

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 2
PEMILIHAN DAN PERULANGAN**



Oleh : Zoe Satrian
2311103121
S1SI07C

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

➤ Guided

Buat project baru dengan LatihanP4Nama. Nama diganti dengan nama kalian, 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.

- **Penyelesaian:**

Source code

- **Menu.java**

```
package LatihanP4ZoeSatrian;

/**
 * Zoe Satrian / 23211103141
 * SISI-07-C
 */
public class Menu {

    String namaltem;
    double harga;
    int kategori;

    public Menu(String namaltem, double harga, int kategori){
        this.namaltem = namaltem;
        this.harga = harga;
        this.kategori = kategori;
    }
}
```

```
}

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(namaltem + " - Kategori: " + jenis + " - Harga: Rp." + harga);
}
}
```

- **Pesanan.Java**

```
package LatihanP4ZoeSatrian;
```

```
/**
```

```
 * Zoe Satrian
```

```
 * S1SI-07-C
```

```
 */
```

```
public class Pesanan {
```

```
    Menu[] daftarPesanan;
```

```
    int jumlahPesanan;
```

```
    double totalHarga;
```

```
    public Pesanan(int kapasitas){
```

```
        daftarPesanan = new Menu[kapasitas];
```

```
        jumlahPesanan = 0;
```

```
totalHarga = 0;
```

```
}
```

```
public void tambahPesanan(Menu item){  
    daftarPesanan[jumlahPesanan] = item;  
    totalHarga += item.harga;  
    jumlahPesanan++;  
}
```

```
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);  
}
```

```
public double hitungDiskon(){  
    if (totalHarga > 200000) {  
        return totalHarga * 0.10;  
    } else {  
        return 0;  
    }  
}
```

```
}
```

- **LatihanP4ZoeSatrian.Java**

```
package LatihanP4ZoeSatrian;
```

```
import java.util.Scanner;
```

```
/**
```

```
 * Zoe Satrian
```

```
 * S1SI-07-C
```

```
 */
```

```
public class LatihanP4ZoeSatrian {
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        Pesanan pesanan = new Pesanan(10);
```

```
        char tambahPesanan;
```

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

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));
}
}

```

- **Screenshot Hasil :**

Output - LatihanP4ZoeSatrian (run)

```

Masukkan harga item : 25000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman) : 1
Apakah ingin menambah pesanan lagi (Y/N) : N
Daftar Pesanan:
burger - Kategori: Pembuka - Harga: Rp.25000.0
Total Harga: Rp.25000.0
Total yang harus dibayar: Rp.25000.0
BUILD SUCCESSFUL (total time: 17 seconds)

```

Output - LatihanP4ZoeSatrian (run)

```

run:
Masukkan nama item : CHICKEN BUCKET
Masukkan harga item : 550000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman) : 2
Apakah ingin menambah pesanan lagi (Y/N) : N
Daftar Pesanan:
CHICKEN BUCKET - Kategori: Utama - Harga: Rp.550000.0
Total Harga: Rp.550000.0
Anda mendapatkan diskon 10% sebesar Rp.55000.0
Total yang harus dibayar: Rp.495000.0
BUILD SUCCESSFUL (total time: 37 seconds)
||

```

- **Penjelasan Program:**

Program di atas merupakan sebuah program restoran yang dirancang untuk menerima inputan dari user terkait pesanan makanan atau minuman beserta harganya. Program ini menggunakan tiga kelas utama, yaitu kelas Menu, Pesanan, dan LatihanP4ZoeSatrian sebagai file utama. Pada kelas Menu, diinisialisasi variabel namaItem, harga, dan kategori yang berfungsi untuk menyimpan data input dari user. Selanjutnya, di dalam kelas Pesanan, terdapat array daftarPesanan yang menyimpan daftar item dari kelas Menu, variabel jumlahPesanan yang mencatat jumlah item yang dipesan, serta totalHarga yang digunakan untuk menghitung total keseluruhan harga pesanan. Fungsi tambahPesanan dalam kelas ini berfungsi untuk menambahkan item ke daftar pesanan dan menghitung total harga. Kemudian, fungsi tampilkanPesanan digunakan untuk menampilkan daftar pesanan secara lengkap dengan informasi harga dan kategori item, menggunakan perulangan. Kelas Pesanan juga dilengkapi dengan fungsi hitungDiskon, yang akan memberikan diskon 10% jika total harga pesanan melebihi Rp200.000.

Pada file utama LatihanP4ZoeSatrian, program menerima input dari saya, seperti nama item, harga, dan kategori. saya juga diberi opsi untuk menambah pesanan menggunakan perulangan do-while. Setelah semua pesanan dimasukkan, informasi lengkap pesanan akan ditampilkan dengan memanggil fungsi tampilkanPesanan. Jika total harga pesanan memenuhi syarat untuk diskon, yaitu melebihi Rp200.000, program akan memberikan informasi diskon 10% dan menghitung total harga yang harus dibayar setelah diskon diterapkan.

➤ 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.

- **Penyelesaian:**

- **Source Code:**

Unguided1.Java

```
package Unguided1Zoe;
// Zoe Satrian / 2311103141
// S1SI07C

import java.util.Scanner;

public class Unguided1 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Pesanan pesanan = new Pesanan(10);
        char tambahPesanan;

        do{
            System.out.print("Masukkan nama item : ");
            String namaItem = scanner.nextLine();
            System.out.print("Masukkan harga item : ");
            double hargaItem = scanner.nextDouble();

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

                if(kategoriItem <= 3){
                    kategori = true;
                }else{
                    System.out.println("Kategori salah input");
                }
            }
        }
```

```
scanner.nextLine();
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();
}while(tambahPesanan == 'y' || tambahPesanan == 'Y');

System.out.println("");
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));
}
}
```

Menu.Java

```
package Unguided1Zoe;

// Zoe Satrian / 2311103141
// S1SI07C

public class Menu {
    String namaItem;
    double harga;
    int kategori;

    public Menu(String namaItem, double harga, int kategori){
        this.namaItem = namaItem;
        this.harga = harga;
        this.kategori = kategori;
    }
    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

```
package Unguided1Zoe;
// Zoe Satrian / 2311103141
// S1SI07C

public class Pesanan {

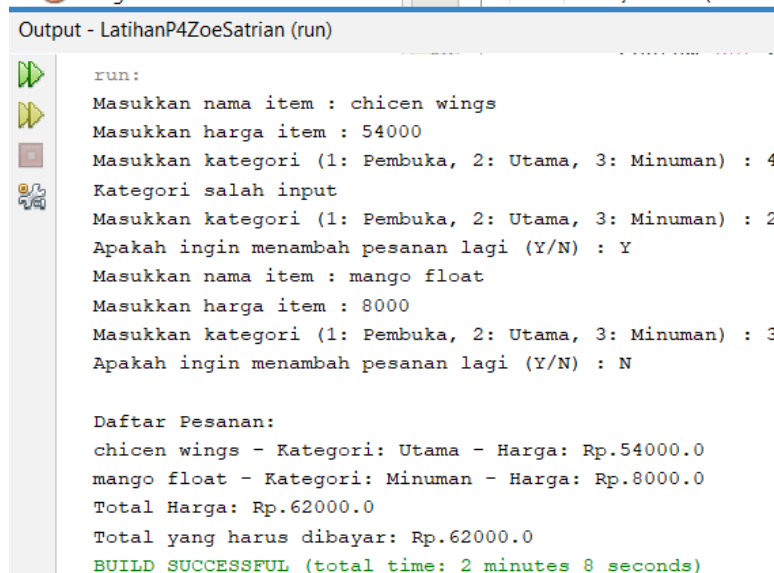
    Menu[] daftarPesanan;
    int jumlahPesanan;
    double totalHarga;

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

    public void tambahPesanan(Menu item){
        daftarPesanan[jumlahPesanan] = item;
        totalHarga += item.harga;
        jumlahPesanan++;
    }
}
```

```
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);
}
public double hitungDiskon(){
    if (totalHarga > 200000) {
        return totalHarga * 0.10;
    } else {
        return 0;
    }
}
}
```

- **Screenshoot Hasil Program :**



```
Output - LatihanP4ZoeSatrian (run)

run:
Masukkan nama item : chicken wings
Masukkan harga item : 54000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman) : 4
Kategori salah input
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman) : 2
Apakah ingin menambah pesanan lagi (Y/N) : Y
Masukkan nama item : mango float
Masukkan harga item : 8000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman) : 3
Apakah ingin menambah pesanan lagi (Y/N) : N

Daftar Pesanan:
chicken wings - Kategori: Utama - Harga: Rp.54000.0
mango float - Kategori: Minuman - Harga: Rp.8000.0
Total Harga: Rp.62000.0
Total yang harus dibayar: Rp.62000.0
BUILD SUCCESSFUL (total time: 2 minutes 8 seconds)
```

- **Penjelasan :**

Pada kode utama, hanya dilakukan sedikit perubahan untuk menambahkan validasi input kategori, sedangkan dua kelas lainnya (Menu dan Pesanan) tetap sama. Di dalam file utama, ditambahkan validasi input menggunakan perulangan do-while untuk memastikan pengguna hanya memasukkan angka 1, 2, atau 3 sebagai kategori item. Pengguna akan diminta memasukkan ulang kategori jika inputan tidak valid. Jika inputan kategori berada di luar rentang tersebut (1-3), program akan menampilkan pesan kesalahan "Kategori salah input". Masukkan angka 1, 2, atau 3" dan mengulang input. Perulangan validasi berhenti ketika input kategori benar, yaitu angka 1, 2, atau 3. Proses ini dilakukan tanpa mengubah struktur kode utama lainnya, sehingga fungsionalitas program tetap berjalan dengan lancar setelah input kategori berhasil divalidasi.

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

- Penyelesaian :

Source Code :

Unguided2..Java

```
package Unguided2Zoe;
// Zoe Satrian / 2311103141
// S1SI07C

import java.util.Scanner;

public class Unguided2 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Pesanan pesanan = new Pesanan(10);
        char tambahPesanan;

        do{
            System.out.print("Masukkan nama item : ");
            String namaItem = scanner.nextLine();
            System.out.print("Masukkan harga item : ");
            double hargaItem = scanner.nextDouble();

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

                if(kategoriItem <= 3){
                    kategori = true;
                }else{
                    System.out.println("Kategori salah input");
                }
            }
            scanner.nextLine();
            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();
            System.out.println("");
        }
```

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

    System.out.println("");
    pesanan.tampilkanPesanan();

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

    double hargaDiskon = pesanan.totalHarga - diskon;
    double pajak = pesanan.hitungPajak(hargaDiskon);
    System.out.println("Pajak 10% Rp." + pajak);
    System.out.println("");
    System.out.println("Total yang harus dibayar (Sudah termasuk pajak): Rp." + (hargaDiskon +
    pajak));
    }
    }
```

Pesanan.Java

```
package Unguided2Zoe;
// Zoe Satrian / 2311103141
// S1SI07C

public class Pesanan {

    Menu[] daftarPesanan;
    int jumlahPesanan;
    double totalHarga;

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

    public void tambahPesanan(Menu item){
        daftarPesanan[jumlahPesanan] =
        item;
        totalHarga += item.harga;
        jumlahPesanan++;

    }
}
```

```
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);
}

public double hitungDiskon(){
    if (totalHarga > 200000) {
        return totalHarga * 0.10;
    } else {
        return 0;
    }
}

public double hitungPajak(double hargaPajak){
    return hargaPajak * 0.10;
}
}
```

Menu.Java

```
package Unguided2Zoe;
// Zoe Satrian / 2311103141
// S1SI-07-C
public class Menu {
    String namaItem;
    double harga;
    int kategori;
    public Menu(String namaItem, double harga, int kategori){
        this.namaItem = namaItem;
        this.harga = harga;
        this.kategori = kategori;
    }

    public Menu(String namaItem, double harga, int kategori){
        this.namaItem = namaItem;
        this.harga = harga;
        this.kategori = kategori;
    }
    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);
}
}

```

- **Screenshoot Hasil :**

Output - LatihanP4ZoeSatrian (run)

```

run:
Masukkan nama item : kentang goreng
Masukkan harga item : 15000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman) : 1
Apakah ingin menambah pesanan lagi (Y/N) : Y

Masukkan nama item : coca cola
Masukkan harga item : 20000
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman) : 4
Kategori salah input
Masukkan kategori (1: Pembuka, 2: Utama, 3: Minuman) : 3
Apakah ingin menambah pesanan lagi (Y/N) : N

Daftar Pesanan:
kentang goreng - Kategori: Pembuka - Harga: Rp.15000.0
coca cola - Kategori: Minuman - Harga: Rp.20000.0
Total Harga: Rp.35000.0
Pajak 10% Rp.3500.0

Total yang harus dibayar (Sudah termasuk pajak): Rp.38500.0
BUILD SUCCESSFUL (total time: 2 minutes 3 seconds)

```

- **Penjelasan:**

Pada program ini, terdapat perubahan sederhana namun signifikan pada file utama untuk menambahkan validasi input kategori. Program ini dirancang untuk menerima pesanan dari pengguna dengan memasukkan nama item, harga, dan kategori (Pembuka, Utama, atau Minuman). Validasi dilakukan menggunakan variabel boolean kategori, yang memastikan pengguna hanya memasukkan nilai kategori yang valid, yaitu angka 1, 2, atau 3. Jika pengguna memasukkan angka di luar rentang tersebut, program akan menampilkan pesan "Kategori salah input" dan terus meminta pengguna untuk memasukkan ulang hingga nilai yang benar diberikan. Selain itu, program ini juga menghitung diskon 10% jika memenuhi syarat, serta menambahkan perhitungan pajak 10% setelah diskon diterapkan. Setelah seluruh proses selesai, total pesanan, diskon, pajak, dan jumlah yang harus dibayar akan ditampilkan secara lengkap. Pendekatan ini menjaga alur program tetap interaktif dan akurat, memberikan pengalaman pengguna yang lebih baik dengan penanganan input yang lebih aman dan perhitungan otomatis yang lengkap.