

LAPORAN PRAKTIKUM MINGGU 7

Java Collections Framework dan Implementasi Keranjang Belanja

NAMA : ALVIRA LIBRA RAMADHANI

NIM : 240202851

KELAS : 3IKRA

1. TUJUAN

1. Mahasiswa mampu memahami penggunaan List (ArrayList) dan Map (HashMap) dalam Java.
2. Mahasiswa dapat mengelola kumpulan objek secara dinamis.
3. Mahasiswa mampu mengimplementasikan logika keranjang belanja (tambah, hapus, hitung total) pada sistem Agri-POS.

2. DASAR TEORI

1. Java Collections Framework: Kumpulan interface dan class untuk menyimpan dan memanipulasi data dalam bentuk grup objek.
2. ArrayList: Implementasi List yang menggunakan array dinamis, memungkinkan penyimpanan data yang terurut dan memperbolehkan duplikasi.
3. HashMap: Implementasi Map yang menyimpan data dalam pasangan Key-Value, sangat efisien untuk pencarian data berdasarkan kunci unik.
4. Iterasi: Proses mengakses setiap elemen dalam koleksi menggunakan perulangan (seperti for-each) untuk melakukan operasi tertentu seperti penghitungan total harga.

3. LANGKAH PRAKTIKUM

1. Membuat class Product sebagai entitas data (Blueprint produk).
2. Membuat class ShoppingCart menggunakan ArrayList<Product> untuk penyimpanan sederhana.
3. Membuat class ShoppingCartMap menggunakan Map<Product, Integer> untuk menangani kuantitas produk.
4. Mengimplementasikan method addProduct, removeProduct, dan getTotal pada kedua versi keranjang.
5. Membuat class MainCart untuk menjalankan simulasi transaksi.
6. Melakukan commit ke repositori dengan pesan: week7-collections: implementasi shopping cart

4. KODE PROGRAM

a) Product.java

```
praktikum > week7-koleksi-keranjang > src > main > java > com > upb > agriplos > Product.java > Product
 1 package com.upb.agriplos;
 2
 3 public class Product {
 4     private final String code;
 5     private final String name;
 6     private final double price;
 7
 8     public Product(String code, String name, double price) {
 9         this.code = code;
10         this.name = name;
11         this.price = price;
12     }
13
14     public String getCode() {
15         return code;
16     }
17     public String getName() {
18         return name;
19     }
20
21     public double getPrice() { return price; }
22 }
```

b) ShoppingCart.java

```
praktikum > week7-koleksi-keranjang > src > main > java > com > upb > agriplos > ShoppingCart.java
 1 package com.upb.agriplos;
 2
 3 import java.util.ArrayList;
 4
 5 public class ShoppingCart {
 6     private final ArrayList<Product> items = new ArrayList<>();
 7
 8     public void addProduct(Product p) { items.add(p); }
 9     public void removeProduct(Product p) { items.remove(p); }
10
11     public double getTotal() {
12         double sum = 0;
13         for (Product p : items) {
14             sum += p.getPrice();
15         }
16         return sum;
17     }
18
19     public void printCart() {
20         System.out.println("Isi Keranjang:");
21         for (Product p : items) {
22             System.out.println("- " + p.getCode() + " " + p.getName() + " = " + p.getPrice());
23         }
24         System.out.println("Total: " + getTotal());
25     }
26 }
```

c) ShoppingCartMap.java

```
praktikum > week7-koleksi-keranjang > src > main > java > com > upb > agripos > ShoppingCartMap.java > ...
1 package com.upb.agripos;
2
3 import java.util.HashMap;
4 import java.util.Map;
5
6 public class ShoppingCartMap {
7     private final Map<Product, Integer> items = new HashMap<>();
8
9     public void addProduct(Product p) { items.put(p, items.getOrDefault(p, defaultValue: 0) + 1); }
10
11    public void removeProduct(Product p) {
12        if (!items.containsKey(p)) return;
13        int qty = items.get(p);
14        if (qty > 1) items.put(p, qty - 1);
15        else items.remove(p);
16    }
17
18    public double getTotal() {
19        double total = 0;
20        for (Map.Entry<Product, Integer> entry : items.entrySet()) {
21            total += entry.getKey().getPrice() * entry.getValue();
22        }
23        return total;
24    }
25
26    public void printCart() {
27        System.out.println("Isi Keranjang (Map):");
28        for (Map.Entry<Product, Integer> e : items.entrySet()) {
29            System.out.println("- " + e.getKey().getCode() + " " + e.getKey().getName() + " x" + e.getValue());
30        }
31        System.out.println("Total: " + getTotal());
32    }
33 }
```

d) MainCart.java

```
praktikum > week7-koleksi-keranjang > src > main > java > com > upb > agripos > MainCart.java > Lan
1 package com.upb.agripos;
2
3 public class MainCart {
4     Run main | Debug main | Run | Debug
5     public static void main(String[] args) {
6         System.out.println("Hello, I am [Alvira]-[240202851] (Week7)");
7
8         Product p1 = new Product("P01", "Beras", 50000);
9         Product p2 = new Product("P02", "Pupuk", 30000);
10
11         ShoppingCart cart = new ShoppingCart();
12         cart.addProduct(p1);
13         cart.addProduct(p2);
14         cart.printCart();
15
16         cart.removeProduct(p1);
17         cart.printCart();
18     }
19 }
```

5. HASIL EKSEKUSI

```
PS C:\Users\LENOVO\oop-202501-240202851> & 'C:\Program Files\IBM\SPSS\Statistics\25\JRE\bin\java.exe' '-cp' 'C:\Users\LENOVO\AppData\Roaming\Code\User\workspaceStorage\43d5f3510c604989e781f371f498d2b7\redhat.java\jdt_ws\oop-202501-240202851_edc4669d\bin' 'com.upb.agripos.MainCart'
Hello, I am [Alvira]-[240202851] (Week7)
Isi Keranjang:
- P01 Beras = 50000.0
- P02 Pupuk = 30000.0
Total: 80000.0
Isi Keranjang:
- P02 Pupuk = 30000.0
Total: 30000.0
PS C:\Users\LENOVO\oop-202501-240202851>
```

- **Analisis**

Penggunaan ArrayList sangat mudah untuk operasi penambahan data secara langsung, namun jika ada produk yang sama, ia akan tercatat sebagai baris yang terpisah. Sebaliknya, HashMap lebih efisien dalam merepresentasikan dunia nyata karena mampu mengelompokkan produk yang sama dan menambah jumlah kuantitasnya saja.

- **Kendala dan Solusi**

Saat menggunakan Map, jika tidak teliti pada perulangan Map.Entry, bisa terjadi kesalahan tipe data. Solusinya adalah memastikan penggunaan generic <Product, Integer> sudah benar.

6. KESIMPULAN

Dengan menggunakan Java Collections, sistem Agri-POS dapat mengelola data transaksi dengan jauh lebih fleksibel. Pemilihan antara List dan Map bergantung pada kebutuhan: List untuk urutan kronologis, dan Map untuk efisiensi manajemen stok/jumlah barang.

7. QUIZ

1. **Jelaskan perbedaan mendasar antara List, Map, dan Set.**

JAWABAN:

List menyimpan elemen secara berurutan dan boleh duplikat. Map menyimpan data dalam pasangan Key-Value (Key harus unik). Set menyimpan elemen secara unik dan tidak menjamin urutan.

2. **Mengapa ArrayList cocok digunakan untuk keranjang belanja sederhana?**

JAWABAN:

Karena ArrayList mudah diimplementasikan, otomatis menyesuaikan ukuran saat data bertambah, dan sangat cepat untuk akses data berurutan (iterasi).

3. **Bagaimana struktur Set mencegah duplikasi data?**

JAWABAN:

Set menggunakan mekanisme *Hashing*. Saat data dimasukkan, Java memeriksa hashCode() dan method equals(). Jika objek dianggap sama dengan yang sudah ada, objek baru tersebut tidak akan ditambahkan.

4. Kapan sebaiknya menggunakan Map dibandingkan List? Jelaskan dengan contoh.

JAWABAN:

Gunakan Map saat kita perlu memetakan suatu nilai ke sebuah kunci.

Contoh: Dalam kasir, Map digunakan untuk menyimpan "Produk A" (Key) dengan jumlah "5" (Value), sehingga kita tidak perlu menulis "Produk A" sebanyak lima kali di dalam daftar.