

        System.out.println("\n=== PILIH METODE PEMBAYARAN ===");
        int idx = 1;
        for (String metode : PembayaranService.getMetodePembayaranTersedia()) {
            System.out.println(idx + ". " + metode);
            idx++;
        }
```

Lanjutan PelangganService.java

```
    System.out.print("Pilih: ");
            int metodePilihan = Integer.parseInt(scanner.nextLine().trim());

            if (metodePilihan < 1 || metodePilihan > PembayaranService.getMetodePembayaranTersedia().size()) {
                throw new IllegalArgumentException("Metode pembayaran tidak valid!");
            }

            String metodePembayaran = PembayaranService.getMetodePembayaranTersedia().get(metodePilihan - 1);
            double totalHarga = jumlah * obatDibeli.getHarga();

            if (!metodePembayaran.equals("Ambil di Apotek")) {
                System.out.println("Biaya admin: Rp " + PembayaranService.getBiayaAdmin());
                totalHarga += PembayaranService.getBiayaAdmin();
            }

            String statusTransaksi = butuhResep ? "Menunggu Verifikasi Resep" : "Selesai";
            String statusPengambilan = metodePembayaran.equals("Ambil di Apotek") ? "Belum Diambil" : "Selesai";

            Pelanggan pelanggan = new Pelanggan(idPelanggan, namaPelanggan, usia, alamat, noTelp);

            System.out.println("\n=== DETAIL PELANGGAN ===");
            pelanggan.tampilkanInfo();

            System.out.println("Identitas: " + pelanggan.getIdentitas());

            // Proses pembayaran menggunakan interface
            try {
                pembayaranService.prosesPembayaran(totalHarga, metodePembayaran);
            } catch (IllegalArgumentException e) {
                System.out.println("Error pembayaran: " + e.getMessage());
                return;
            }

            Transaksi transaksi = new Transaksi(
                daftarTransaksi.size() + 1, idPelanggan, idObat, jumlah, totalHarga, LocalDate.now(),
                statusTransaksi, metodePembayaran, statusPengambilan, alamat, namaPelanggan, usia, noTelp
            );

            daftarTransaksi.add(transaksi);
            obatDibeli.setStok(obatDibeli.getStok() - jumlah);
            System.out.println("\nTransaksi berhasil dibuat!");
            idPelanggan++;

        } catch (NumberFormatException e) {
            System.out.println("Error: Masukkan angka yang valid!");
        } catch (IllegalArgumentException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Terjadi kesalahan: " + e.getMessage());
        }
    }

    // Getter
    public ArrayList<Transaksi> getDaftarTransaksi() {
        return daftarTransaksi;
    }

    // Setter
    public void setDaftarTransaksi(ArrayList<Transaksi> daftarTransaksi) {
        this.daftarTransaksi = daftarTransaksi;
    }

    private void lihatRiwayatTransaksi() {
        if (daftarTransaksi.isEmpty()) {
            System.out.println("Belum ada riwayat transaksi.");
            return;
        }

        System.out.println("\n=== RIWAYAT TRANSAKSI ===");
        for (Transaksi transaksi : daftarTransaksi) {
            transaksi.tampilkanTransaksi();
        }
    }
}
```