

## POSTTEST 6

### PEMROGRAMAN BERORIENTASI OBJEK

Nama : Salom sefanya onibala

NIM : 2209106084

Kelas : Informatika B2

#### SS PROGRAM

##### 1. RotiInterface.java

```
src > Roti > RotiInterface.java > ...
1  package Roti;
2
3  public interface RotiInterface {
4      String getTipeRoti();
5      int getId();
6      String getName();
7      int getHarga();
8      int getStok();
9      void setName(String nama);
10     void setHarga(int harga);
11     void setStok(int stok);
12 }
```

##### 2. roti.java

```
1  package Roti;
2
3  public abstract class roti implements RotiInterface {
4      protected int id;
5      protected String nama;
6      protected int harga;
7      protected int stok;
8      static int idCounter = 1; //variabel static untuk menghitung ID
9
10     public roti(String nama, int harga, int stok) {
11         this.id = idCounter++;
12         this.nama = nama;
13         this.harga = harga;
14         this.stok = stok;
15     }
16
17     // Implementasi dari interface RotiInterface
18     @Override
19     public int getId() {
20         return id;
21     }
22
23     @Override
24     public String getName() {
25         return nama;
26     }
27
28     @Override
29     public int getHarga() {
30         return harga;
31     }
32
33     @Override
34     public int getStok() {
35         return stok;
36     }
37
38     @Override
39     public void setName(String nama) {
40         this.nama = nama;
41     }
42
43     @Override
44     public void setHarga(int harga) {
45         this.harga = harga;
46     }
47
48     @Override
49     public void setStok(int stok) {
50         this.stok = stok;
51     }
52 }
```

### 3. RotiManis.java

```
1 package Roti;
2
3 public class RotiManis extends roti {
4     protected String rasa;
5
6     public RotiManis(String nama, int harga, int stok, String rasa) {
7         super(nama, harga, stok);
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 getTipeRoti() {
21         return "Manis";
22     }
23
24     @Override
25     public String toString() {
26         return super.toString() + " - Rasa: " + rasa;
27     }
28 }
```

### 4. RotiTawar.java

```
1 package Roti;
2
3 public class RotiTawar extends roti {
4     protected String jenis;
5
6     public RotiTawar(String nama, int harga, int stok, String jenis) {
7         super(nama, harga, stok);
8         this.jenis = jenis;
9     }
10
11     public String getJenis() {
12         return jenis;
13     }
14
15     public void setJenis(String jenis) {
16         this.jenis = jenis;
17     }
18
19     @Override
20     public String getTipeRoti() {
21         return "Tawar";
22     }
23
24     @Override
25     public String toString() {
26         return super.toString() + " - Jenis: " + jenis;
27     }
28 }
```

### 5. App.java

```
1 import java.util.ArrayList;
2 import java.util.InputMismatchException;
3 import java.util.NoSuchElementException;
4 import java.util.Scanner;
5 import java.io.IOException;
6 import Roti.*;
7
8 public final class App {
9     public static final int idCounter = 1;
10     private static Scanner scanner = new Scanner(System.in);
11     private static ArrayList<roti> daftarRoti = new ArrayList<>();
12
13     Run|Debug
14     public static final void main(String[] args) throws IOException {
15         while (true) {
16             clearScreen();
17             System.out.println(x:"=====");
18             System.out.println(x:"          Selamat datang di Toko Roti Onibala          ");
19             System.out.println(x:"=====");
20             System.out.println(x:" [1] Tambah Roti");
21             System.out.println(x:" [2] Lihat Semua Roti");
22             System.out.println(x:" [3] Perbarui Informasi Roti");
23             System.out.println(x:" [4] Hapus Roti");
24             System.out.println(x:" [5] Keluar");
25             System.out.print(s:" Pilih menu: ");
26
27             int pilihan = scanner.nextInt();
28             scanner.nextLine();
29
30             switch (pilihan) {
31                 case 1:
32                     tambahRoti();
33                     waitForEnter();
34                     break;
35                 case 2:
36                     lihatSemuaRoti();
```

```

35         case 2:
36             lihatSemuaRoti();
37             waitForEnter();
38             break;
39         case 3:
40             perbaruiRoti();
41             waitForEnter();
42             break;
43         case 4:
44             hapusRoti();
45             waitForEnter();
46             break;
47         case 5:
48             System.out.println(x:"Terima kasih telah menggunakan layanan Toko Roti Onibala");
49             return;
50         default:
51             System.out.println(x:"Pilihan tidak valid, silakan coba lagi");
52             waitForEnter();
53     }
54 }
55
56
57 private static void tambahRoti() {
58     clearScreen();
59     System.out.println(x:"=====");
60     System.out.println(x:"                Selamat datang di Toko Roti Onibala                ");
61     System.out.println(x:"=====");
62     System.out.print(s:"Masukkan nama roti    : ");
63     String nama = scanner.nextLine();
64     System.out.print(s:"Masukkan harga roti   : ");
65     int harga = scanner.nextInt();
66     System.out.print(s:"Masukkan stok roti    : ");
67     int stok = scanner.nextInt();
68     scanner.nextLine();
69
70     System.out.print(s:"Roti Tawar atau Roti Manis (Tawar/Manis) : ");
71     String jenisRoti = scanner.nextLine();
72
73     if (jenisRoti.equalsIgnoreCase(anotherString:"Tawar")) {
74         System.out.print(s:"Masukkan jenis roti tawar (Misal: Gandum/Roti Sobek): ");
75         String jenis = scanner.nextLine();
76         RotiTawar roti = new RotiTawar(nama, harga, stok, jenis);
77         daftarRoti.add(roti);
78     } else if (jenisRoti.equalsIgnoreCase(anotherString:"Manis")) {
79         System.out.print(s:"Masukkan rasa roti manis (Misal: coklat/Keju): ");
80         String rasa = scanner.nextLine();
81         RotiManis roti = new RotiManis(nama, harga, stok, rasa);
82         daftarRoti.add(roti);
83     } else {
84         System.out.println(x:"Jenis roti tidak valid");
85         return;
86     }
87     System.out.println(x:"Roti berhasil ditambahkan");
88 }
89
90 private static void lihatSemuaRoti() {
91     clearScreen();
92     System.out.println(x:"=====");
93     System.out.println(x:"                Selamat datang di Toko Roti Onibala                ");
94     System.out.println(x:"=====");
95
96     if (daftarRoti.isEmpty()) {
97         System.out.println(x:"Tidak ada roti yang tersedia");
98     } else {
99         System.out.println(x:"                DAFTAR ROTI                ");
100         for (roti roti : App.daftarRoti) {
101             //menampilkan informasi roti
102             System.out.println(roti.getId() + ". " + "Roti " + roti.getNama());
103             System.out.println("    Harga    : Rp. " + roti.getHarga());
104             System.out.println("    Stok    : " + roti.getStok());
105             //memeriksa apakah roti adalah RotiManis atau RotiTawar
106             if (roti instanceof RotiManis) {
107                 RotiManis rotiManis = (RotiManis) roti;
108                 System.out.println("    Rasa    : " + rotiManis.getRasa());
109             } else if (roti instanceof RotiTawar) {
110                 RotiTawar rotiTawar = (RotiTawar) roti;
111                 System.out.println("    Jenis    : " + rotiTawar.getJenis());
112             }
113         }
114     }
115 }
116
117 private static void perbaruiRoti() {
118     clearScreen();
119     System.out.println(x:"=====");
120     System.out.println(x:"                Selamat datang di Toko Roti Onibala                ");
121     System.out.println(x:"=====");
122
123     if (daftarRoti.isEmpty()) {
124         System.out.println(x:"Tidak ada roti yang tersedia untuk diperbarui");
125         return;
126     }
127
128     System.out.print(s:"Masukkan ID roti yang ingin diperbarui: ");
129     int idRoti = scanner.nextInt();
130     scanner.nextLine();
131
132     for (roti roti : daftarRoti) {
133         if (roti.getId() == idRoti) {
134             System.out.print(s:"Masukkan nama baru untuk roti    : ");
135             String newName = scanner.nextLine();
136             System.out.print(s:"Masukkan harga baru untuk roti  : ");
137             int newHarga = scanner.nextInt();
138             System.out.print(s:"Masukkan stok baru untuk roti   : ");

```

```

139         int newStok = scanner.nextInt();
140         scanner.nextLine();
141
142         roti.setNama(newNama);
143         roti.setHarga(newHarga);
144         roti.setStok(newStok);
145         System.out.println(x:"Informasi roti berhasil diperbarui");
146         return;
147     }
148 }
149 System.out.println(x:"Roti dengan ID tersebut tidak ditemukan");
150 }
151
152 private static void hapusRoti() {
153     clearScreen();
154     System.out.println(x:"=====");
155     System.out.println(x:"                Selamat datang di Toko Roti Onibala                ");
156     System.out.println(x:"=====");
157
158     // Periksa apakah daftarRoti kosong
159     if (daftarRoti.isEmpty()) {
160         System.out.println(x:"Tidak ada roti yang tersedia untuk dihapus");
161         return;
162     }
163
164     try {
165
166         System.out.print(s:"Masukkan ID roti yang ingin dihapus: ");
167         int idRoti = scanner.nextInt();
168         scanner.nextLine();
169
170         //mencari roti berdasarkan ID
171         roti rotiToDelete = null;
172         for (roti roti : daftarRoti) {
173             if (roti.getId() == idRoti) {
174                 rotiToDelete = roti;
175                 break;
176             }
177         }
178
179         if (rotiToDelete != null) {
180             // cek apakah stok roti lebih dari 1
181             if (rotiToDelete.getStok() > 1) {
182                 // Jika stok lebih dari 1, kurangi stok roti
183                 rotiToDelete.setStok(rotiToDelete.getStok() - 1);
184                 System.out.println(x:"Stok roti berhasil dikurangi");
185             } else {
186                 // Jika stok hanya 1, hapus roti dari daftar
187                 daftarRoti.remove(rotiToDelete);
188                 System.out.println(x:"Roti berhasil dihapus");
189             }
190         } else {
191             // Jika roti dengan ID tersebut tidak ditemukan
192             System.out.println("Error: Roti dengan ID " + idRoti + " tidak ditemukan");
193         }
194     } catch (InputMismatchException e) {
195         //kesalahan jika input ID roti bukan bilangan bulat
196         System.out.println(x:"Error: Masukkan ID roti dalam bentuk bilangan bulat.");
197     } catch (NoSuchElementException e) {
198         //kesalahan jika tidak ada input tersedia
199         System.out.println(x:"Error: Tidak ada input tersedia.");
200     } catch (Exception e) {
201         //kesalahan umum lainnya
202         System.out.println(x:"Error: Terjadi kesalahan saat menghapus roti.");
203         e.printStackTrace();
204     }
205 }
206
207
208
209 //metode static untuk membersihkan layar
210 private static void clearScreen() {
211     System.out.print(s:"\033[H\033[2J");
212     System.out.flush();
213 }
214
215
216 private static void waitForEnter() {
217     System.out.print(s:"Tekan Enter untuk melanjutkan...");
218     try {
219         System.in.read();
220     } catch (Exception e) {
221         e.printStackTrace();
222     }
223 }
224 }

```