

## Kategori.Java

```
1
2  package struktur;
3
4  public final class Kategori {
5      private final String namaKategori;
6
7
8      public Kategori(String namaKategori) {
9          this.namaKategori = namaKategori;
10     }
11
12
13     public String getNamaKategori() {
14         return namaKategori;
15     }
16
17     @Override
18     public String toString() {
19         return namaKategori;
20     }
21 }
22
```

## DeskripsiProduk.Java

```
1  package struktur;
2
3  public interface DeskripsiProduk {
4      void tampilkanDeskripsi();
5      boolean stokMenipis();
6  }
```

## ManajemenProduk.java (1)

```
1 package struktur;
2
3 import java.util.ArrayList;
4 import java.util.InputMismatchException;
5 import java.util.Scanner;
6
7 public class ManajemenProduk {
8     private ArrayList<Produk> daftarProduk = new ArrayList<>();
9     private ArrayList<Kategori> daftarKategori = new ArrayList<>();
10    private Scanner scanner = new Scanner(System.in);
11    private static int totalProduk = 2;
12
13    public static int getTotalProduk() {
14        return totalProduk;
15    }
16
17    public ManajemenProduk() {
18        daftarKategori.add(new Kategori("Roti"));
19        daftarKategori.add(new Kategori("Pastry"));
20    }
21
22    public ArrayList<Produk> getDaftarProduk() {
23        return daftarProduk;
24    }
25
26    public ArrayList<Kategori> getDaftarKategori() {
27        return daftarKategori;
28    }
29
30
31    public void tambahProduk() {
32        try {
33            System.out.print("Masukkan nama produk: ");
34            String nama = scanner.nextLine();
35
36            System.out.print("Masukkan harga produk: ");
37            double harga = scanner.nextDouble();
38            scanner.nextLine();
39
40            System.out.print("Masukkan stok produk: ");
41            int stok = scanner.nextInt();
42            scanner.nextLine();
43
44            System.out.println("Pilih kategori:");
45            for (int i = 0; i < daftarKategori.size(); i++) {
46                System.out.println((i + 1) + ". " + daftarKategori.get(i).getNamaKategori());
47            }
48            System.out.print("Masukkan nomor kategori: ");
49            int kategoriIndex = scanner.nextInt() - 1;
50            scanner.nextLine();
51
52            if (kategoriIndex >= 0 && kategoriIndex < daftarKategori.size()) {
53                Kategori kategoriDipilih = daftarKategori.get(kategoriIndex);
54                Produk produkBaru;
55
56                if (kategoriDipilih.getNamaKategori().equalsIgnoreCase("Roti")) {
57                    System.out.print("Masukkan rasa roti: ");
58                    String rasa = scanner.nextLine();
59                    produkBaru = new ProdukRoti(nama, harga, stok, kategoriDipilih, rasa);
60                } else if (kategoriDipilih.getNamaKategori().equalsIgnoreCase("Pastry")) {
61                    System.out.println("Pilih ukuran pastry:");
62                    System.out.println("1. Mini");
63                    System.out.println("2. Sedang");
64                    System.out.println("3. Jumbo");
65                    System.out.print("Masukkan pilihan (1-3): ");
66                    int ukuran = scanner.nextInt();
67                    scanner.nextLine();
68
69                    if (ukuran < 1 || ukuran > 3) {
70                        System.out.println("Ukuran tidak valid! Default ke Sedang (2)");
71                        ukuran = 2;
72                    }
73
74                    produkBaru = new ProdukPastry(nama, harga, stok, kategoriDipilih, ukuran);
75                } else {
76                    System.out.println("Kategori tidak dikenali! Produk tidak ditambahkan.");
77                    return;
78                }
79
80                daftarProduk.add(produkBaru);
81                System.out.println("Produk berhasil ditambahkan!");
82                totalProduk++;
83            } else {
84                System.out.println("Kategori tidak valid!");
85            }
86        } catch (InputMismatchException e) {
87            System.out.println("Input tidak valid. Pastikan memasukkan angka untuk harga, stok, dan pilihan.");
88            scanner.nextLine();
89        }
90    }
91}
```

## ManajemenProduk.java (2)

```
91 // Overloading 1: Tambah produk roti dengan rasa (String)
92 public void tambahProduk(String nama, double harga, int stok, Kategori kategori, String rasa) {
93     if (!kategori.getNamaKategori().equalsIgnoreCase("Roti")) {
94         System.out.println("Error: Kategori harus 'Roti'");
95         return;
96     }
97     daftarProduk.add(new ProdukRoti(nama, harga, stok, kategori, rasa));
98     System.out.println("Produk roti berhasil ditambahkan!");
99 }
100
101 // Overloading 2: Tambah produk pastry dengan ukuran (int)
102 public void tambahProduk(String nama, double harga, int stok, Kategori kategori, int ukuran) {
103     if (!kategori.getNamaKategori().equalsIgnoreCase("Pastry")) {
104         System.out.println("Error: Kategori harus 'Pastry'");
105         return;
106     }
107     if (ukuran < 1 || ukuran > 3) {
108         System.out.println("Error: Ukuran harus 1 (Mini), 2 (Sedang), atau 3 (Jumbo)");
109         return;
110     }
111     daftarProduk.add(new ProdukPastry(nama, harga, stok, kategori, ukuran));
112     System.out.println("Produk pastry berhasil ditambahkan!");
113 }
114
115 public void tampilkanProduk() {
116     if (daftarProduk.isEmpty()) {
117         System.out.println("Belum ada produk!");
118     } else {
119         System.out.println("\n=== Daftar Produk ===");
120         for (int i = 0; i < daftarProduk.size(); i++) {
121             Produk p = daftarProduk.get(i);
122             System.out.print((i + 1) + ". " + p);
123
124             if (p instanceof DeskripsiProduk d && d.stokMenipis()) {
125                 System.out.print(" [Stok Menipis]");
126             }
127
128             System.out.println();
129         }
130         System.out.println("Total produk terdaftar: " + totalProduk);
131     }
132 }
133
134 public void editProduk() {
135     tampilkanProduk();
136     if (daftarProduk.isEmpty()) return;
137
138     try {
139         System.out.print("Pilih nomor produk yang ingin diedit: ");
140         int index = scanner.nextInt() - 1;
141         scanner.nextLine();
142
143         if (index >= 0 && index < daftarProduk.size()) {
144             Produk produk = daftarProduk.get(index);
145
146             System.out.print("Masukkan nama baru: ");
147             produk.setNama(scanner.nextLine());
148
149             System.out.print("Masukkan harga baru: ");
150             produk.setHarga(scanner.nextDouble());
151             scanner.nextLine();
152
153             System.out.print("Masukkan stok baru: ");
154             produk.setStok(scanner.nextInt());
155             scanner.nextLine();
156
157             System.out.println("Pilih kategori baru:");
158             for (int i = 0; i < daftarKategori.size(); i++) {
159                 System.out.print((i + 1) + ". " + daftarKategori.get(i).getNamaKategori());
160             }
161             System.out.print("Masukkan nomor kategori: ");
162             int kategoriIndex = scanner.nextInt() - 1;
163             scanner.nextLine();
164
165             if (kategoriIndex >= 0 && kategoriIndex < daftarKategori.size()) {
166                 produk.setKategori(daftarKategori.get(kategoriIndex));
167             }
168
169             System.out.println("Produk berhasil diperbarui!");
170
171             if (produk instanceof DeskripsiProduk deskriptif && deskriptif.stokMenipis()) {
172                 System.out.println("Peringatan: Stok produk ini menipis!");
173             }
174         } else {
175             System.out.println("Nomor produk tidak valid!");
176         }
177     } catch (InputMismatchException e) {
178         System.out.println("Input tidak valid. Pastikan angka dimasukkan dengan benar.");
179         scanner.nextLine();
180     }
181 }
182
183 }
```

## ManajemenProduk.java (3)

```
188
189     public void hapusProduk() {
190         tampilkanProduk();
191         if (daftarProduk.isEmpty()) return;
192
193         try {
194             System.out.print("Pilih nomor produk yang ingin dihapus: ");
195             int index = scanner.nextInt() - 1;
196             scanner.nextLine();
197
198             if (index >= 0 && index < daftarProduk.size()) {
199                 daftarProduk.remove(index);
200                 totalProduk--;
201                 System.out.println("Produk berhasil dihapus!");
202             } else {
203                 System.out.println("Nomor produk tidak valid!");
204             }
205         } catch (InputMismatchException e) {
206             System.out.println("Input tidak valid. Harus berupa angka.");
207             scanner.nextLine();
208         }
209     }
210
211     public void tampilkanDetailProduk() {
212         if (daftarProduk.isEmpty()) {
213             System.out.println("Belum ada produk.");
214             return;
215         }
216
217         tampilkanProduk();
218
219         try {
220             System.out.print("Pilih nomor produk untuk melihat detail: ");
221             int index = scanner.nextInt() - 1;
222             scanner.nextLine();
223
224             if (index >= 0 && index < daftarProduk.size()) {
225                 Produk produk = daftarProduk.get(index);
226                 System.out.println("\n--- Detail Produk ---");
227                 System.out.println(produk);
228
229                 if (produk instanceof DeskripsiProduk deskripsi) {
230                     deskripsi.tampilkanDeskripsi();
231                 }
232             } else {
233                 System.out.println("Nomor tidak valid!");
234             }
235         } catch (InputMismatchException e) {
236             System.out.println("Input harus berupa angka.");
237             scanner.nextLine();
238         }
239     }
240 }
```

## Produk.Java

```
1 package struktur;
2
3 public abstract class Produk {
4     private String nama;
5     private double harga;
6     private int stok;
7     private Kategori kategori;
8
9     public Produk(String nama, double harga, int stok, Kategori kategori) {
10         this.nama = nama;
11         this.harga = harga;
12         this.stok = stok;
13         this.kategori = kategori;
14     }
15
16     public String getNama() {
17         return nama;
18     }
19
20     public void setNama(String nama) {
21         this.nama = nama;
22     }
23
24     public double getHarga() {
25         return harga;
26     }
27
28     public void setHarga(double harga) {
29         this.harga = harga;
30     }
31
32     public int getStok() {
33         return stok;
34     }
35
36     public void setStok(int stok) {
37         this.stok = stok;
38     }
39
40     public Kategori getKategori() {
41         return kategori;
42     }
43
44     public void setKategori(Kategori kategori) {
45         this.kategori = kategori;
46     }
47
48     public abstract String getJenisProduk(); //abstract method
49
50     public String getDetailTambahan() {
51         return "";
52     }
53
54     @Override
55     public String toString() {
56         return nama + " - Rp" + harga + " - Stok: " + stok + " - Kategori: " + kategori.getNamaKategori() + getDetailTambahan();
57     }
58 }
59
```

## ProdukPastry.java

```
1 package struktur;
2
3 public class ProdukPastry extends Produk implements DeskripsiProduk {
4     private int ukuran;
5
6     public ProdukPastry(String nama, double harga, int stok, Kategori kategori, int ukuran) {
7         super(nama, harga, stok, kategori);
8         this.ukuran = ukuran;
9     }
10
11     public int getUkuran() {
12         return ukuran;
13     }
14
15     public String getUkuranString() {
16         switch (ukuran) {
17             case 1: return "Mini";
18             case 2: return "Sedang";
19             case 3: return "Jumbo";
20             default: return "Tidak Valid";
21         }
22     }
23
24     @Override
25     public String getDetailTambahan() {
26         return " - Ukuran: " + getUkuranString();
27     }
28
29     @Override
30     public String getJenisProduk() {
31         return "Pastry";
32     }
33
34     @Override
35     public void tampilkanDeskripsi() {
36         System.out.println(getNama() + " adalah pastry ukuran " + getUkuranString() + ", cocok untuk acara spesial.");
37     }
38
39     @Override
40     public boolean stokMenipis() {
41         return getStok() <= 3;
42     }
43
44 }
```

## ProdukRoti.java

```
1  package struktur;
2
3  public class ProdukRoti extends Produk implements DeskripsiProduk {
4      private String rasa;
5
6      public ProdukRoti(String nama, double harga, int stok, Kategori kategori, String rasa) {
7          super(nama, harga, stok, kategori);
8          this.rasa = rasa;
9      }
10
11     public String getRasa() {
12         return rasa;
13     }
14
15     public void setRasa(String rasa) {
16         this.rasa = rasa;
17     }
18
19     @Override
20     public String getDetailTambahan() {
21         return " - Rasa: " + rasa;
22     }
23
24     @Override
25     public String getJenisProduk() {
26         return "Roti";
27     }
28
29     @Override
30     public void tampilkanDeskripsi() {
31         System.out.println(getNama() + " adalah roti rasa " + rasa + ", cocok untuk sarapan dan camilan.");
32     }
33
34     @Override
35     public boolean stokMenipis() {
36         return getStok() <= 5;
37     }
38
39 }
40
```

## Pelanggan.java

```
1 package struktur;
2
3 import java.util.InputMismatchException;
4 import java.util.Scanner;
5
6 public class Pelanggan {
7     private ManajemenProduk manajemenProduk;
8     private Scanner scanner;
9
10    public Pelanggan(ManajemenProduk manajemenProduk) {
11        this.manajemenProduk = manajemenProduk;
12        this.scanner = new Scanner(System.in);
13    }
14
15    public void beliProduk() {
16        try {
17            manajemenProduk.tampilkanProduk();
18            if (manajemenProduk.getDaftarProduk().isEmpty()) return;
19
20            System.out.print("Pilih nomor produk yang ingin dibeli: ");
21            int index = getValidIntInput(scanner) - 1;
22            scanner.nextLine();
23
24            if (index >= 0 && index < manajemenProduk.getDaftarProduk().size()) {
25                Produk produk = manajemenProduk.getDaftarProduk().get(index);
26
27                System.out.print("Masukkan jumlah yang ingin dibeli: ");
28                int jumlah = getValidIntInput(scanner);
29                scanner.nextLine();
30
31                if (jumlah <= 0) {
32                    System.out.println("Jumlah harus lebih dari 0!");
33                    return;
34                }
35
36                if (produk.getStok() >= jumlah) {
37                    double totalHarga = produk.getHarga() * jumlah;
38                    System.out.printf("Total harga: Rp%,.2f\n", totalHarga);
39
40                    System.out.print("Masukkan uang pembayaran: Rp");
41                    double pembayaran = getValidDoubleInput(scanner);
42                    scanner.nextLine();
43
44                    if (pembayaran >= totalHarga) {
45                        produk.setStok(produk.getStok() - jumlah);
46                        System.out.println("Pembelian berhasil!");
47                        if (pembayaran > totalHarga) {
48                            System.out.printf("Kembalian: Rp%,.2f\n", (pembayaran - totalHarga));
49                        }
50                    } else {
51                        System.out.println("Uang tidak cukup! Pembelian dibatalkan.");
52                    }
53                } else {
54                    System.out.println("Stok tidak mencukupi! Stok tersedia: " + produk.getStok());
55                }
56            } else {
57                System.out.println("Nomor produk tidak valid!");
58            }
59        } catch (Exception e) {
60            System.out.println("Terjadi kesalahan saat proses pembelian: " + e.getMessage());
61        }
62    }
63
64    private int getValidIntInput(Scanner scanner) {
65        while (true) {
66            try {
67                return scanner.nextInt();
68            } catch (InputMismatchException e) {
69                System.out.print("Input harus berupa angka. Silakan coba lagi: ");
70                scanner.nextLine();
71            }
72        }
73    }
74
75    private double getValidDoubleInput(Scanner scanner) {
76        while (true) {
77            try {
78                return scanner.nextDouble();
79            } catch (InputMismatchException e) {
80                System.out.print("Input harus berupa angka. Silakan coba lagi: ");
81                scanner.nextLine();
82            }
83        }
84    }
85 }
```



## LunarBakeshop.java

```
1 import java.util.InputMismatchException;
2 import java.util.Scanner;
3 import struktur.Kategori;
4 import struktur.ManajemenProduk;
5 import struktur.Pelanggan;
6
7 public class LunarBakeshop {
8     public static void main(String[] args) {
9         ManajemenProduk manajemenProduk = new ManajemenProduk();
10        Pelanggan pelanggan = new Pelanggan(manajemenProduk);
11        Scanner scanner = new Scanner(System.in);
12
13        try {
14            Kategori roti = manajemenProduk.getDaftarKategori().get(0);
15            Kategori pastry = manajemenProduk.getDaftarKategori().get(1);
16
17            // Tambah produk roti (overloading 1)
18            manajemenProduk.tambahProduk("Roti Coklat", 15000, 30, roti, "Coklat");
19            // Tambah produk pastry (overloading 2)
20            manajemenProduk.tambahProduk("Croissant", 25000, 20, pastry, 3);
21
22            while (true) {
23                try {
24                    System.out.println("\n== Lunar Bakeshop ==");
25                    System.out.println("1. Menu Admin");
26                    System.out.println("2. Menu Pelanggan");
27                    System.out.println("3. Keluar");
28                    System.out.print("Pilih menu utama: ");
29
30                    int mainMenu = getValidIntInput(scanner);
31
32                    switch (mainMenu) {
33                        case 1:
34                            menuAdmin(manajemenProduk, scanner);
35                            break;
36                        case 2:
37                            menuPelanggan(pelanggan, scanner);
38                            break;
39                        case 3:
40                            System.out.println("Terima kasih telah menggunakan sistem Lunar Bakeshop!");
41                            return;
42                        default:
43                            System.out.println("Pilihan tidak valid! Silakan pilih 1-3.");
44                    }
45                } catch (InputMismatchException e) {
46                    System.out.println("Input tidak valid! Harap masukkan angka.");
47                    scanner.nextLine();
48                } catch (Exception e) {
49                    System.out.println("Terjadi kesalahan: " + e.getMessage());
50                }
51            }
52        } catch (Exception e) {
53            System.out.println("Terjadi kesalahan sistem: " + e.getMessage());
54        } finally {
55            scanner.close();
56        }
57    }
58
59    private static void menuAdmin(ManajemenProduk manajemenProduk, Scanner scanner) {
60        while (true) {
61            try {
62                System.out.println("\n== Menu Admin ==");
63                System.out.println("1. Tambah Produk");
64                System.out.println("2. Tampilkan Produk");
65                System.out.println("3. Edit Produk");
66                System.out.println("4. Hapus Produk");
67                System.out.println("5. Lihat Detail Produk");
68                System.out.println("6. Kembali ke Menu Utama");
69                System.out.print("Pilih menu: ");
70
71                int pilihan = getValidIntInput(scanner);
72
73                switch (pilihan) {
74                    case 1:
75                        manajemenProduk.tambahProduk();
76                        break;
77                    case 2:
78                        manajemenProduk.tampilkanProduk();
79                        break;
80                    case 3:
81                        manajemenProduk.editProduk();
82                        break;
83                    case 4:
84                        manajemenProduk.hapusProduk();
85                        break;
86                    case 5:
87                        manajemenProduk.tampilkanDetailProduk();
88                        break;
89                    case 6:
90                        return;
91                    default:
92                        System.out.println("Pilihan tidak valid! Silakan pilih 1-6.");
93                }
94            } catch (InputMismatchException e) {
95                System.out.println("Input tidak valid! Harap masukkan angka.");
96                scanner.nextLine();
97            } catch (Exception e) {
98                System.out.println("Terjadi kesalahan: " + e.getMessage());
99            }
100        }
101    }
102}
```

## LunarBakeshop.Java (2)

```
102
103     private static void menuPelanggan(Pelanggan pelanggan, Scanner scanner) {
104         while (true) {
105             try {
106                 System.out.println("\n=== Menu Pelanggan ===");
107                 System.out.println("1. Beli Produk");
108                 System.out.println("2. Kembali ke Menu Utama");
109                 System.out.print("Pilih menu: ");
110
111                 int pilihan = getValidIntInput(scanner);
112
113                 switch (pilihan) {
114                     case 1:
115                         pelanggan.beliProduk();
116                         break;
117                     case 2:
118                         return;
119                     default:
120                         System.out.println("Pilihan tidak valid! Silakan pilih 1-2.");
121                 }
122             } catch (InputMismatchException e) {
123                 System.out.println("Input tidak valid! Harap masukkan angka.");
124                 scanner.nextLine();
125             } catch (Exception e) {
126                 System.out.println("Terjadi kesalahan: " + e.getMessage());
127             }
128         }
129     }
130
131     private static int getValidIntInput(Scanner scanner) {
132         while (true) {
133             try {
134                 return scanner.nextInt();
135             } catch (InputMismatchException e) {
136                 System.out.print("Input harus berupa angka. Silakan coba lagi: ");
137                 scanner.nextLine();
138             }
139         }
140     }
141 }
```