



```
1  // Subclass Parfum
2  class Parfum extends Produk {
3      private final 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     @Override
15     public String deskripsi() {
16         return "Parfum Biasa: " + getNama() + " | Aroma: " + getAroma();
17     }
18
19     @Override
20     public String toString() {
21         return super.toString() + ", Aroma: " + aroma;
22     }
23 }
24
```

```

1 // Final Main Program
2 public final 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
10        while (true) {
11            System.out.println("\n=== Sistem Penjualan Parfum ===");
12            System.out.println("1. Tambah Parfum");
13            System.out.println("2. Tambah Parfum Edisi Terbatas");
14            System.out.println("3. Tampilkan Produk");
15            System.out.println("4. Edit Produk");
16            System.out.println("5. Hapus Produk");
17            System.out.println("6. Tampilkan Deskripsi Produk (Polymorphism)");
18            System.out.println("7. Keluar");
19            System.out.print("Pilih menu: ");
20
21            int pilihan = scanner.nextInt();
22            scanner.nextLine();
23
24            switch (pilihan) {
25                case 1 -> tambahParfum();
26                case 2 -> tambahParfumEdisiTerbatas();
27                case 3 -> tampilkanProduk();
28                case 4 -> editProduk();
29                case 5 -> hapusProduk();
30                case 6 -> tampilkanDeskripsiProduk();
31                case 7 -> {
32                    System.out.println("Terima kasih telah menggunakan sistem ini!");
33                    return;
34                }
35                default -> System.out.println("Pilihan tidak valid, coba lagi.");
36            }
37        }
38    }
39
40    private static void tambahParfum() {
41        System.out.print("Masukkan nama parfum: ");
42        String nama = scanner.nextLine();
43        System.out.print("Masukkan harga parfum: ");
44        double harga = scanner.nextDouble();
45        scanner.nextLine();
46        System.out.print("Masukkan aroma parfum: ");
47        String aroma = scanner.nextLine();
48
49        daftarProduk.add(new Parfum(nextId++, nama, harga, aroma));
50        System.out.println("Parfum berhasil ditambahkan!");
51    }
52
53    private static void tambahParfum(String nama, double harga, String aroma) {
54        daftarProduk.add(new Parfum(nextId++, nama, harga, aroma));
55        System.out.println("[OTOMATIS] Parfum berhasil ditambahkan!");
56    }
57
58    private static void tambahParfumEdisiTerbatas() {
59        System.out.print("Masukkan nama parfum: ");
60        String nama = scanner.nextLine();
61        System.out.print("Masukkan harga parfum: ");
62        double harga = scanner.nextDouble();
63        scanner.nextLine();

```

```
1 private static void tampilkanProduk() {
2     if (daftarProduk.isEmpty()) {
3         System.out.println("Tidak ada produk yang tersedia.");
4         return;
5     }
6     for (Produk produk : daftarProduk) {
7         System.out.println(produk);
8     }
9 }
10
11 private static void editProduk() {
12     System.out.print("Masukkan ID produk yang ingin diedit: ");
13     int id = scanner.nextInt();
14     scanner.nextLine();
15
16     for (Produk produk : daftarProduk) {
17         if (produk.getId() == id) {
18             System.out.print("Masukkan nama baru: ");
19             produk.setNama(scanner.nextLine());
20             System.out.print("Masukkan harga baru: ");
21             produk.setHarga(scanner.nextDouble());
22             scanner.nextLine();
23             System.out.println("Produk berhasil diperbarui!");
24             return;
25         }
26     }
27     System.out.println("Produk dengan ID tersebut tidak ditemukan.");
28 }
29
30 private static void hapusProduk() {
31     System.out.print("Masukkan ID produk yang ingin dihapus: ");
32     int id = scanner.nextInt();
33     scanner.nextLine();
34
35     boolean removed = daftarProduk.removeIf(produk -> produk.getId() == id);
36     if (removed) {
37         System.out.println("Produk berhasil dihapus!");
38     } else {
39         System.out.println("Produk tidak ditemukan.");
40     }
41 }
42
43 private static void tampilkanDeskripsiProduk() {
44     if (daftarProduk.isEmpty()) {
45         System.out.println("Tidak ada produk untuk ditampilkan.");
46         return;
47     }
48
49     System.out.println("Deskripsi Produk:");
50     for (Produk produk : daftarProduk) {
51         System.out.println(produk.deskripsi());
52     }
53 }
54 }
```

```
1 // Subclass ParfumEdisiTerbatas
2 class ParfumEdisiTerbatas extends Parfum {
3     private String edisi;
4
5     public ParfumEdisiTerbatas(int id, String nama, double harga, String aroma, String edisi) {
6         super(id, nama, harga, aroma);
7         this.edisi = (edisi.isEmpty()) ? "Tidak Diketahui" : edisi;
8     }
9
10    public String getEdisi() {
11        return edisi;
12    }
13
14    public void setEdisi(String edisi) {
15        if (edisi.isEmpty()) {
16            throw new IllegalArgumentException("Edisi tidak boleh kosong!");
17        }
18        this.edisi = edisi;
19    }
20
21    @Override
22    public String deskripsi() {
23        return "Parfum Edisi Terbatas: " + getNama() + " | Aroma: " + getAroma() + " | Edisi: " + getEdisi();
24    }
25
26    @Override
27    public String toString() {
28        return super.toString() + ", Edisi: " + edisi;
29    }
30 }
```



```
1  // Subclass Parfum
2  class Parfum extends Produk {
3      private final 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     @Override
15     public String deskripsi() {
16         return "Parfum Biasa: " + getNama() + " | Aroma: " + getAroma();
17     }
18
19     @Override
20     public String toString() {
21         return super.toString() + ", Aroma: " + aroma;
22     }
23 }
24
```