

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**MODUL 2**  
**MATERI PEMILIHAN & PERULANGAN**



Oleh :  
Anggit Refiyan  
2311103142  
S1SI-07C

**PROGRAM STUDI S1 SISTEM INFORMASI**  
**FAKULTAS REKAYASA INDUSTRI**  
**UNIVERSITAS TELKOM PURWOKERTO**  
**2024**

# 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 pesanan        do
    {

        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));
}
}

```

Hasil Running & Penjelasan:

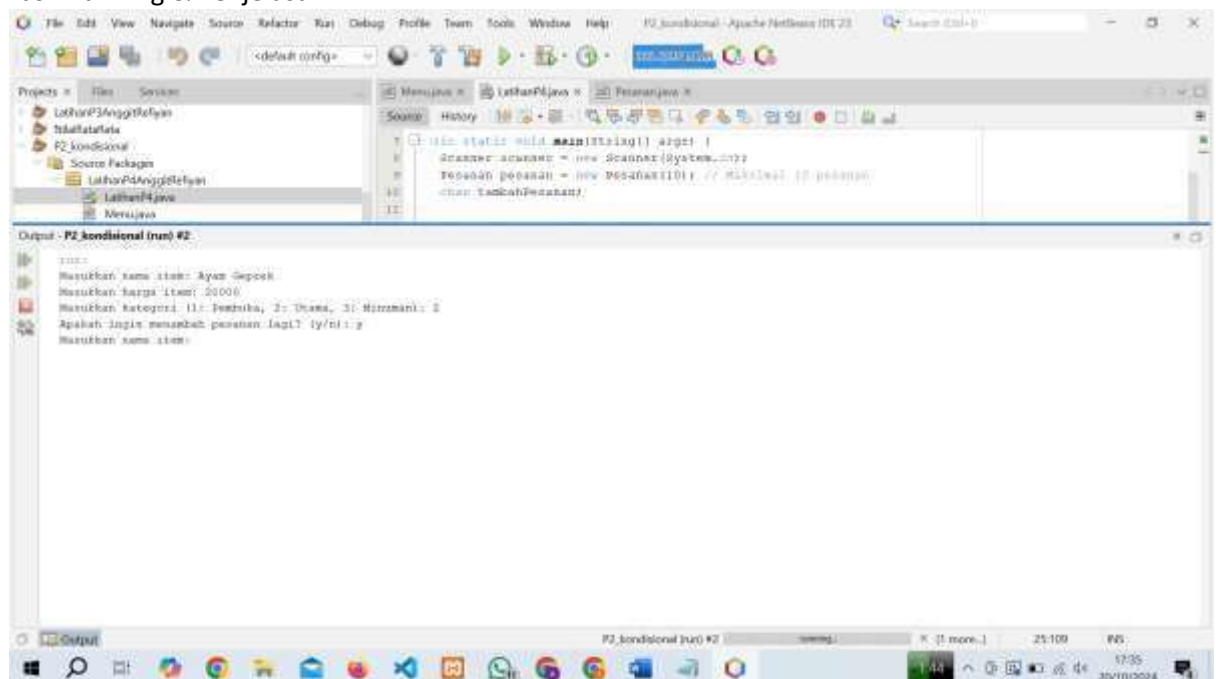
### Hasil Running & Penjelasan:

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

```
// Validasi kategori (hanya menerima Angka 1, 2, atau 3)
int kategoriItem;
do {
    System.out.print("Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman): ");
    kategoriItem = scanner.nextInt();
    if (kategoriItem < 1 || kategoriItem > 3) {
        System.out.println("Angka yang dimasukan tidak valid, Masukkan angka yang sesuai (1,2,3)");
    }
} while (kategoriItem < 1 || kategoriItem > 3);
```

Pada bagian input kategori, menggunakan perulangan do-while untuk memastikan bahwa hanya nilai/angka 1, 2, atau 3 yang dapat di proses. Jika pengguna memasukkan nilai/angka selain 1,2,3 maka program akan meminta input ulang.

Hasil Running & Penjelasan:



Gambar diatas merupakan hasil running dari program yang sudah di modifikasi, gambar ini menunjukan hasil yang benar inputan angka nya.

Output - P2\_kondisional (run) #2

```
run:
Masukkan nama item: Ayam Geprek
Masukkan harga item: 20000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman): 2
Apakah ingin menambah pesanan lagi? (y/n): y
Masukkan nama item: Es Teh
Masukkan harga item: 5000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman): 4
Angka yang dimasukan tidak valid, Masukkan angka yang sesuai (1,2,3).
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman): 3
Apakah ingin menambah pesanan lagi? (y/n): n
```

Gambar diatas merupakan hasil running dari program yang sudah di modifikasi, gambar ini menunjukan hasil yang salah inputan angka nya.



## 2. Tambahkan perhitungan pajak 10%.

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

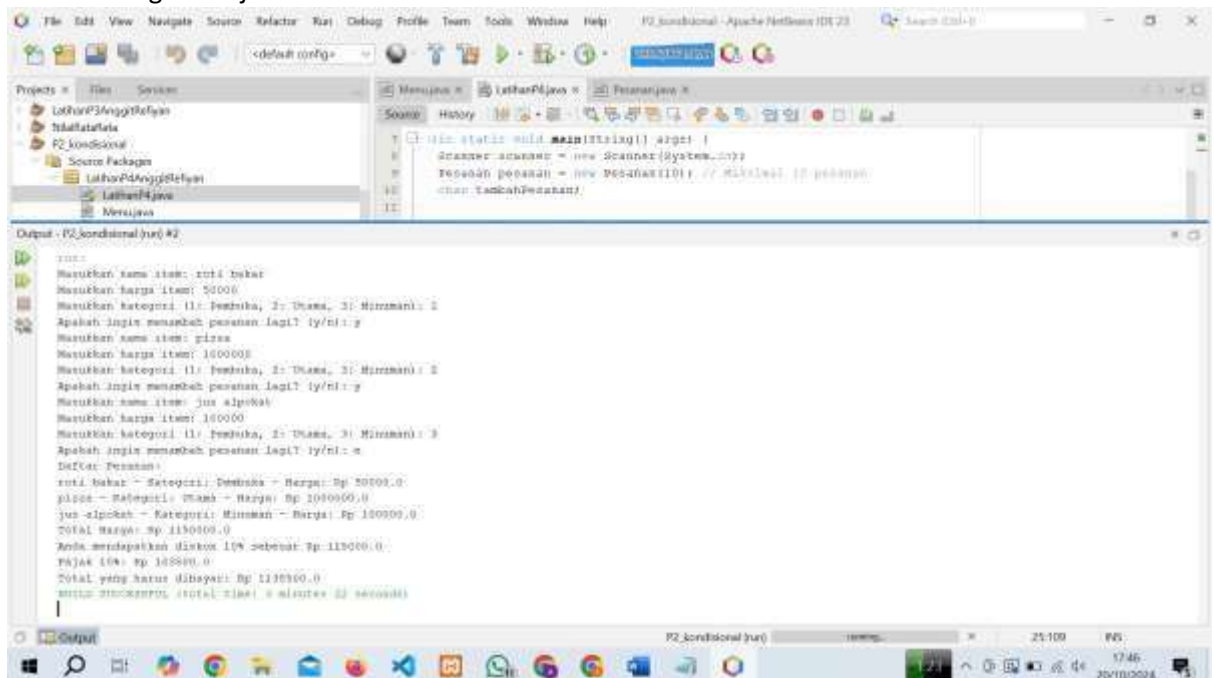
```
// Tampilkan pesanan dan hitung diskon
pesanan.tampilkanPesanan();
double diskon = pesanan.hitungDiskon();
if (diskon > 0) {
    System.out.println("Anda mendapatkan diskon 10% sebesar Rp " + diskon);
}

// Hitung pajak 10% setelah diskon
double totalSetelahDiskon = pesanan.totalHarga - diskon;
double pajak = totalSetelahDiskon * 0.10;
System.out.println("Pajak 10%: Rp " + pajak);

// Tampilkan total yang harus dibayar (setelah diskon dan pajak)
double totalYangHarusDibayar = totalSetelahDiskon + pajak;
System.out.println("Total yang harus dibayar: Rp " + totalYangHarusDibayar);
}
```

Gambar kode diatas merupakan hasil modifikasi untuk menghitung pajak sebesar 10% dari total harga yang sudah dikurangi diskon.

## Hasil Running & Penjelasan:



Gambar diatas merupakan hasil running dari kode program yang dimodifikasi sebelumnya, saya memasukkan inputan pembuka, utama dan minuman kemudian program akan menghitung dan menentukan diskon kemudian di tambahkan adanya penghitungan pajak 10%.