



```
1  import java.util.ArrayList;
2  import java.util.Scanner;
3
4  // Superclass Produk
5  class Produk {
6      protected final int id;
7      protected String nama;
8      protected double harga;
9
10     public Produk(int id, String nama, double harga) {
11         this.id = id;
12         this.nama = (nama.isEmpty()) ? "Tidak Diketahui" : nama;
13         this.harga = (harga < 0) ? 0 : harga;
14     }
15
16     public int getId() {
17         return id;
18     }
19
20     public String getNama() {
21         return nama;
22     }
23
24     public double getHarga() {
25         return harga;
26     }
27
28     public void setNama(String nama) {
29         if (nama.isEmpty()) {
30             throw new IllegalArgumentException("Nama tidak boleh kosong!");
31         }
32         this.nama = nama;
33     }
34
35     public void setHarga(double harga) {
36         if (harga < 0) {
37             throw new IllegalArgumentException("Harga tidak boleh negatif!");
38         }
39         this.harga = harga;
40     }
41
42     @Override
43     public String toString() {
44         return "ID: " + id + ", Nama: " + nama + ", Harga: Rp" + harga;
45     }
46 }
```

```

1  // Subclass Parfum
2  class Parfum extends Produk {
3      private String aroma;
4
5      public Parfum(int id, String nama, double harga, String aroma) {
6          super(id, nama, harga);
7          this.aroma = (aroma.isEmpty()) ? "Tidak Diketahui" : aroma;
8      }
9
10     public String getAroma() {
11         return aroma;
12     }
13
14     public void setAroma(String aroma) {
15         if (aroma.isEmpty()) {
16             throw new IllegalArgumentException("Aroma tidak boleh kosong!");
17         }
18         this.aroma = aroma;
19     }
20
21     @Override
22     public String toString() {
23         return super.toString() + ", Aroma: " + aroma;
24     }
25 }
26
27 // Subclass ParfumEdisiTerbatas
28 class ParfumEdisiTerbatas extends Parfum {
29     private String edisi;
30
31     public ParfumEdisiTerbatas(int id, String nama, double harga, String aroma, String edisi) {
32         super(id, nama, harga, aroma);
33         this.edisi = (edisi.isEmpty()) ? "Tidak Diketahui" : edisi;
34     }
35
36     public String getEdisi() {
37         return edisi;
38     }
39
40     public void setEdisi(String edisi) {
41         if (edisi.isEmpty()) {
42             throw new IllegalArgumentException("Edisi tidak boleh kosong!");
43         }
44         this.edisi = edisi;
45     }
46
47     @Override
48     public String toString() {
49         return super.toString() + ", Edisi: " + edisi;
50     }
51 }
52
53 // Main program
54 public class SistemPenjualanParfum {
55     private static final ArrayList<Produk> daftarProduk = new ArrayList<>();
56     private static int nextId = 1;
57     private static final Scanner scanner = new Scanner(System.in);
58
59     public static void main(String[] args)

```

```

1 // Main program
2 public class SistemPenjualanParfum {
3     private static final ArrayList<Produk> daftarProduk = new ArrayList<>();
4     private static int nextId = 1;
5     private static final Scanner scanner = new Scanner(System.in);
6
7     public static void main(String[] args) {
8         tambahParfum("Tester Citrus", 120000, "Citrus");
9         while (true) {
10             System.out.println("\n== Sistem Penjualan Parfum ==");
11             System.out.println("1. Tambah Parfum");
12             System.out.println("2. Tambah Parfum Edisi Terbatas");
13             System.out.println("3. Tampilkan Produk");
14             System.out.println("4. Edit Produk");
15             System.out.println("5. Hapus Produk");
16             System.out.println("6. Tampilkan Deskripsi Produk (Polymorphism)");
17             System.out.println("7. Keluar");
18             System.out.print("Pilih menu: ");
19
20             int pilihan = scanner.nextInt();
21             scanner.nextLine();
22
23             switch (pilihan) {
24                 case 1 -> tambahParfum();
25                 case 2 -> tambahParfumEdisiTerbatas();
26                 case 3 -> tampilkanProduk();
27                 case 4 -> editProduk();
28                 case 5 -> hapusProduk();
29                 case 6 -> tampilkanDeskripsiProduk();
30                 case 7 -> {
31                     System.out.println("Terima kasih telah menggunakan sistem ini!");
32                     return;
33                 }
34                 default -> System.out.println("Pilihan tidak valid, coba lagi.");
35             }
36         }
37     }
38
39     private static void tambahParfum() {
40         System.out.print("Masukkan nama parfum: ");
41         String nama = scanner.nextLine();
42         System.out.print("Masukkan harga parfum: ");
43         double harga = scanner.nextDouble();
44         scanner.nextLine();
45         System.out.print("Masukkan aroma parfum: ");
46         String aroma = scanner.nextLine();
47
48         daftarProduk.add(new Parfum(nextId++, nama, harga, aroma));
49         System.out.println("Parfum berhasil ditambahkan!");
50     }
51
52     private static void tambahParfum(String nama, double harga, String aroma) {
53         daftarProduk.add(new Parfum(nextId++, nama, harga, aroma));
54         System.out.println("[OTOMATIS] Parfum berhasil ditambahkan!");
55     }
56
57     private static void tambahParfumEdisiTerbatas() {
58         System.out.print("Masukkan nama parfum: ");
59         String nama = scanner.nextLine();
60         System.out.print("Masukkan harga parfum: ");
61         double harga = scanner.nextDouble();
62         scanner.nextLine();
63         System.out.print("Masukkan aroma parfum: ");
64         String aroma = scanner.nextLine();
65         System.out.print("Masukkan edisi parfum: ");
66         String edisi = scanner.nextLine();
67
68         daftarProduk.add(new ParfumEdisiTerbatas(nextId++, nama, harga, aroma, edisi));
69         System.out.println("Parfum edisi terbatas berhasil ditambahkan!");
70     }
71
72     private static void tampilkanProduk() {
73         if (daftarProduk.isEmpty()) {
74             System.out.println("Tidak ada produk yang tersedia.");
75             return;
76         }
77         for (Produk produk : daftarProduk) {
78             System.out.println(produk);
79         }
80     }
81
82     private static void editProduk() {
83         System.out.print("Masukkan ID produk yang ingin diedit: ");
84         int id = scanner.nextInt();
85         scanner.nextLine();
86
87         for (Produk produk : daftarProduk) {
88             if (produk.getId() == id) {
89                 System.out.print("Masukkan nama baru: ");
90                 produk.setNama(scanner.nextLine());
91                 System.out.print("Masukkan harga baru: ");
92                 produk.setHarga(scanner.nextDouble());
93                 scanner.nextLine();
94                 System.out.println("Produk berhasil diperbarui!");
95                 return;
96             }
97         }
98         System.out.println("Produk dengan ID tersebut tidak ditemukan.");
99     }
100
101     private static void hapusProduk() {
102         System.out.print("Masukkan ID produk yang ingin dihapus: ");
103         int id = scanner.nextInt();
104         scanner.nextLine();
105
106         boolean removed = daftarProduk.removeIf(produk -> produk.getId() == id);
107         if (removed) {
108             System.out.println("Produk berhasil dihapus!");
109         } else {
110             System.out.println("Produk tidak ditemukan.");
111         }
112     }
113
114     private static void tampilkanDeskripsiProduk() {
115         if (daftarProduk.isEmpty()) {
116             System.out.println("Tidak ada produk untuk ditampilkan.");
117             return;
118         }
119
120         System.out.println("Deskripsi Produk:");
121         for (Produk produk : daftarProduk) {
122             System.out.println(produk.getDeskripsiProd

```

```
1
2     private static void tampilkanDeskripsiProduk() {
3         if (daftarProduk.isEmpty()) {
4             System.out.println("Tidak ada produk untuk ditampilkan.");
5             return;
6         }
7
8         System.out.println("Deskripsi Produk:");
9         for (Produk produk : daftarProduk) {
10             System.out.println(getDeskripsiProduk(produk));
11         }
12     }
13
14     // Method overloading - polimorfisme dinamis
15     private static String getDeskripsiProduk(Produk produk) {
16         if (produk == null) {
17             return "Produk tidak dikenali (null)";
18         } else if (produk instanceof ParfumEdisiTerbatas p) {
19             return "Parfum Edisi Terbatas: " + p.getNama() + " | Aroma: " + p.getAroma() + " | Edisi: " + p.getEdisi();
20         } else if (produk instanceof Parfum p) {
21             return "Parfum Biasa: " + p.getNama() + " | Aroma: " + p.getAroma();
22         } else {
23             return "Produk Umum: " + produk.getNama();
24         }
25     }
26 }
27
```