

Nama : Nelsi

Kelas : C'23

Nim : 2309106120

Posttest 5

```
1  import java.util.*;
2
3
4  abstract class Barang {
5      protected String id;
6      protected String nama;
7
8      public Barang(String id, String nama) {
9          this.id = id;
10         this.nama = nama;
11     }
12
13     public String getId() {
14         return id;
15     }
16
17     public String getName() {
18         return nama;
19     }
20
21     public void setName(String nama) {
22         this.nama = nama;
23     }
24
25     public abstract void infoBarang();
26 }
27
28
29 class BahanBaku extends Barang {
30     private int stok;
31     private double harga;
32
33     public BahanBaku(String id, String nama, int stok, double harga) {
34         super(id, nama);
35         this.stok = Math.max(stok, 0);
36         this.harga = Math.max(harga, 0);
37     }
38
39     public int getStok() {
40         return stok;
41     }
42
43     public double getHarga() {
44         return harga;
45     }
46
47     public void setStok(int stok) {
48         this.stok = Math.max(stok, 0);
49     }
50
51     public void setHarga(double harga) {
52         this.harga = Math.max(harga, 0);
53     }
54
55     @Override
56     public String toString() {
57         return "ID: " + id + ", Nama: " + nama + ", Stok: " + stok + ", Harga: " + harga;
58     }
59
60     @Override
61     public void infoBarang() {
62         System.out.println("Bahan Baku: " + nama + " (ID: " + id + ")");
63     }
64 }
65
66
67 abstract class Orang {
68     protected String nama;
69     protected String kontak;
70
71     public Orang(String nama, String kontak) {
72         this.nama = nama;
73         this.kontak = kontak;
74     }
75
76     public String getName() {
77         return nama;
78     }
79
80     public String getKontak() {
81         return kontak;
82     }
```

```

84     public void setKontak(String kontak) {
85         this.kontak = kontak;
86     }
87
88     public void setName(String nama) {
89         this.nama = nama;
90     }
91
92
93     public abstract void tampilkanInfo();
94 }
95
96 class Supplier extends Orang {
97     private String idSupplier;
98
99     public Supplier(String idSupplier, String nama, String kontak) {
100         super(nama, kontak);
101         this.idSupplier = idSupplier;
102     }
103
104     public String getIdSupplier() {
105         return idSupplier;
106     }
107
108     @Override
109     public String toString() {
110         return "ID Supplier: " + idSupplier + ", Nama: " + nama + ", Kontak: " + kontak;
111     }
112
113     @Override
114     public void tampilkanInfo() {
115         System.out.println("Supplier: " + nama + " (ID: " + idSupplier + ")");
116     }
117 }
118
119
120 final class Transaksi {
121     private String idTransaksi;
122     private String idBahan;
123     private int jumlah;
124     private double totalHarga;

```

```

126     public Transaksi(String idTransaksi, String idBahan, int jumlah, double totalHarga) {
127         this.idTransaksi = idTransaksi;
128         this.idBahan = idBahan;
129         this.jumlah = jumlah;
130         this.totalHarga = totalHarga;
131     }
132
133     @Override
134     public String toString() {
135         return "ID Transaksi: " + idTransaksi + ", ID Bahan: " + idBahan + ", Jumlah: " + jumlah + ", Total Harga: " + totalHarga;
136     }
137 }
138
139
140 public class SistemWarung {
141     private static final ArrayList<BahanBaku> daftarBahan = new ArrayList<>();
142     private static final ArrayList<Supplier> daftarSupplier = new ArrayList<>();
143     private static final ArrayList<Transaksi> daftarTransaksi = new ArrayList<>();
144     private static final Scanner scanner = new Scanner(System.in); // final variable
145
146     Run | Debug
147     public static void main(String[] args) {
148         int pilihan;
149         do {
150             System.out.println("\n== Sistem Pengelolaan Warung Sembako ==");
151             System.out.println("1. Tambah Bahan Baku");
152             System.out.println("2. Hapus Bahan Baku");
153             System.out.println("3. Edit Bahan Baku");
154             System.out.println("4. Tambah Supplier");
155             System.out.println("5. Hapus Supplier");
156             System.out.println("6. Edit Supplier");
157             System.out.println("7. Tambah Transaksi");
158             System.out.println("8. Tampilkan Semua Data");
159             System.out.println("9. Cari Supplier");
160             System.out.println("10. Cari Transaksi");
161             System.out.println("11. Urutkan Bahan Baku");
162             System.out.println("12. Keluar");
163             System.out.print("Pilih menu: ");
164             pilihan = scanner.nextInt();
165             scanner.nextLine();

```

```

166         switch (pilihan) {
167             case 1 -> tambahBahan();
168             case 2 -> hapusBahan();
169             case 3 -> editBahan();
170             case 4 -> tambahSupplier();
171             case 5 -> hapusSupplier();
172             case 6 -> editSupplier();
173             case 7 -> tambahTransaksi();
174             case 8 -> tampilkanSemuaData(); // final method
175             case 9 -> cariSupplier();
176             case 10 -> cariTransaksi();
177             case 11 -> urutkanBahan();
178             case 12 -> System.out.println(x:"Keluar dari program...");
179             default -> System.out.println(x:"Pilihan tidak valid!");
180         }
181     } while (pilihan != 12);
182 }
183
184 private static void tambahBahan() {
185     System.out.print(s:"Masukkan ID Bahan: ");
186     String id = scanner.nextLine();
187     System.out.print(s:"Masukkan Nama Bahan: ");
188     String nama = scanner.nextLine();
189     System.out.print(s:"Masukkan Stok: ");
190     int stok = scanner.nextInt();
191     System.out.print(s:"Masukkan Harga: ");
192     double harga = scanner.nextDouble();
193     scanner.nextLine();
194     daftarBahan.add(new BahanBaku(id, nama, stok, harga));
195     System.out.println(x:"Bahan Baku Berhasil Ditambahkan!");
196 }
197
198 private static void hapusBahan() {
199     System.out.print(s:"Masukkan ID Bahan yang ingin dihapus: ");
200     String id = scanner.nextLine();
201     daftarBahan.removeIf(bahan -> bahan.getId().equals(id));
202     System.out.println(x:"Bahan berhasil dihapus.");
203 }

```

```

205 private static void editBahan() {
206     System.out.print(s:"Masukkan ID Bahan yang ingin diedit: ");
207     String id = scanner.nextLine();
208     for (BahanBaku bahan : daftarBahan) {
209         if (bahan.getId().equals(id)) {
210             System.out.print(s:"Masukkan Nama Baru: ");
211             String namaBaru = scanner.nextLine();
212             System.out.print(s:"Masukkan Stok Baru: ");
213             int stokBaru = scanner.nextInt();
214             System.out.print(s:"Masukkan Harga Baru: ");
215             double hargaBaru = scanner.nextDouble();
216             scanner.nextLine();
217             bahan.setNama(namaBaru);
218             bahan.setStok(stokBaru);
219             bahan.setHarga(hargaBaru);
220             System.out.println(x:"Data Bahan Berhasil Diubah!");
221             return;
222         }
223     }
224     System.out.println(x:"Bahan tidak ditemukan.");
225 }
226
227 private static void tambahSupplier() {
228     System.out.print(s:"Masukkan ID Supplier: ");
229     String id = scanner.nextLine();
230     System.out.print(s:"Masukkan Nama Supplier: ");
231     String nama = scanner.nextLine();
232     System.out.print(s:"Masukkan Kontak: ");
233     String kontak = scanner.nextLine();
234     daftarSupplier.add(new Supplier(id, nama, kontak));
235     System.out.println(x:"Supplier Berhasil Ditambahkan!");
236 }
237
238 private static void hapusSupplier() {
239     System.out.print(s:"Masukkan ID Supplier yang ingin dihapus: ");
240     String id = scanner.nextLine();
241     daftarSupplier.removeIf(supplier -> supplier.getIdSupplier().equals(id));
242     System.out.println(x:"Supplier berhasil dihapus.");
243 }
244

```

```

245     private static void editSupplier() {
246         System.out.print(s:"Masukkan ID Supplier yang ingin diedit: ");
247         String id = scanner.nextLine();
248         for (Supplier supplier : daftarSupplier) {
249             if (supplier.getIdSupplier().equals(id)) {
250                 System.out.print(s:"Masukkan Nama Baru: ");
251                 String namaBaru = scanner.nextLine();
252                 System.out.print(s:"Masukkan Kontak Baru: ");
253                 String kontakBaru = scanner.nextLine();
254                 supplier.setNama(namaBaru);
255                 supplier.setKontak(kontakBaru);
256                 System.out.println(x:"Data Supplier Berhasil Diubah!");
257                 return;
258             }
259         }
260         System.out.println(x:"Supplier tidak ditemukan.");
261     }
262
263     private static void tambahTransaksi() {
264         System.out.print(s:"Masukkan ID Transaksi: ");
265         String idTransaksi = scanner.nextLine();
266         System.out.print(s:"Masukkan ID Bahan: ");
267         String idBahan = scanner.nextLine();
268         System.out.print(s:"Masukkan Jumlah: ");
269         int jumlah = scanner.nextInt();
270         scanner.nextLine();
271
272         for (BahanBaku bahan : daftarBahan) {
273             if (bahan.getId().equals(idBahan)) {
274                 double total = bahan.getHarga() * jumlah;
275                 daftarTransaksi.add(new Transaksi(idTransaksi, idBahan, jumlah, total));
276                 System.out.println(x:"Transaksi Berhasil Ditambahkan!");
277                 return;
278             }
279         }
280         System.out.println(x:"ID Bahan tidak ditemukan.");
281     }

```

```

283     private final static void tampilkanSemuaData() {
284         System.out.println(x:"\n== Daftar Bahan Baku ==");
285         daftarBahan.forEach(System.out::println);
286         System.out.println(x:"\n== Daftar Supplier ==");
287         daftarSupplier.forEach(System.out::println);
288         System.out.println(x:"\n== Daftar Transaksi ==");
289         daftarTransaksi.forEach(System.out::println);
290     }
291
292     private static void cariSupplier() {
293         System.out.print(s:"Masukkan ID Supplier: ");
294         String id = scanner.nextLine();
295         daftarSupplier.stream().filter(s -> s.getIdSupplier().equals(id)).forEach(System.out::println);
296     }
297
298     private static void cariTransaksi() {
299         System.out.print(s:"Masukkan ID Transaksi: ");
300         String id = scanner.nextLine();
301         daftarTransaksi.stream().filter(t -> t.toString().contains(id)).forEach(System.out::println);
302     }
303
304     private static void urutkanBahan() {
305         System.out.println(x:"Urutkan berdasarkan: ");
306         System.out.println(x:"1. Nama (A-Z)");
307         System.out.println(x:"2. Nama (Z-A)");
308         System.out.println(x:"3. Harga (Termurah)");
309         System.out.println(x:"4. Harga (Termahal)");
310         int pilihan = scanner.nextInt();
311         scanner.nextLine();
312
313         Comparator<BahanBaku> comparator = switch (pilihan) {
314             case 1 -> Comparator.comparing(BahanBaku::getNama);
315             case 2 -> Comparator.comparing(BahanBaku::getNama).reversed();
316             case 3 -> Comparator.comparingDouble(BahanBaku::getHarga);
317             case 4 -> Comparator.comparingDouble(BahanBaku::getHarga).reversed();
318             default -> null;
319         };
320     }

```

```

321         if (comparator != null) {
322             daftarBahan.sort(comparator);
323             System.out.println(x:"Bahan Baku setelah diurutkan:");
324             daftarBahan.forEach(System.out::println);
325         } else {
326             System.out.println(x:"Pilihan tidak valid.");
327         }
328     }
329 }
330

```