

Nama : Davina Putri Ananta

NIM : 2309106002

Kelas: IT A'23

Posttttest: 5

---

SS Pelanggan.java

```
1 package kelas;  
2  
3 public abstract class Pelanggan {  
4     private final String idPelanggan;  
5     private String nama;  
6     private String email;  
7     private String password;  
8     private String alamat;  
9     private String noTelp;  
10  
11     public Pelanggan(String idPelanggan, String nama, String email,  
12         String password, String alamat, String noTelp) {  
13         this.idPelanggan = idPelanggan;  
14         this.nama = nama;  
15         this.email = email;  
16         this.password = password;  
17         this.alamat = alamat;  
18         this.noTelp = noTelp;  
19     }  
20  
21     public abstract void displayInfo();  
22  
23     public final String getBasicInfo() {  
24         return "ID: " + idPelanggan + ", Nama: " + nama;  
25     }  
26  
27     public final String getIdPelanggan() {  
28         return idPelanggan;  
29     }  
30     public String getNama() {  
31         return nama;  
32     }  
33  
34     public void setNama(String nama) {  
35         this.nama = nama;  
36     }  
37  
38     public String getEmail() {  
39         return email;  
40     }  
41  
42     public void setEmail(String email) {  
43         this.email = email;  
44     }  
45 }
```

```

45
46     public String getPassword() {
47         return password;
48     }
49
50     public void setPassword(String password) {
51         this.password = password;
52     }
53
54     public String getAlamat() {
55         return alamat;
56     }
57
58     public void setAlamat(String alamat) {
59         this.alamat = alamat;
60     }
61
62     public String getNoTelp() {
63         return noTelp;
64     }
65
66     public void setNoTelp(String noTelp) {
67         this.noTelp = noTelp;
68     }
69
70     public final void resetPassword(String newPassword) {
71         if (newPassword.length() >= 8) {
72             this.password = newPassword;
73             System.out.println(x:"Password berhasil direset");
74         } else {
75             System.out.println(x:"Password harus minimal 8 karakter");
76         }
77     }
78 }

```

Parfum.java

```

1  package kelas;
2
3  public final class Parfum {
4      private final String idParfum;
5      private String namaParfum;
6      private final double hargaParfum;
7      private int stok;
8
9      public Parfum(String idParfum, String namaParfum, double hargaParfum, int stok) {
10         this.idParfum = idParfum;
11         this.namaParfum = namaParfum;
12         this.hargaParfum = hargaParfum;
13         this.stok = stok;
14     }
15
16     public final double getHargaWithTax() {
17         return hargaParfum * 1.11;
18     }
19
20     public String getIdParfum() {
21         return idParfum;
22     }
23
24     public String getNamaParfum() {
25         return namaParfum;
26     }
27
28     public void setNamaParfum(String namaParfum) {
29         this.namaParfum = namaParfum;
30     }
31
32     public double getHargaParfum() {
33         return hargaParfum;
34     }
35
36     public int getStok() {
37         return stok;
38     }
39
40     public void setStok(int stok) {
41         this.stok = stok;
42     }
43
44     public final boolean isTersedia() {
45         return stok > 0;
46     }
47
48     @Override
49     public String toString() {
50         return "Parfum[id=" + idParfum +
51             ", nama=" + namaParfum +
52             ", harga=" + hargaParfum +
53             ", stok=" + stok + "]";
54     }
55 }

```

## Admin.java

```
src > kelas > Admin.java > Language Support for Java(TM) by Red Hat > Admin
1  package kelas;
2
3  public class Admin extends Pelanggan {
4      public Admin(String idAdmin, String nama, String email, String password) {
5          super(idAdmin, nama, email, password, alamat: "-", noTelp: "-");
6      }
7
8      @Override
9      public void displayInfo() {
10         System.out.println(getBasicInfo() + " (Role: Administrator)");
11     }
12
13     public final void kelolaSistem() { // final method
14         System.out.println("Admin " + getName() + " sedang mengelola sistem...");
15     }
16 }
```

## CustomParfum.java

```
1  package kelas;
2
3  public class CustomParfum extends Pelanggan {
4      private final String idCustom; // final attribute
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
18     @Override
19     public void displayInfo() {
20         System.out.println(getBasicInfo() + ", Custom ID: " + idCustom);
21     }
22
23     public String getKelasParfum() {
24         throw new UnsupportedOperationException(message: "Not supported yet.");
25     }
26
27     public String getNotesParfum() {
28         throw new UnsupportedOperationException(message: "Not supported yet.");
29     }
30
31     public String getHargaCustom() {
32         throw new UnsupportedOperationException(message: "Not supported yet.");
33     }
34 }
```

## ScentiqueSystem.java

```
1 import java.util.*;
2 import kelas.*;
3
4 public final class ScentiqueSystem { // Final class
5     private static final Scanner scanner = new Scanner(System.in); // Final variable
6     private static final ArrayList<Parfum> daftarParfum = new ArrayList<>();
7     private static final ArrayList<Pelanggan> daftarPelanggan = new ArrayList<>();
8
9     // Final method - cannot be overridden
10    public static final void tampilkanInfoPelanggan(Pelanggan pelanggan) {
11        pelanggan.displayInfo();
12    }
13
14    // Overloaded final method
15    public static final void tampilkanInfoPelanggan(Pelanggan pelanggan, boolean showDetail) {
16        if (showDetail) {
17            System.out.println(x:"\n=== DETAIL PELANGGAN ===");
18            pelanggan.displayInfo();
19            if (pelanggan instanceof Admin) { // instanceof pattern can be used here
20                ((Admin)pelanggan).kelolaSistem();
21            } else if (pelanggan instanceof CustomParfum) { // instanceof pattern can be used here
22                CustomParfum cp = (CustomParfum)pelanggan;
23                System.out.println("Kelas Parfum: " + cp.getKelasParfum() +
24                    ", Notes: " + cp.getNotesParfum() +
25                    ", Harga Custom: " + cp.getHargaCustom());
26            }
27        } else {
28            pelanggan.displayInfo();
29        }
30    }
31
32    Run main | Debug main | Run | Debug
33    public static void main(String[] args) {
34        // Initialize sample data
35        daftarPelanggan.add(new Admin(idAdmin:"ADM001", nama:"Admin Scentique", email:"admin@scentique.com", password:"admin123"));
36        daftarPelanggan.add(new CustomParfum(idPelanggan:"CUST001", idCustom:"ORDER123", kelasParfum:"Premium", notesParfu:"Aroma Kayu", 500000));
37        daftarParfum.add(new Parfum(idParfum:"PRF001", namaParfum:"Floral Dream", hargaParfum:250000, stok:10));
38        daftarParfum.add(new Parfum(idParfum:"PRF002", namaParfum:"Ocean Breeze", hargaParfum:300000, stok:5));
39
40        int pilihan;
41        do {
42            System.out.println(x:"\n=== Scentique Management System ===");
43            System.out.println(x:"1. Kelola Parfum");
44            System.out.println(x:"2. Kelola Pelanggan");
45            System.out.println(x:"3. Demo Polymorphism");
46            System.out.println(x:"4. Exit");
47            System.out.print(s:"Pilih menu: ");
48
49            pilihan = getIntInput();
50        } while (pilihan != 4);
51    }
52
53    private static void demoPolymorphism() {
54        System.out.println(x:"\n=== DEMO POLYMORPHISM ===");
55
56        // Method overriding demo
57        System.out.println(x:"\n1. Method Overriding:");
58        for (Pelanggan p : daftarPelanggan) {
59            p.displayInfo(); // Calls the overridden method
60        }
61
62        // Method overloading demo
63        System.out.println(x:"\n2. Method Overloading:");
64        tampilkanInfoPelanggan(daftarPelanggan.get(index:0)); // Simple version
65        tampilkanInfoPelanggan(daftarPelanggan.get(index:1), showDetail:true); // Detailed version
66    }
67
68    private static void kelolaParfum() {
69        int pilihan;
70        do {
71            System.out.println(x:"\n=== Kelola Parfum ===");
72            System.out.println(x:"1. Tambah Parfum");
73            System.out.println(x:"2. Lihat Parfum");
74            System.out.println(x:"3. Hapus Parfum");
75            System.out.println(x:"4. Kembali");
76            System.out.print(s:"Pilih: ");
77
78            pilihan = getIntInput();
79        } while (pilihan != 4);
80    }
81
82    private static void kelolaPelanggan() {
83        int pilihan;
84        do {
85            System.out.println(x:"\n=== Kelola Pelanggan ===");
86            System.out.println(x:"1. Tambah Pelanggan");
87            System.out.println(x:"2. Lihat Pelanggan");
88            System.out.println(x:"3. Hapus Pelanggan");
89            System.out.println(x:"4. Kembali");
90            System.out.print(s:"Pilih: ");
91
92            pilihan = getIntInput();
93        } while (pilihan != 4);
94    }
95}
```

```
32    public static void main(String[] args) {
47
48        pilihan = getIntInput();
49
50        switch (pilihan) { // Convert switch to rule switch
51            case 1:
52                kelolaParfum();
53                break;
54            case 2:
55                kelolaPelanggan();
56                break;
57            case 3:
58                demoPolymorphism();
59                break;
60            case 4:
61                System.out.println(x:"Terima kasih telah menggunakan sistem!");
62                break;
63            default:
64                System.out.println(x:"Pilihan tidak valid!");
65        }
66        while (pilihan != 4);
67    }
68
69    private static void demoPolymorphism() {
70        System.out.println(x:"\n=== DEMO POLYMORPHISM ===");
71
72        // Method overriding demo
73        System.out.println(x:"\n1. Method Overriding:");
74        for (Pelanggan p : daftarPelanggan) {
75            p.displayInfo(); // Calls the overridden method
76        }
77
78        // Method overloading demo
79        System.out.println(x:"\n2. Method Overloading:");
80        tampilkanInfoPelanggan(daftarPelanggan.get(index:0)); // Simple version
81        tampilkanInfoPelanggan(daftarPelanggan.get(index:1), showDetail:true); // Detailed version
82    }
83
84    private static void kelolaParfum() {
85        int pilihan;
86        do {
87            System.out.println(x:"\n=== Kelola Parfum ===");
88            System.out.println(x:"1. Tambah Parfum");
89            System.out.println(x:"2. Lihat Parfum");
90            System.out.println(x:"3. Hapus Parfum");
91            System.out.println(x:"4. Kembali");
92            System.out.print(s:"Pilih: ");
93
94            pilihan = getIntInput();
95        } while (pilihan != 4);
96    }
97
98    private static void kelolaPelanggan() {
99        int pilihan;
100        do {
101            System.out.println(x:"\n=== Kelola Pelanggan ===");
102            System.out.println(x:"1. Tambah Pelanggan");
103            System.out.println(x:"2. Lihat Pelanggan");
104            System.out.println(x:"3. Hapus Pelanggan");
105            System.out.println(x:"4. Kembali");
106            System.out.print(s:"Pilih: ");
107
108            pilihan = getIntInput();
109        } while (pilihan != 4);
110    }
111}
```

```

93
94     pilihan = getIntInput();
95
96     switch (pilihan) { // Convert switch to rule switch
97     case 1:
98         tambahParfum();
99         break;
100     case 2:
101         lihatParfum();
102         break;
103     case 3:
104         hapusParfum();
105         break;
106     case 4:
107         break;
108     default:
109         System.out.println(x:"Pilihan tidak valid!");
110     }
111 } while (pilihan != 4);
112 }
113
114 private static void kelolaPelanggan() {
115     System.out.println(x:"\nfitur ini dalam pengembangan...");
116 }
117
118 private static void tambahParfum() {
119     System.out.print(s:"\nMasukkan ID Parfum: ");
120     String idParfum = getNonEmptyInput(errorMessage:"ID Parfum tidak boleh kosong!");
121
122     System.out.print(s:"Masukkan Nama Parfum: ");
123     String namaParfum = getNonEmptyInput(errorMessage:"Nama Parfum tidak boleh kosong!");
124
125     System.out.print(s:"Masukkan Harga Parfum: ");
126     double hargaParfum = getDoubleInput();
127
128     System.out.print(s:"Masukkan Stok Parfum: ");
129     int stok = getIntInput();
130
131     daftarParfum.add(new Parfum(idParfum, namaParfum, hargaParfum, stok));
132     System.out.println(x:"Parfum berhasil ditambahkan!");
133 }
134
135 private static void lihatParfum() {
136     if (daftarParfum.isEmpty()) {
137         System.out.println(x:"\nTidak ada parfum dalam sistem.");
138         return;
139     }
140
141     System.out.println(x:"\nDaftar Parfum:");
142     System.out.println(x:"=====");
143     System.out.printf(format:"%-8s %-20s %-12s %-6s\n", ...args:"ID", "Nama", "Harga", "Stok");
144     System.out.println(x:"=====");
145 }

```

```

145
146     for (Parfum p : daftarParfum) {
147         System.out.printf(format:"%-8s %-20s %-12.0f %-6d\n",
148             p.getIdParfum(),
149             p.getNamaParfum(),
150             p.getHargaParfum(),
151             p.getStok());
152     }
153 }
154
155 private static void hapusParfum() {
156     if (daftarParfum.isEmpty()) {
157         System.out.println(x:"\nTidak ada parfum dalam sistem.");
158         return;
159     }
160
161     lihatParfum();
162     System.out.print(s:"\nMasukkan ID Parfum yang akan dihapus: ");
163     String idParfum = scanner.nextLine();
164
165     Iterator<Parfum> iterator = daftarParfum.iterator();
166     while (iterator.hasNext()) {
167         Parfum p = iterator.next();
168         if (p.getIdParfum().equalsIgnoreCase(idParfum)) {
169             iterator.remove();
170             System.out.println(x:"Parfum berhasil dihapus.");
171             return;
172         }
173     }
174
175     System.out.println("Parfum dengan ID " + idParfum + " tidak ditemukan.");
176 }
177
178 // Utility methods
179 private static int getIntInput() {
180     while (!scanner.hasNextInt()) {
181         System.out.println(x:"Input harus berupa bilangan bulat!");
182         scanner.next();
183     }
184     int input = scanner.nextInt();
185     scanner.nextLine();
186     return input;
187 }
188
189 private static double getDoubleInput() {
190     while (!scanner.hasNextDouble()) {
191         System.out.println(x:"Input harus berupa bilangan!");
192         scanner.next();
193     }
194     double input = scanner.nextDouble();
195     scanner.nextLine();
196     return input;
197 }

```

```
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 }
```