

LAPORAN MODUL 4 : COLLECTION TYPE
PEMROGRAMAN BERBASIS OBJEK



Dosen pengampu : Radinal Dwiki Novendra, S.T., M.T

Disusun oleh ;

Gusti Panji Widodo (2407113145)

Muhammad Nabil Nadif (2407112714)

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS RIAU
2025

PEMBAHASAN

1. Alur Pemrograman

Pada tahap awal menuntut untuk developer dapat memahami inti sari dari permintaan yang ada. Pada case ini konsumen ingin memiliki aplikasi yang berguna untuk mencatat pesanan harian. Syarat yang ditambahkan konsumen ialah setiap pesanan nantinya akan terdaftar pada sistem, tetapi nama menu tidak boleh double. Selain itu, konsumen juga meminta beberapa fitur tambahan, seperti adanya tampilan awal yang akan berfungsi sebagai home yang akan menampilkan beberapa fitur utama, yaitu: tambah pesanan, hapus pesanan, dan tampilkan pesanan yang mana fitur ini memerlukan input dari pengguna.

Setelah memahami kebutuhan konsumen, langsung saja menyusun kode program agar aplikasi segera selesai. Langkah awal pemrograman dimulai dengan melakukan deklarasi beberapa variabel, melakukan import package untuk menjalankan kelas Scanner serta interface Set dan Map, dan membuat kelas tambahan untuk pesanan. Selanjutnya membuat objek Scanner agar nantinya pengguna dapat melakukan input untuk memilih menu. Setelah itu lakukan deklarasi suatu set bertipe data String, yang mana set ini akan mencegah menampilkan menu yang sama (double) dan lakukan deklarasi Map untuk membuat tampilan pesanan. Setelah itu, buat sebuah percabangan dengan jenis Switch yang bertujuan akan menjalankan fitur utama (tambah, hapus, dan tampilkan pesanan) sesuai dengan input dari pengguna. Untuk menunjang fitur ini juga ditambahkan dua percabangan lainnya, yaitu if-else dan do-while untuk memastikan semua program berjalan dengan baik.

Tahapan terakhir dalam membuat program adalah membuat cetakan yang rapi pada terminal dengan menggunakan `System.out.println()`. Setelah pemrograman selesai perlu dilakukan uji coba untuk mengetes apakah performa program sesuai dengan ekspektasi yang diharapkan oleh konsumen.

2. Penjelasan Kode

- a. Pertama kita akan membuat sebuah class untuk menampung data dari pesanan yang akan kita terima, disini class Pesanan memiliki 2 atribut yakni nama untuk pemesan dan menu untuk makanan yang dipesan.

```
class Pesanan { 4 usages
    private String nama; 2 usages
    private String menu; 2 usages

    public Pesanan(String nama, String menu) { 1 usage
        this.nama = nama;
        this.menu = menu;
    }

    @Override
    public String toString() {
        return "Nama: " + nama + "\nMenu: " + menu;
    }
}
```

- b. Selanjutnya, di Main, kita akan mendeklarasikan beberapa variabel seperti counter untuk pengisian key hashmap secara otomatis, boolean on untuk kondisi dalam do-while kita agar program bisa berulang. Scanner sc menggunakan objek Scanner untuk menerima input, Set “menu” untuk menyimpan menu makanan yang dipesan dan mencegah penduplikasian. Dan terakhir Map “pesanan” untuk menyimpan data pesanan di sebuah HashMap.

```
int counter = 1;
boolean on = true;
Scanner sc = new Scanner(System.in);
Set<String> menu = new HashSet<>();
Map<String, Pesanan> pesanan = new HashMap<>();
```

- c. Kita akan membuat sebuah perulangan do-while agar kode bisa berulang terus berjalan hingga user memilih untuk menutup program

```
do {...} while (on);
```

- d. Di dalam do-while, pertama kita akan menampilkan teks awal ketika user menjalankan program yang berupa sapaan, dan pilihan menu yang bisa diakses. Lalu user diberikan input untuk memilih menu-menu dalam program yang nilainya disimpan dalam variabel “choice”

```
System.out.println("### SELAMAT DATANG DI SISTEM MANAJEMEN PESANAN WARUNG ###");
System.out.println("1. Tambah Pesanan");
System.out.println("2. Tampilkan Pesanan");
System.out.println("3. Hapus Pesanan");
System.out.println("4. Keluar");
System.out.print("Pilih Fitur (1-4) :");
String choice = sc.nextLine();
```

- e. Untuk menampung berbagai pilihan user, kita akan menggunakan switch dimana setiap case itu sesuai dengan angka pilihan dari variabel “choice”.

```
switch (choice) {...}
```

- f. Untuk case 1, yakni menu Tambah Pesanan. Pertama user akan diminta untuk menginput nama pelanggan dan pilihan menunya. Pilihan menu pertama kita cek apakah sudah ada di Set “menu” untuk mencegah duplikasi, jika ada maka program ngulang ke menu awal, jika tidak, maka kita akan membuat pesanan baru dengan key dari counter, dan value nama dengan namaMenu. Lalu, counter ditambah nilainya agar key tetap unik untuk setiap iterasi, dan akan ditampilkan pesan konfirmasi bahwa pesanan berhasil ditambah.

```
case "1":
    System.out.println("## Tambah Pesanan ##\n");
    System.out.print("Masukkan nama pelanggan : ");
    String nama = sc.nextLine();
    System.out.print("Masukkan pilihan menu : ");
    String namaMenu = sc.nextLine();
    if (menu.contains(namaMenu)) {
        System.out.println("Menu sudah tidak bisa dipilih!");
    } else {
        menu.add(namaMenu);
        String key = String.valueOf(counter);
        pesanan.put(key, new Pesanan(nama, namaMenu));
        counter++;
        System.out.println("Pesanan berhasil ditambah!\n");
    }
    break;
```

- g. Untuk case 2, yakni menu Tampilkan Pesanan. Pada case ini akan memprint atau menampilkan semua pesanan yang ada pada sistem dengan memanfaatkan map entry yang akan menampilkan setiap pesanan yang telah ada pada sistem dari key.

```

case "2":
    System.out.println("## Semua Pesanan ##");
    for (Map.Entry<String, Pesanan> entry : pesanan.entrySet()) {
        String key = entry.getKey();
        Pesanan pesan = entry.getValue();

        System.out.println("Pesanan #" + key + "\n" + pesan + "\n");
    }
    break;

```

- h. Case 3, yakni menu Hapus Pesanan. Pengguna akan diminta memasukkan pesanan mana yang ingin dihapus dan setelah itu sistem akan menghapus pesanan itu dari keySet yang sebelumnya menu pesanan tersimpan di sana.

```

case "3":
    System.out.println("## Hapus Pesanan ##");
    System.out.println("Pilih pesanan yang ingin dihapus : ");
    for (String key : pesanan.keySet()) {
        System.out.println(key + " : " + "\t" + pesanan.get(key));
    }
    String pilihanHapus = sc.nextLine();
    if (pesanan.containsKey(pilihanHapus)) {
        pesanan.remove(pilihanHapus);
    }
    break;

```

- i. Output yang ditampilkan pada terminal sebagai hasil akhir

SELAMAT DATANG DI SISTEM MANAJEMEN PESANAN WARUNG

Pilih Fitur (1-3) :

1. Tambah Pesanan
2. Tampilkan Pesanan
3. Hapus Pesanan

1

Tambah Pesanan

Masukkan nama pelanggan : Gusti

Masukkan pilihan menu : Ayam Goreng

SELAMAT DATANG DI SISTEM MANAJEMEN PESANAN WARUNG

Pilih Fitur (1-3) :

1. Tambah Pesanan
2. Tampilkan Pesanan
3. Hapus Pesanan

1

Tambah Pesanan

Masukkan nama pelanggan : Nabil

Masukkan pilihan menu : Ayam Bakar

SELAMAT DATANG DI SISTEM MANAJEMEN PESANAN WARUNG

Pilih Fitur (1-3) :

1. Tambah Pesanan
2. Tampilkan Pesanan
3. Hapus Pesanan

2

Semua Pesanan

Pesanan #1

Nama: Gusti

Menu: Ayam Goreng

Pesanan #2

Nama: Nabil

Menu: Ayam Bakar

Pilih Fitur (1-3) :

1. Tambah Pesanan
2. Tampilkan Pesanan
3. Hapus Pesanan

3

Hapus Pesanan

Pilih pesanan yang ingin dihapus :

1 : Nama: Gusti

Menu: Ayam Goreng

2 : Nama: Nabil

Menu: Ayam Bakar

1

Pesanan telah dihapus!

SELAMAT DATANG DI SISTEM MANAJEMEN PESANAN WARUNG

Pilih Fitur (1-3) :

1. Tambah Pesanan
2. Tampilkan Pesanan
3. Hapus Pesanan

2

Semua Pesanan

Pesanan #2

Nama: Nabil

Menu: Ayam Bakar