

NAMA : Davina Putri Ananta

NIM : 2309106002

Kelas : IT A'23

SS program Pesanan.java

```
1  package kelas;
2
3  interface Pembayaran {
4      void prosesPembayaran(double jumlah) throws IllegalArgumentException;
5      String generateInvoice();
6  }
7
8  public class Pesanan implements Pembayaran {
9      private String idPesanan;
10     private String idPelanggan;
11     private String idAdmin;
12     private String status;
13     private String tanggalPesanan;
14     private double totalHarga;
15
16     public Pesanan(String idPesanan, String idPelanggan, String idAdmin,
17         String status, String tanggalPesanan, double totalHarga) {
18         this.idPesanan = idPesanan;
19         this.idPelanggan = idPelanggan;
20         this.idAdmin = idAdmin;
21         this.status = status;
22         this.tanggalPesanan = tanggalPesanan;
23         this.totalHarga = totalHarga;
24     }
25
26     public String getIdPesanan() {
27         return idPesanan;
28     }
29
30     public void setIdPesanan(String idPesanan) {
31         this.idPesanan = idPesanan;
32     }
33
34     public String getIdPelanggan() {
35         return idPelanggan;
36     }
37 }
```

```
37
38     public void setIdPelanggan(String idPelanggan) {
39         |     this.idPelanggan = idPelanggan;
40     }
41
42     public String getIdAdmin() {
43         |     return idAdmin;
44     }
45
46     public void setIdAdmin(String idAdmin) {
47         |     this.idAdmin = idAdmin;
48     }
49
50     public String getStatus() {
51         |     return status;
52     }
53
54     public void setStatus(String status) {
55         |     this.status = status;
56     }
57
58     public String getTanggalPesanan() {
59         |     return tanggalPesanan;
60     }
61
62     public void setTanggalPesanan(String tanggalPesanan) {
63         |     this.tanggalPesanan = tanggalPesanan;
64     }
65
66     public double getTotalHarga() {
67         |     return totalHarga;
68     }
69
```

```

69
70     public void setTotalHarga(double totalHarga) {
71         this.totalHarga = totalHarga;
72     }
73
74     @Override
75     public void prosesPembayaran(double jumlah) throws IllegalArgumentException {
76         if (jumlah <= 0) {
77             throw new IllegalArgumentException(s:"Jumlah pembayaran harus positif");
78         }
79         if (jumlah < totalHarga) {
80             System.out.println("[PEMBAYARAN] Pembayaran kurang: Rp" + (totalHarga - jumlah));
81         } else {
82             System.out.println("[PEMBAYARAN] Berhasil! Kembalian: Rp" + (jumlah - totalHarga));
83             this.status = "Lunas";
84         }
85     }
86
87     @Override
88     public String generateInvoice() {
89         return "=== INVOICE PESANAN ===\n" +
90             "ID Pesanan      : " + idPesanan + "\n" +
91             "Tanggal         : " + tanggalPesanan + "\n" +
92             "Total Harga      : Rp" + totalHarga + "\n" +
93             "Status           : " + status + "\n" +
94             "ID Pelanggan     : " + idPelanggan;
95     }
96 }

```

SS program customParfum.java

```

1 package kelas;
2
3 public class CustomParfum extends Pelanggan {
4     public String idCustom;
5     public String kelasParfum;
6     public String notesParfum;
7     public double hargaCustom;
8
9     public CustomParfum(String idPelanggan, String idCustom, String kelasParfum,
10         String notesParfum, double hargaCustom) {
11         super(idPelanggan, nama:"Pelanggan Custom", email:"-", password:"-", alamat:"-", noTelp:"-");
12         this.idCustom = idCustom;
13         this.kelasParfum = kelasParfum;
14         this.notesParfum = notesParfum;
15         this.hargaCustom = hargaCustom;
16     }
17     @Override
18     public void displayInfo() {
19         System.out.println("Pelanggan Custom - ID: " + getIdPelanggan() +
20             ", Custom ID: " + idCustom + ", Kelas: " + kelasParfum);
21     }
22     public String getkelasparfum() {
23         return this.kelasParfum;
24     }
25
26     public String getIdCustom() {
27         return idCustom;
28     }
29
30     public void setIdCustom(String idCustom) {
31         this.idCustom = idCustom;
32     }
33
34     public String getKelasParfum() {
35         return kelasParfum;
36     }
37
38     public void setKelasParfum(String kelasParfum) {
39         this.kelasParfum = kelasParfum;
40     }
41
42     public String getNotesParfum() {
43         return notesParfum;
44     }
45
46     public void setNotesParfum(String notesParfum) {
47         this.notesParfum = notesParfum;
48     }
49
50     public double getHargaCustom() {
51         return hargaCustom;
52     }
53
54     public void setHargaCustom(double hargaCustom) {
55         this.hargaCustom = hargaCustom;
56     }
57 }

```

SS program scentiquesystem

```

src > ScentiqueSystem.java > Java > ScentiqueSystem > main(String[] args)
1  import java.util.*;
2  import kelas.*;
3
4  public class ScentiqueSystem {
5      static Scanner scanner = new Scanner(System.in);
6      static ArrayList<Parfum> daftarParfum = new ArrayList<>();
7      static ArrayList<Pelanggan> daftarPelanggan = new ArrayList<>();
8
9      public static final double DISKON_MEMBER = 0.1;
10     public static double hitungHargaSetelahDiskon(double harga, boolean isMember) {
11         return isMember ? harga * (1 - DISKON_MEMBER) : harga;
12     }
13
14     public static void tampilkanInfoPelanggan(Pelanggan pelanggan) {
15         pelanggan.displayInfo();
16     }
17
18     public static void tampilkanInfoPelanggan(Pelanggan pelanggan, boolean showDetail) {
19         if (showDetail) {
20             System.out.println(x: "\n=== DETAIL PELANGGAN ===");
21             pelanggan.displayInfo();
22             if (pelanggan instanceof Admin) { instance of pattern can be used here
23                 ((Admin)pelanggan).kelolaSistem();
24             } else if (pelanggan instanceof CustomParfum) { instance of pattern can be used here
25                 System.out.println("Kelas Parfum: " + ((CustomParfum)pelanggan).getkelasparfum());
26             }
27         } else {
28             pelanggan.displayInfo();
29         }
30     }
31
32     Run main | Debug main | Run | Debug
33     public static void main(String[] args) {
34         daftarPelanggan.add(new Admin(idAdmin:"ADM001", nama:"Admin Scentique", email:"admin@scentique.com", password:"admin123"));
35         daftarPelanggan.add(new CustomParfum(idPelanggan:"CUST001", idCustom:"ORDER123", kelasParfum:"Premium", notesParfu:"Aroma Kayu", 500000));
36
37         Pesanan demoPesanan = new Pesanan(idPesanan:"ORDER999", idPelanggan:"CUST001", idAdmin:"ADM001", status:"Pending", tanggal:"2023-11-20", 250000);
38         try {
39             System.out.println(x: "\n=== DEMO INTERFACE ===");
40             demoPesanan.prosesPembayaran(jumlah:300000);
41             System.out.println(demoPesanan.generateInvoice());
42         } catch (IllegalArgumentException e) {
43             System.out.println("Error pembayaran: " + e.getMessage());
44         }
45
46         System.out.println(x: "\n=== DEMO STATIC METHOD ===");
47         System.out.println("Harga 100000 untuk member: " + hitungHargaSetelahDiskon(harga:100000, isMember:true));
48         System.out.println("Harga 100000 untuk non-member: " + hitungHargaSetelahDiskon(harga:100000, isMember:false));
49
50         int pilihan;
51         do {
52             System.out.println(x: "\nSistem Pemesanan dan Custom Parfum di Scentique");
53             System.out.println(x: "1. Tambah Parfum");
54             System.out.println(x: "2. Lihat Parfum");
55             System.out.println(x: "3. Hapus Parfum");
56             System.out.println(x: "4. Lihat Contoh Polymorphism");
57             System.out.println(x: "5. Keluar");
58             System.out.print(s: "Pilih menu: ");
59
60             try {
61                 pilihan = Integer.parseInt(scanner.nextLine());
62             } catch (NumberFormatException e) {
63                 System.out.println(x: "Input harus berupa angka!");
64                 pilihan = 0;
65                 continue;
66             }
67         }

```

```

65     }
66
67     switch (pilihan) {      Convert switch to rule switch
68     case 1:
69         tambahParfum();
70         break;
71     case 2:
72         lihatParfum();
73         break;
74     case 3:
75         hapusParfum();
76         break;
77     case 4:
78         System.out.println(x:"\nDemo Polymorphism:");
79         for (Pelanggan p : daftarPelanggan) {
80             tampilkanInfoPelanggan(p, showDetail:true);
81         }
82         break;
83     case 5:
84         System.out.println(x:"Terima kasih telah menggunakan sistem Scentique!");
85         break;
86     default:
87         System.out.println(x:"Pilihan tidak valid.");
88     }
89     } while (pilihan != 5);
90 }
91
92 static void tambahParfum() {
93     try {
94         System.out.print(s:"Masukkan ID Parfum: ");
95         String idParfum = scanner.nextLine().trim();
96         if (idParfum.isEmpty()) {
97             throw new IllegalArgumentException(s:"ID Parfum tidak boleh kosong!");
98         }
99
100         System.out.print(s:"Masukkan Nama Parfum: ");
101         String namaParfum = scanner.nextLine().trim();
102         if (namaParfum.isEmpty()) {
103             throw new IllegalArgumentException(s:"Nama Parfum tidak boleh kosong!");
104         }
105
106         System.out.print(s:"Masukkan Harga Parfum: ");
107         double hargaParfum = Double.parseDouble(scanner.nextLine());
108         if (hargaParfum <= 0) {
109             throw new IllegalArgumentException(s:"Harga harus lebih dari 0");
110         }
111
112         System.out.print(s:"Masukkan Stok Parfum: ");
113         int stok = Integer.parseInt(scanner.nextLine());
114         if (stok < 0) {
115             throw new IllegalArgumentException(s:"Stok tidak boleh negatif");
116         }
117
118         daftarParfum.add(new Parfum(idParfum, namaParfum, hargaParfum, stok));
119         System.out.println(x:"Parfum berhasil ditambahkan!");
120
121     } catch (NumberFormatException e) {
122         System.out.println(x:"Input harus berupa angka!");
123     } catch (IllegalArgumentException e) {
124         System.out.println("Error: " + e.getMessage());
125     }
126 }
127

```

```

127
128     static void lihatParfum() {
129         if (daftarParfum.isEmpty()) {
130             System.out.println(x:"Tidak ada parfum dalam daftar.");
131         } else {
132             System.out.println(x:"\nDaftar Parfum:");
133             for (Parfum p : daftarParfum) {
134                 System.out.printf(format:"ID: %s | Nama: %-15s | Harga: Rp%,.2f | Stok: %d%n",
135                     p.getIdParfum(), p.getNamaParfum(), p.getHargaParfum(), p.getStok());
136             }
137         }
138     }
139
140     static void hapusParfum() {
141         if (daftarParfum.isEmpty()) {
142             System.out.println(x:"Tidak ada parfum dalam daftar.");
143             return;
144         }
145
146         try {
147             System.out.print(s:"Masukkan ID Parfum yang ingin dihapus: ");
148             String idParfum = scanner.nextLine().trim();
149             boolean ditemukan = false;
150
151             for (int i = 0; i < daftarParfum.size(); i++) {
152                 if (daftarParfum.get(i).getIdParfum().equalsIgnoreCase(idParfum)) {
153                     daftarParfum.remove(i);
154                     System.out.println(x:"Parfum berhasil dihapus.");
155                     ditemukan = true;
156                     break;
157                 }
158             }
159
160             if (!ditemukan) {
161                 System.out.println(x:"Parfum tidak ditemukan.");
162             }
163         } catch (Exception e) {
164             System.out.println("Error: " + e.getMessage());
165         }
166     }
167 }

```

Kode lainnya masih sama