

## LATIHAN PEMILIHAN DAN PERULANGAN

### A. Studi Kasus

Manajemen Pesanan di Restoran

### B. Deskripsi

Sebuah restoran ingin mengembangkan sistem manajemen pesanan berbasis objek yang dapat mengelola berbagai pesanan dari pelanggan. Sistem ini harus mampu mencatat pesanan, menghitung total harga, mengelola berbagai kategori pesanan (makanan dan minuman), dan memberikan diskon atau biaya tambahan sesuai kondisi. Dalam studi kasus ini, mahasiswa akan mempelajari penggunaan Statement IF, Statement Case, dan Statement While dalam paradigma Pemrograman Berorientasi Objek (PBO).

### C. Guided

Buat project baru dengan LatihanP4**Nama**. Nama diganti dengan nama kalian, contoh : LatihanP4SenaWijayanto

Buatlah program untuk restoran yang dapat:

1. Mencatat pesanan pelanggan dari berbagai kategori (makanan pembuka, makanan utama, minuman).
2. Menggunakan Statement CASE untuk memilih kategori pesanan.
3. Menghitung total harga berdasarkan jumlah item dan kategori.
4. Menggunakan Statement IF untuk memberikan diskon 10% jika total harga pesanan melebihi Rp 200.000.
5. Menggunakan Statement WHILE untuk menerima beberapa pesanan hingga pelanggan selesai.

### Kode

Menu.java

```
public class Menu {  
    String namaItem;  
    double harga;  
    int kategori; // 1: Pembuka, 2: Utama, 3: Minuman  
  
    public Menu(String namaItem, double harga, int kategori) {
```

```
        this.namaItem = namaItem;
        this.harga = harga;
        this.kategori = kategori;
    }

    // Method untuk menampilkan informasi menu
    public void tampilkanInfo() {
        String jenis = "";
        switch (kategori) {
            case 1: jenis = "Pembuka"; break;
            case 2: jenis = "Utama"; break;
            case 3: jenis = "Minuman"; break;
        }
        System.out.println(namaItem + " - Kategori: " + jenis + " - Harga:
Rp " + harga);
    }
}
```

### Pesanan.java

```
public class Pesanan { Menu[]
    daftarPesanan;      int
    jumlahPesanan;      double
    totalHarga;

    public Pesanan(int kapasitas) { daftarPesanan = new
        Menu[kapasitas]; jumlahPesanan = 0;
        totalHarga = 0;
    }

    // Method untuk menambahkan pesanan public void
    tambahPesanan(Menu item) {
        daftarPesanan[jumlahPesanan] = item;
        totalHarga += item.harga; jumlahPesanan++;
    }

    // Method untuk menampilkan semua pesanan public
    void tampilkanPesanan() {
        System.out.println("Daftar Pesanan:"); for (int i = 0; i
        < jumlahPesanan; i++) {
            daftarPesanan[i].tampilkanInfo();
        }
    }
}
```

}

```
        System.out.println("Total Harga: Rp " + totalHarga);
    }

    // Method untuk menghitung diskon jika total melebihi 200.000
    public double hitungDiskon() {
        if (totalHarga > 200000) {
            return totalHarga * 0.10; // Diskon 10%
        } else {
            return 0;
        }
    }
}
```

File Utama (Nama file sesuai dengan nama project, contoh LatihanP4SenaWijayanto.java)

```
public class LatihanP4SenaWijayanto {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) { Scanner scanner =
        new Scanner(System.in);
        Pesanan pesanan = new Pesanan(10); // Maksimal 10 pesanan char
        tambahPesanan;

        // Perulangan untuk menambah pesanando {
        System.out.print("Masukkan nama item: "); String
        namaItem = scanner.nextLine();

        System.out.print("Masukkan harga item: ");double hargaItem
        = scanner.nextDouble();

        System.out.print("Masukkan kategori (1: Pembuka, 2: Utama, 3:Minuman): ");
        int kategoriItem = scanner.nextInt(); scanner.nextLine(); // Bersihkan
        buffer

        Menu item = new Menu(namaItem, hargaItem, kategoriItem);
        pesanan.tambahPesanan(item);
    }
}
```

```
");  
System.out.print("Apakah ingin menambah pesanan lagi? (y/n): tambahPesanan =  
scanner.next().charAt(0);
```

```
        scanner.nextLine(); // Bersihkan buffer

    } while (tambahPesanan == 'y' || tambahPesanan == 'Y');

    // Tampilkan pesanan dan hitung diskon
    pesanan.tampilkanPesanan();
    double diskon = pesanan.hitungDiskon();
    if (diskon > 0) {
        System.out.println("Anda mendapatkan diskon 10% sebesar Rp " +
diskon);
    }
    System.out.println("Total yang harus dibayar: Rp " +
(pesanan.totalHarga - diskon));
    }
}
```

#### Output :

```
--- exec:3.1.0:exec (default-cli) @ LatihanP4HanafiIlham ---
Masukkan nama item: Nasi
Masukkan harga item: 5000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman): 1
Apakah ingin menambah pesanan lagi? (y/n):y
Masukkan nama item: Es teh
Masukkan harga item: 3000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman): 1
Apakah ingin menambah pesanan lagi? (y/n):n
Daftar Pesanan:
Nasi - Kategori: Pembuka - Harga: Rp 5000.0
Es teh - Kategori: Pembuka - Harga: Rp 3000.0
Total Harga: Rp 8000.0
Total yang harus dibayar: Rp 8000.0

-----
BUILD SUCCESS
-----

Total time: 02:46 min
Finished at: 2024-10-19T17:22:53+07:00
-----
```

#### Penjelasan :

Pada program tersebut membuat simulasi pemesanan sederhana, yang mana user dapat melakukan pemesanan dengan memasukan : nama item, harga item, kategori dan pilihan apakah menambah pesanan atau tidak.

## D. Unguided

Kembangkan Sistem Manajemen Pesanan dengan menambahkan fitur berikut:

1. Tambahkan validasi input kategori.

Validasi input kategori sehingga program hanya menerima angka 1, 2, atau 3, dan jika input tidak valid, program akan meminta input ulang.

2. Tambahkan perhitungan pajak 10%.

Setelah diskon diberikan, tambahkan fitur untuk menghitung pajak sebesar 10% dari total harga yang sudah dikurangi diskon.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
 * change this license
 */
package com.mycompany.latihanp4hanafiilham;
/**
 *
 * @author hanaf
 */
import java.util.Scanner;
public class LatihanP4HanafiIlham {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Pesanan pesanan = new Pesanan(10); // Maksimal 10 pesanan
        char tambahPesanan;

        // Perulangan untuk menambah pesanan
        do {
            System.out.print("Masukkan nama item: ");
            String namaItem = scanner.nextLine();

            System.out.print("Masukkan harga item: ");
            double hargaItem = Double.parseDouble(scanner.nextLine());

            int kategoriItem = 0;
            boolean kategoriValid = false;
            while (!kategoriValid) {
                System.out.print("Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman):");

```

```
String inputKategori = scanner.nextLine();
try {
    kategoriItem = Integer.parseInt(inputKategori);
    if (kategoriItem >= 1 && kategoriItem <= 3) {
        kategoriValid = true;
    } else {
        System.out.println("Kategori tidak valid. Silakan masukkan 1, 2, atau
3.");
    }
} catch (NumberFormatException e) {
    System.out.println("Input tidak valid. Silakan masukkan angka 1, 2, atau
3.");
}
}

Menu item = new Menu(namaItem, hargaItem, kategoriItem);
pesanan.tambahPesanan(item);

System.out.print("Apakah ingin menambahkan pesanan lagi? (y/n): ");
tambahPesanan = scanner.nextLine().charAt(0);
} while (tambahPesanan == 'y' || tambahPesanan == 'Y');

// Tampilkan pesanan dan hitung diskon
pesanan.tampilkanPesanan();
double diskon = pesanan.hitungDiskon();
if (diskon > 0) {
    System.out.println("Anda mendapatkan diskon 10% sebesar Rp " + diskon);
}
System.out.println("Total yang harus dibayar: Rp " + (pesanan.totalHarga -
diskon));

// Hitung dan tampilkan pajak
double totalSetelahDiskon = pesanan.totalHarga - diskon;
double pajak = pesanan.hitungPajak(totalSetelahDiskon);
System.out.println("Pajak 10%: Rp " + pajak);
System.out.println("Total akhir yang harus dibayar: Rp " + (totalSetelahDiskon +
pajak));
}
}
```





Berikut adalah code yang diperbarui, yaitu pada file utama. Berikut adalah perubahan yang dilakukan :

1. Menggunakan “nextLine( )” untuk menghindari buffer.
2. Menggunakan “Double.parseDouble()” untuk input harga.
3. Untuk input kategori menggunakan while dengan try-catch untuk menangani input non-numerik dan memastikan input yang valid (1, 2, atau 3).

Dengan adanya perubahan tersebut, maka ketika user menginputkan kategori dengan huruf, maka akan menampilkan alert :

“Input tidak valid. Silakan masukkan angka 1, 2, atau 3 Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman)”.

Serta terdapat fitur menghitung pajak sebesar 10%.

Output :

```
--- exec:3.1.0:exec (default-cli) @ LatihanP4HanafiIlham ---
Masukkan nama item: nasi
Masukkan harga item: 3000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman): 1
Apakah ingin menambah pesanan lagi? (y/n): y
Masukkan nama item: ikan
Masukkan harga item: 5000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman): utama
Input tidak valid. Silakan masukkan angka 1, 2, atau 3.
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman): 2
Apakah ingin menambah pesanan lagi? (y/n): n
Daftar Pesanan:
nasi - Kategori: Pembuka - Harga: Rp 3000.0
ikan - Kategori: Utama - Harga: Rp 5000.0
Total Harga: Rp 8000.0
Total yang harus dibayar: Rp 8000.0
Pajak 10%: Rp 800.0
Total akhir yang harus dibayar: Rp 8800.0

-----
BUILD SUCCESS
-----

Total time: 34.804 s
Finished at: 2024-10-19T17:50:04+07:00
-----
```