

## TUGAS PERTEMUAN 1

**NAMA : SALSABILLA PUTERI SANDI WARDANA**  
**NIM : H1D021063**  
**SHIFT BARU : F**  
**SHIFT LAMA : C**

### Penjelasan mengenai source code

1. `import 'dart:io';`

Digunakan untuk mengimpor Pustaka Dart yang menyediakan fungsi agar dapat berinteraksi memberikan input/output pada sistem operasi, pada sistem kali ini dart:io digunakan untuk membaca input dari pengguna melalui terminal.

2. 

```
class ItemBelanjaan {  
    String kode;  
    String nama;  
    double hargaPerUnit;  
  
    ItemBelanjaan(this.kode, this.nama, this.hargaPerUnit);  
}
```

Ini adalah kelas yang digunakan untuk mewakili item belanjaan, setiap item punya 3 atribut yaitu kode (kode barang), nama (nama barang), dan hargaPerUnit (harga barang per unit)

Kelas ini berfungsi untuk menyimpan detail barang yang dijual di toko

3. 

```
class Kasir {  
    List<ItemBelanjaan> daftarBarang = [];  
    List<ItemBelanjaan> daftarBelanja = [];  
    Map<String, int> jumlahBelanja = {};
```

Kelas kasir ini adalah yang utama dalam sistem ini, didalamnya ada 3 variabel utama yaitu:

- a. daftarBarang : menyimpan daftar barang yang tersedia di toko
- b. daftarBelanja : menyimpan barang yang dibeli
- c. jumlahBelanja : menyimpan jumlah barang yang dibeli dengan key nya kode barang

```
void tambahBarang(ItemBelanjaan item) {
    daftarBarang.add(item);
}
```

4.

Metode ini digunakan untuk menambahkan daftar barang yang tersedia ditoko (tidak diinputkan di terminal) hanya digunakan untuk menambah barang ke dalam daftar barang yang dapat dibeli

```
void tambahItemBelanjaan(String kode, int jumlah) {
    ItemBelanjaan? item = daftarBarang.firstWhere((item) => item.kode == kode,
        orElse: () => ItemBelanjaan('', '', 0));

    if (item.kode.isNotEmpty) {
        if (jumlahBelanja.containsKey(kode)) {
            jumlahBelanja[kode] = jumlahBelanja[kode]! + jumlah;
        } else {
            jumlahBelanja[kode] = jumlah;
        }
        print('${item.nama} sebanyak $jumlah telah ditambahkan.');
```

5.

Metode ini akan menerima kode barang dan jumlahnya yang diinput pengguna melalui terminal, lalu akan mencari barang dengan kode tersebut di daftarBarang, kalau ada maka barang tersebut ditambahkan ke jumlahBelanja.

Jika kode yang dimasukkan tidak sesuai maka akan menampilkan “kode barang tidak ditemukan”

*jumlahBelanja[kode] = jumlahBelanja[kode]! + jumlah;*

bagian tersebut digunakan untuk mengecek apakah kode barang yang diinput sudah ada di keranjang (jumlahBelanja) sebelumnya maka jika diinput jumlah belanja baru akan menambah jumlahBelanja sebelumnya, jika belum ada di keranjang maka akan ditambahkan ke jumlahBelanja dengan jumlah sesuai yang dimasukkan oleh pengguna.

Setelah berhasil dimasukkan maka program akan menampilkan pesan konfirmasi :

*print('\${item.nama} sebanyak \$jumlah telah ditambahkan.');*

```
double hitungTotalHarga() {
    double total = 0;
    for (var kode in jumlahBelanja.keys) {
        ItemBelanjaan item = daftarBarang.firstWhere((item) => item.kode == kode);
        total += item.hargaPerUnit * jumlahBelanja[kode]!;
    }
    return total;
}
```

6.

Digunakan untuk menghitung total harga belanjaan berdasarkan barang yang telah dibeli sebelum diskon

```
double hitungDiskon(double total) {
    if (total > 100000) {
        return total *
            0.1; //ini untuk dikasih diskon 10% pas udah diatas 100k total belanja
    } else {
        return 0;
    }
}
```

7.

Jika barang yang dibeli lebih dari 100.000 maka toko memberikan diskon sebesar 10%, jika dibawah 100.000 maka akan mencetak harga sebelum diskon.

```
void tampilkanTotal() {
    double total = hitungTotalHarga();
    double diskon = hitungDiskon(total);
    double totalSetelahDiskon = total - diskon;
```

8.

Digunakan untuk menampilkan rincian belanja termasuk daftar barang apa saja yang dibeli, total harga sebelum diskon, banyaknya diskon (jika ada) dan harga akhir setelah diskon (jika ada)

```
print('\n-----');
print('Salsa Shop\nMadiun');
print('Kasir: Salsa');
print('-----');
print('Daftar Belanja:');
for (var kode in jumlahBelanja.keys) {
    ItemBelanjaan item = daftarBarang.firstWhere((item) => item.kode == kode);
    print(
        '${item.nama} x${jumlahBelanja[kode]} - Rp ${item.hargaPerUnit * jumlahBelanja[kode]!.toStringAsFixed(2)}');
}

print('-----');
print('Total belanja: Rp ${total.toStringAsFixed(2)}');
if (diskon > 0) {
    print('Diskon 10%: Rp ${diskon.toStringAsFixed(2)}');
} else {
    print('Tidak ada diskon.');
```

9.

Digunakan untuk menampilkan struk belanja di terminal ketika proses belanja telah selesai, struk ini memberikan informasi mengenai toko, kasir, daftar barang yang dibeli, total belanja, diskon (jika ada), dan harga akhir setelah diskon (jika ada) lalu diberikan juga pesan penutup.

```
void tampilkanDaftarBarang() {  
    print('Daftar Barang:');  
    for (var item in daftarBarang) {  
        print(  
            '${item.kode}: ${item.nama} - Rp ${item.hargaPerUnit.toStringAsFixed(2)}');  
    }  
}
```

10.

Menampilkan seluruh barang yang tersedia di toko lengkap dengan kode barang, nama, dan harganya per unit.

```
void main() {  
    Kasir kasir = Kasir();
```

11.

Fungsi ini adalah titik awal program dimana beberapa item ditambahkan ke daftar barang lalu sistem menunggu input pengguna untuk membeli barang apa saja

```
// Menambahkan barang ke daftar  
kasir.tambahBarang(ItemBelanjaan('B1', 'Beras', 50000));  
kasir.tambahBarang(ItemBelanjaan('B2', 'Minyak Goreng', 20000));  
kasir.tambahBarang(ItemBelanjaan('B3', 'Gula Pasir', 15000));  
kasir.tambahBarang(ItemBelanjaan('B4', 'Garam', 3000));  
kasir.tambahBarang(ItemBelanjaan('B5', 'Lada', 1000));  
kasir.tambahBarang(ItemBelanjaan('B6', 'Kecap', 5000));
```

12.

```

print('Salsa Shop\n Madiun \n ----- \n ');

// Menampilkan daftar barang
kasir.tampilkanDaftarBarang();

bool selesai = false;

while (!selesai) {
    print('\nMasukkan kode barang (ketik "selesai" untuk menghitung total): ');
    String? kode = stdin.readLineSync();

    if (kode?.toLowerCase() == 'selesai') {
        selesai = true;
        kasir.tampilkanTotal();
    } else {
        print('Masukkan jumlah: ');
        String? jumlahInput = stdin.readLineSync();
        int jumlah = int.parse(jumlahInput!);

        kasir.tambahItemBelanjaan(kode!, jumlah);
    }
}
}

```

Digunakan untuk menambahkan barang yang nantinya dapat dilihat di terminal, terdapat logika interaktif di while loop yang berfungsi bahwa loop ini akan terus berjalan hingga pengguna mengetik “selesai”. Selama loop, pengguna akan terus diminta kode barang dan jumlah barang yang akan dibeli hingga mengetik “selesai” untuk menampilkan total harga belanja.

### 13. OUTPUT

```
PS D:\SEMESTER 7\praktikum mobile\SALSABILLAPUTERISANDI
WARDANA_H1D021063_SHIFT-F_PRAKTIKUMMOBILE2024\pertemuan
1> dart run
Building package executable...
Built pertemuan1:pertemuan1.
Salsa Shop
Madiun
-----

Daftar Barang:
B1: Beras - Rp 50000.00
B2: Minyak Goreng - Rp 20000.00
B3: Gula Pasir - Rp 15000.00
B4: Garam - Rp 3000.00
B5: Lada - Rp 1000.00
B6: Kecap - Rp 5000.00

Masukkan kode barang (ketik "selesai" untuk menghitung
total):
B1
Masukkan jumlah:
2
Beras sebanyak 2 telah ditambahkan.

Masukkan kode barang (ketik "selesai" untuk menghitung
total):
1
```

```
Masukkan kode barang (ketik "selesai" untuk menghitung
total):
selesai

-----
Salsa Shop
Madiun
Kasir: Salsa
-----

Daftar Belanjaan:
Beras x2 - Rp 100000.00
-----

Total belanja: Rp 100000.00
Tidak ada diskon.
Total setelah diskon: Rp 100000.00
-----

Terima kasih telah berbelanja di Salsa Shop!
-----
```