Nama : Davina Putri Ananta

NIM : 2309106002

Kelas: IT A’23

Postttest: 5

SS Pelanggan.java

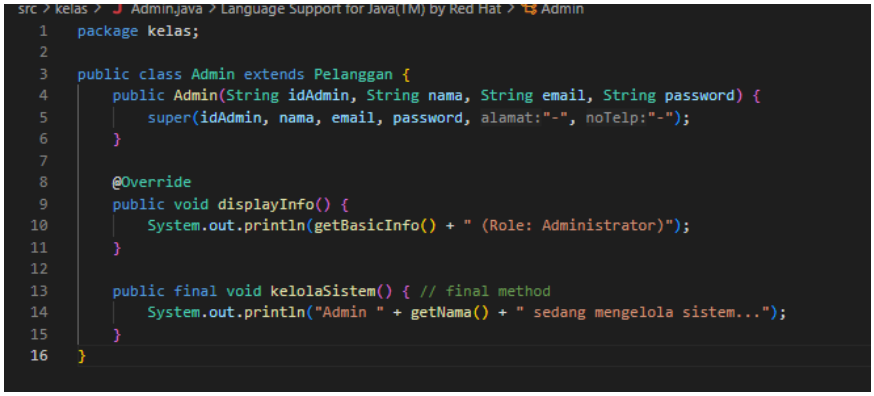
```java
package kelas;

public abstract class Pelanggan {
    private final String idPelanggan;
    private String nama;
    private String email;
    private String password;
    private String alamat;
    private String noTelp;

    public Pelanggan(String idPelanggan, String nama, String email,
                     String password, String alamat, String noTelp) {
        this.idPelanggan = idPelanggan;
        this.nama = nama;
        this.email = email;
        this.password = password;
        this.alamat = alamat;
        this.noTelp = noTelp;
    }

    public abstract void displayInfo();

    public final String getBasicInfo() {
        return "ID: " + idPelanggan + ", Nama: " + nama;
    }

    public final String getIdPelanggan() {
        return idPelanggan;
    }
    public String getNama() {
        return nama;
    }

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

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
```

```java

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getAlamat() {
        return alamat;
    }

    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }

    public String getNoTelp() {
        return noTelp;
    }

    public void setNoTelp(String noTelp) {
        this.noTelp = noTelp;
    }

    public final void resetPassword(String newPassword) {
        if (newPassword.length() >= 8) {
            this.password = newPassword;
            System.out.println(x:"Password berhasil direset");
        } else {
            System.out.println(x:"Password harus minimal 8 karakter");
        }
    }
}
```

Parfum.java

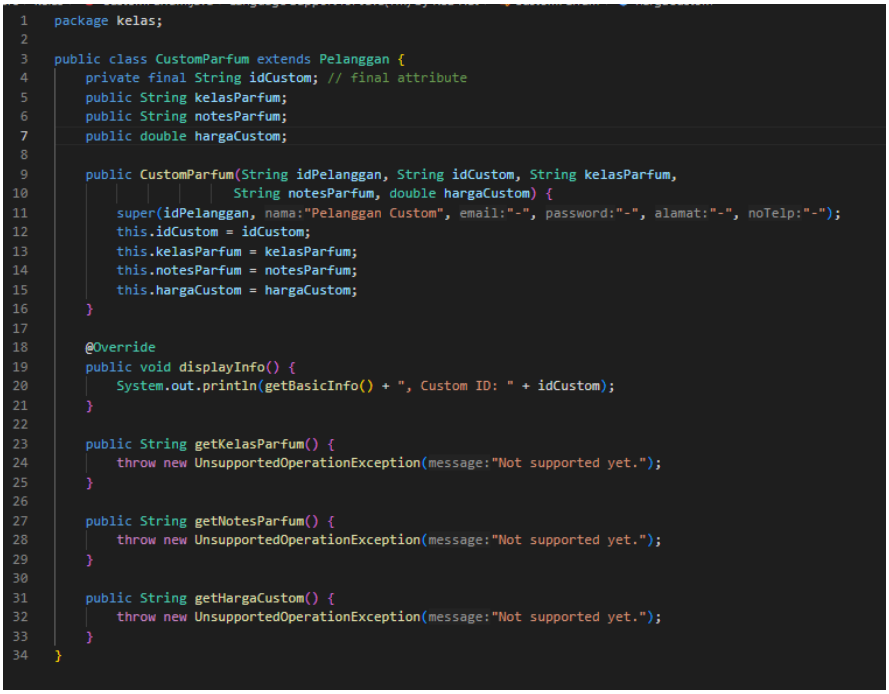
```java
package kelas;

public final class Parfum {
    private final String idParfum;
    private String namaParfum;
    private final double hargaParfum;
    private int stok;

    public Parfum(String idParfum, String namaParfum, double hargaParfum, int stok) {
        this.idParfum = idParfum;
        this.namaParfum = namaParfum;
        this.hargaParfum = hargaParfum;
        this.stok = stok;
    }

    public final double getHargaWithTax() {
        return hargaParfum * 1.11;
    }

    public String getIdParfum() {
        return idParfum;
    }

    public String getNamaParfum() {
        return namaParfum;
    }

    public void setNamaParfum(String namaParfum) {
        this.namaParfum = namaParfum;
    }

    public double getHargaParfum() {
        return hargaParfum;
    }

    public int getStok() {
        return stok;
    }

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

    public final boolean isTersedia() {
        return stok > 0;
    }

    @Override
    public String toString() {
        return "Parfum[id=" + idParfum +
               ", nama=" + namaParfum +
               ", harga=" + hargaParfum +
               ", stok=" + stok + "]";
    }
}
```

Admin.java



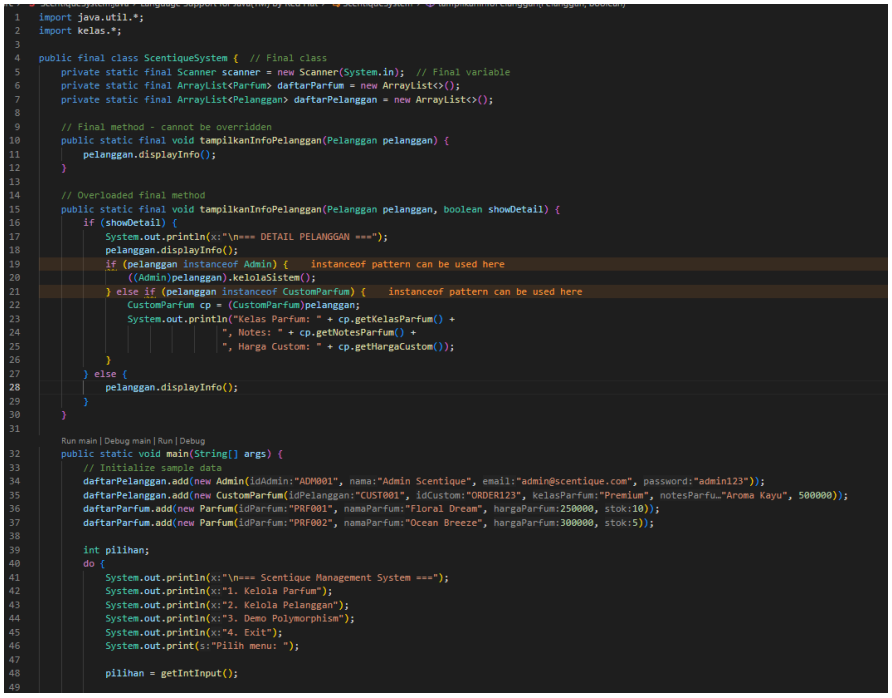
```java
package kelas;

public class Admin extends Pelanggan {
    public Admin(String idAdmin, String nama, String email, String password) {
        super(idAdmin, nama, email, password, alamat:"-", noTelp:"-");
    }

    @Override
    public void displayInfo() {
        System.out.println(getBasicInfo() + " (Role: Administrator)");
    }

    public final void kelolaSistem() { // final method
        System.out.println("Admin " + getNama() + " sedang mengelola sistem...");
    }
}
```

CustomParfum.java

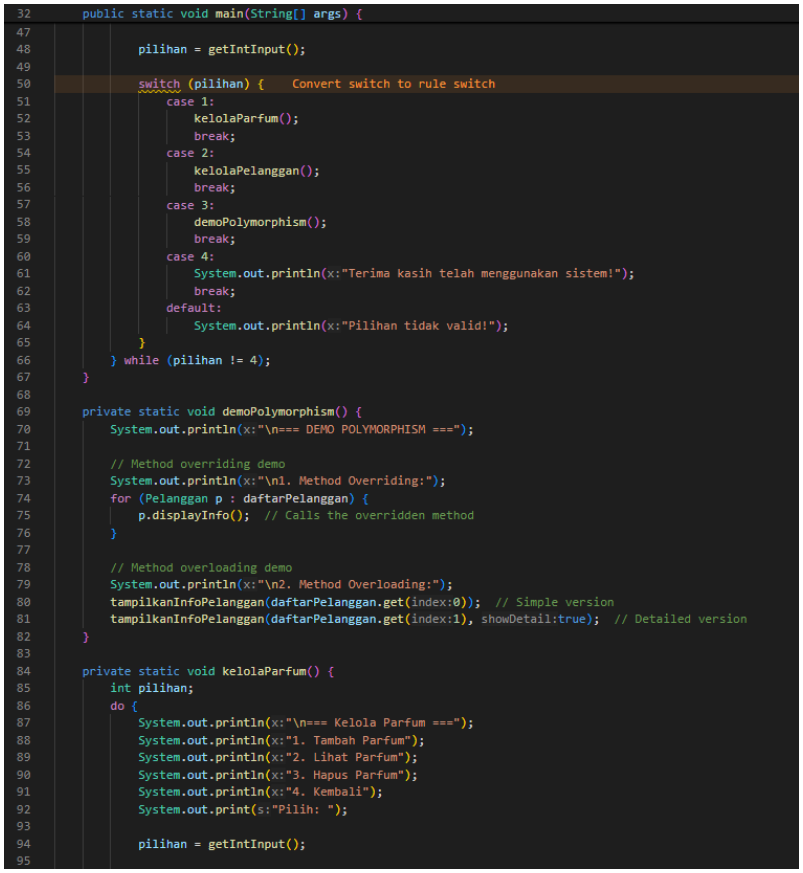


```java
package kelas;

public class CustomParfum extends Pelanggan {
    private final String idCustom; // final attribute
    public String kelasParfum;
    public String notesParfum;
    public double hargaCustom;

    public CustomParfum(String idPelanggan, String idCustom, String kelasParfum,
                        String notesParfum, double hargaCustom) {
        super(idPelanggan, nama:"Pelanggan Custom", email:"-", password:"-", alamat:"-", noTelp:"-");
        this.idCustom = idCustom;
        this.kelasParfum = kelasParfum;
        this.notesParfum = notesParfum;
        this.hargaCustom = hargaCustom;
    }

    @Override
    public void displayInfo() {
        System.out.println(getBasicInfo() + ", Custom ID: " + idCustom);
    }

    public String getKelasParfum() {
        throw new UnsupportedOperationException(message:"Not supported yet.");
    }

    public String getNotesParfum() {
        throw new UnsupportedOperationException(message:"Not supported yet.");
    }

    public String getHargaCustom() {
        throw new UnsupportedOperationException(message:"Not supported yet.");
    }
}
```

ScentiqueSystem.java



```java
import java.util.*;
import kelas.*;

public final class ScentiqueSystem {  // Final class
    private static final Scanner scanner = new Scanner(System.in);  // Final variable
    private static final ArrayList<Parfum> daftarParfum = new ArrayList<>();
    private static final ArrayList<Pelanggan> daftarPelanggan = new ArrayList<>();

    // Final method - cannot be overridden
    public static final void tampilkanInfoPelanggan(Pelanggan pelanggan) {
        pelanggan.displayInfo();
    }

    // Overloaded final method
    public static final void tampilkanInfoPelanggan(Pelanggan pelanggan, boolean showDetail) {
        if (showDetail) {
            System.out.println(x:"\n=== DETAIL PELANGGAN ===");
            pelanggan.displayInfo();
            if (pelanggan instanceof Admin) {    instanceof pattern can be used here
                ((Admin)pelanggan).kelolaSistem();
            } else if (pelanggan instanceof CustomParfum) {    instanceof pattern can be used here
                CustomParfum cp = (CustomParfum)pelanggan;
                System.out.println("Kelas Parfum: " + cp.getKelasParfum() +
                                   ", Notes: " + cp.getNotesParfum() +
                                   ", Harga Custom: " + cp.getHargaCustom());
            }
        } else {
            pelanggan.displayInfo();
        }
    }

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        // Initialize sample data
        daftarPelanggan.add(new Admin(idAdmin:"ADM001", nama:"Admin Scentique", email:"admin@scentique.com", password:"admin123"));
        daftarPelanggan.add(new CustomParfum(idPelanggan:"CUST001", idCustom:"ORDER123", kelasParfum:"Premium", notesParfu…"Aroma Kayu", 500000));
        daftarParfum.add(new Parfum(idParfum:"PRF001", namaParfum:"Floral Dream", hargaParfum:250000, stok:10));
        daftarParfum.add(new Parfum(idParfum:"PRF002", namaParfum:"Ocean Breeze", hargaParfum:300000, stok:5));

        int pilihan;
        do {
            System.out.println(x:"\n=== Scentique Management System ===");
            System.out.println(x:"1. Kelola Parfum");
            System.out.println(x:"2. Kelola Pelanggan");
            System.out.println(x:"3. Demo Polymorphism");
            System.out.println(x:"4. Exit");
            System.out.print(s:"Pilih menu: ");

            pilihan = getIntInput();
```



```java
    public static void main(String[] args) {

            pilihan = getIntInput();

            switch (pilihan) {    Convert switch to rule switch
                case 1:
                    kelolaParfum();
                    break;
                case 2:
                    kelolaPelanggan();
                    break;
                case 3:
                    demoPolymorphism();
                    break;
                case 4:
                    System.out.println(x:"Terima kasih telah menggunakan sistem!");
                    break;
                default:
                    System.out.println(x:"Pilihan tidak valid!");
            }
        } while (pilihan != 4);
    }

    private static void demoPolymorphism() {
        System.out.println(x:"\n=== DEMO POLYMORPHISM ===");

        // Method overriding demo
        System.out.println(x:"\n1. Method Overriding:");
        for (Pelanggan p : daftarPelanggan) {
            p.displayInfo();  // Calls the overridden method
        }

        // Method overloading demo
        System.out.println(x:"\n2. Method Overloading:");
        tampilkanInfoPelanggan(daftarPelanggan.get(index:0));  // Simple version
        tampilkanInfoPelanggan(daftarPelanggan.get(index:1), showDetail:true);  // Detailed version
    }

    private static void kelolaParfum() {
        int pilihan;
        do {
            System.out.println(x:"\n=== Kelola Parfum ===");
            System.out.println(x:"1. Tambah Parfum");
            System.out.println(x:"2. Lihat Parfum");
            System.out.println(x:"3. Hapus Parfum");
            System.out.println(x:"4. Kembali");
            System.out.print(s:"Pilih: ");

            pilihan = getIntInput();
```

```
            pilihan = getIntInput();

            switch (pilihan) {    Convert switch to rule switch
                case 1:
                    tambahParfum();
                    break;
                case 2:
                    lihatParfum();
                    break;
                case 3:
                    hapusParfum();
                    break;
                case 4:
                    break;
                default:
                    System.out.println(x:"Pilihan tidak valid!");
            }
        } while (pilihan != 4);
    }

    private static void kelolaPelanggan() {
        System.out.println(x:"\nFitur ini dalam pengembangan...");
    }

    private static void tambahParfum() {
        System.out.print(s:"\nMasukkan ID Parfum: ");
        String idParfum = getNonEmptyInput(errorMessage:"ID Parfum tidak boleh kosong!");

        System.out.print(s:"Masukkan Nama Parfum: ");
        String namaParfum = getNonEmptyInput(errorMessage:"Nama Parfum tidak boleh kosong!");

        System.out.print(s:"Masukkan Harga Parfum: ");
        double hargaParfum = getDoubleInput();

        System.out.print(s:"Masukkan Stok Parfum: ");
        int stok = getIntInput();

        daftarParfum.add(new Parfum(idParfum, namaParfum, hargaParfum, stok));
        System.out.println(x:"Parfum berhasil ditambahkan!");
    }

    private static void lihatParfum() {
        if (daftarParfum.isEmpty()) {
            System.out.println(x:"\nTidak ada parfum dalam sistem.");
            return;
        }

        System.out.println(x:"\nDaftar Parfum:");
        System.out.println(x:"=============================================");
        System.out.printf(format:"%-8s %-20s %-12s %-6s\n", ...args:"ID", "Nama", "Harga", "Stok");
        System.out.println(x:"=============================================");
```

```
        for (Parfum p : daftarParfum) {
            System.out.printf(format:"%-8s %-20s %-12.0f %-6d\n",
                              p.getIdParfum(),
                              p.getNamaParfum(),
                              p.getHargaParfum(),
                              p.getStok());
        }
    }

    private static void hapusParfum() {
        if (daftarParfum.isEmpty()) {
            System.out.println(x:"\nTidak ada parfum dalam sistem.");
            return;
        }

        lihatParfum();
        System.out.print(s:"\nMasukkan ID Parfum yang akan dihapus: ");
        String idParfum = scanner.nextLine();

        Iterator<Parfum> iterator = daftarParfum.iterator();
        while (iterator.hasNext()) {
            Parfum p = iterator.next();
            if (p.getIdParfum().equalsIgnoreCase(idParfum)) {
                iterator.remove();
                System.out.println(x:"Parfum berhasil dihapus.");
                return;
            }
        }

        System.out.println("Parfum dengan ID " + idParfum + " tidak ditemukan.");
    }

    // Utility methods
    private static int getIntInput() {
        while (!scanner.hasNextInt()) {
            System.out.println(x:"Input harus berupa bilangan bulat!");
            scanner.next();
        }
        int input = scanner.nextInt();
        scanner.nextLine();
        return input;
    }

    private static double getDoubleInput() {
        while (!scanner.hasNextDouble()) {
            System.out.println(x:"Input harus berupa bilangan!");
            scanner.next();
        }
        double input = scanner.nextDouble();
        scanner.nextLine();
        return input;
```

```
198
199     private static String getNonEmptyInput(String errorMessage) {
200         String input;
201         while ((input = scanner.nextLine().trim()).isEmpty()) {
202             System.out.println(errorMessage);
203         }
204         return input;
205     }
206 }
```