

Nama : Nelsi

Nim : 2309106120

Kelas : C'23

Posttest 6

```
1  import java.util.*;
2
3  abstract class Barang {
4      protected String id;
5      protected String nama;
6
7      public Barang(String id, String nama) {
8          this.id = id;
9          this.nama = nama;
10     }
11
12     public String getId() {
13         return id;
14     }
15
16     public String getName() {
17         return nama;
18     }
19
20     public void setName(String nama) {
21         this.nama = nama;
22     }
23
24     public abstract void infoBarang();
25 }
26
27 class BahanBaku extends Barang {
28     private int stok;
29     private double harga;
30
31     public BahanBaku(String id, String nama, int stok, double harga) {
32         super(id, nama);
33         this.stok = Math.max(stok, 0);
34         this.harga = Math.max(harga, 0);
35     }
36
37     public int getStok() {
38         return stok;
39     }
40
41     public double getHarga() {
42         return harga;
43     }
```

```
45     public void setStok(int stok) {
46         this.stok = Math.max(stok, 0);
47     }
48
49     public void setHarga(double harga) {
50         this.harga = Math.max(harga, 0);
51     }
52
53     @Override
54     public String toString() {
55         return "ID: " + id + ", Nama: " + nama + ", Stok: " + stok + ", Harga: " + harga;
56     }
57
58     @Override
59     public void infoBarang() {
60         System.out.println("Bahan Baku: " + nama + " (ID: " + id + ")");
61     }
62 }
63
64 abstract class Orang {
65     protected String nama;
66     protected String kontak;
67
68     public Orang(String nama, String kontak) {
69         this.nama = nama;
70         this.kontak = kontak;
71     }
72
73     public String getName() {
74         return nama;
75     }
76
77     public String getKontak() {
78         return kontak;
79     }
80
81     public void setName(String nama) {
82         this.nama = nama;
83     }
```

```

85     public void setKontak(String kontak) {
86         this.kontak = kontak;
87     }
88
89     public abstract void tampilkanInfo();
90 }
91
92 class Supplier extends Orang {
93     private String idSupplier;
94
95     public Supplier(String idSupplier, String nama, String kontak) {
96         super(nama, kontak);
97         this.idSupplier = idSupplier;
98     }
99
100     public String getIdSupplier() {
101         return idSupplier;
102     }
103
104     @Override
105     public String toString() {
106         return "ID Supplier: " + idSupplier + ", Nama: " + nama + ", Kontak: " + kontak;
107     }
108
109     @Override
110     public void tampilkanInfo() {
111         System.out.println("Supplier: " + nama + " (ID: " + idSupplier + ")");
112     }
113 }
114
115 final class Transaksi {
116     private String idTransaksi;
117     private String idBahan;
118     private int jumlah;
119     private double totalHarga;
120
121     public Transaksi(String idTransaksi, String idBahan, int jumlah, double totalHarga) {
122         this.idTransaksi = idTransaksi;
123         this.idBahan = idBahan;
124         this.jumlah = jumlah;
125         this.totalHarga = totalHarga;
126     }

```

```

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

```

```

161         switch (pilihan) {
162             case 1 -> tambahBahan();
163             case 2 -> hapusBahan();
164             case 3 -> editBahan();
165             case 4 -> tambahSupplier();
166             case 5 -> hapusSupplier();
167             case 6 -> editSupplier();
168             case 7 -> tambahTransaksi();
169             case 8 -> tampilkanSemuaData();
170             case 9 -> cariSupplier();
171             case 10 -> cariTransaksi();
172             case 11 -> urutkanBahan();
173             case 12 -> System.out.println(x:"Keluar dari program...");
174             default -> System.out.println(x:"Pilihan tidak valid!");
175         }
176     } while (pilihan != 12);
177 }
178
179 private static void tambahBahan() {
180     System.out.print(s:"Masukkan ID Bahan: ");
181     String id = scanner.nextLine();
182     System.out.print(s:"Masukkan Nama Bahan: ");
183     String nama = scanner.nextLine();
184     System.out.print(s:"Masukkan Stok: ");
185     int stok = scanner.nextInt();
186     System.out.print(s:"Masukkan Harga: ");
187     double harga = scanner.nextDouble();
188     scanner.nextLine();
189     daftarBahan.add(new BahanBaku(id, nama, stok, harga));
190     System.out.println(x:"Bahan Baku Berhasil Ditambahkan!");
191 }

```

```

193 private static void hapusBahan() {
194     System.out.print(s:"Masukkan ID Bahan yang ingin dihapus: ");
195     String id = scanner.nextLine();
196     boolean removed = daftarBahan.removeIf(bahan -> bahan.getId().equals(id));
197     if (removed) {
198         System.out.println(x:"Bahan berhasil dihapus.");
199     } else {
200         System.out.println(x:"ID tidak ditemukan.");
201     }
202 }
203
204 private static void editBahan() {
205     System.out.print(s:"Masukkan ID Bahan yang ingin diedit: ");
206     String id = scanner.nextLine();
207     for (BahanBaku bahan : daftarBahan) {
208         if (bahan.getId().equals(id)) {
209             System.out.print(s:"Masukkan Nama Baru: ");
210             String namaBaru = scanner.nextLine();
211             System.out.print(s:"Masukkan Stok Baru: ");
212             int stokBaru = scanner.nextInt();
213             System.out.print(s:"Masukkan Harga Baru: ");
214             double hargaBaru = scanner.nextDouble();
215             scanner.nextLine();
216             bahan.setNama(namaBaru);
217             bahan.setStok(stokBaru);
218             bahan.setHarga(hargaBaru);
219             System.out.println(x:"Data Bahan Berhasil Diubah!");
220             return;
221         }
222     }
223     System.out.println(x:"Bahan tidak ditemukan.");
224 }

```

```

226 private static void tambahSupplier() {
227     System.out.print(s:"Masukkan ID Supplier: ");
228     String id = scanner.nextLine();
229     System.out.print(s:"Masukkan Nama Supplier: ");
230     String nama = scanner.nextLine();
231     System.out.print(s:"Masukkan Kontak: ");
232     String kontak = scanner.nextLine();
233     daftarSupplier.add(new Supplier(id, nama, kontak));
234     System.out.println(x:"Supplier Berhasil Ditambahkan!");
235 }
236
237 private static void hapusSupplier() {
238     System.out.print(s:"Masukkan ID Supplier yang ingin dihapus: ");
239     String id = scanner.nextLine();
240     boolean removed = daftarSupplier.removeIf(supplier -> supplier.getIdSupplier().equals(id));
241     if (removed) {
242         System.out.println(x:"Supplier berhasil dihapus.");
243     } else {
244         System.out.println(x:"ID tidak ditemukan.");
245     }
246 }
247
248 private static void editSupplier() {
249     System.out.print(s:"Masukkan ID Supplier yang ingin diedit: ");
250     String id = scanner.nextLine();
251     for (Supplier supplier : daftarSupplier) {
252         if (supplier.getIdSupplier().equals(id)) {
253             System.out.print(s:"Masukkan Nama Baru: ");
254             String namaBaru = scanner.nextLine();
255             System.out.print(s:"Masukkan Kontak Baru: ");
256             String kontakBaru = scanner.nextLine();
257             supplier.setNama(namaBaru);
258             supplier.setKontak(kontakBaru);
259             System.out.println(x:"Data Supplier Berhasil Diubah!");
260             return;
261         }
262     }
263     System.out.println(x:"Supplier tidak ditemukan.");
264 }

```

```

266 private static void tambahTransaksi() {
267     System.out.print(s:"Masukkan ID Transaksi: ");
268     String idTransaksi = scanner.nextLine();
269     System.out.print(s:"Masukkan ID Bahan: ");
270     String idBahan = scanner.nextLine();
271     System.out.print(s:"Masukkan Jumlah: ");
272     int jumlah = scanner.nextInt();
273     scanner.nextLine();
274
275     for (BahanBaku bahan : daftarBahan) {
276         if (bahan.getId().equals(idBahan)) {
277             double total = bahan.getHarga() * jumlah;
278             daftarTransaksi.add(new Transaksi(idTransaksi, idBahan, jumlah, total));
279             System.out.println(x:"Transaksi Berhasil Ditambahkan!");
280             return;
281         }
282     }
283     System.out.println(x:"ID Bahan tidak ditemukan.");
284 }
285
286 private final static void tampilkanSemuaData() {
287     System.out.println(x:"\n== Daftar Bahan Baku ==");
288     daftarBahan.forEach(System.out::println);
289     System.out.println(x:"\n== Daftar Supplier ==");
290     daftarSupplier.forEach(System.out::println);
291     System.out.println(x:"\n== Daftar Transaksi ==");
292     daftarTransaksi.forEach(System.out::println);
293 }
294
295 private static void cariSupplier() {
296     System.out.print(s:"Masukkan ID Supplier: ");
297     String id = scanner.nextLine();
298     daftarSupplier.stream()
299         .filter(s -> s.getIdSupplier().equals(id))
300         .forEach(System.out::println);
301 }

```

```

303 private static void cariTransaksi() {
304     System.out.print(s:"Masukkan ID Transaksi: ");
305     String id = scanner.nextLine();
306     daftarTransaksi.stream()
307         .filter(t -> t.toString().contains(id))
308         .forEach(System.out::println);
309 }
310
311 private static void urutkanBahan() {
312     System.out.println(x:"Urutkan berdasarkan: ");
313     System.out.println(x:"1. Nama (A-Z)");
314     System.out.println(x:"2. Nama (Z-A)");
315     System.out.println(x:"3. Harga (Termurah)");
316     System.out.println(x:"4. Harga (Termahal)");
317     int pilihan = scanner.nextInt();
318     scanner.nextLine();
319
320     Comparator<BahanBaku> comparator = switch (pilihan) {
321         case 1 -> Comparator.comparing(BahanBaku::getNama);
322         case 2 -> Comparator.comparing(BahanBaku::getNama).reversed();
323         case 3 -> Comparator.comparingDouble(BahanBaku::getHarga);
324         case 4 -> Comparator.comparingDouble(BahanBaku::getHarga).reversed();
325         default -> null;
326     };
327
328     if (comparator != null) {
329         daftarBahan.sort(comparator);
330         System.out.println(x:"Bahan Baku setelah diurutkan:");
331         daftarBahan.forEach(System.out::println);
332     } else {
333         System.out.println(x:"Pilihan tidak valid.");
334     }
335 }
336 }

```