

# **UJIAN TENGAH SEMESTER**

## **Membuat Program Kasir**



**NAMA : ROUFAN AWALUNA ROMADHON**

**NIM : 312410423**

**KELAS : TI.24.A.3**

**Mata kuliah: Algoritma dan Struktur Data**

**Dosen: M. Fatchan, S.Kom, M.Kom**

# Membuat Program Kasir

Program kasir ini dibuat dengan Bahasa python untuk memenuhi kebutuhan kasir, dengan fitur:

- Menambah barang ke dalam keranjang
- Menampilkan daftar barang
- Menghitung total harga
- Mencetak struk belanja

## 1. Kelas “Item”

```
class Item:
    def __init__(self, nama, harga, jumlah):
        self.nama = nama
        self.harga = harga
        self.jumlah = jumlah

    def total_harga(self):
        return self.harga * self.jumlah
```

Kelas ini merepresentasikan satu barang yang akan ditambahkan ke keranjang. Setiap barang memiliki atribut:

- nama: Nama barang.
- harga: Harga per unit barang.
- jumlah: Jumlah unit barang.

Metode `total_harga` menghitung total harga barang berdasarkan harga per unit dikalikan dengan jumlah barang. Contoh penggunaan:

```
item = Item("Indomie", 3500, 6)
total_harga_item = item.get_total_price() # Totalnya 21000
```

## 2. Kelas “Keranjang”

```
class Keranjang:
    def __init__(self):
        self.items = []

    def add_item(self, nama, harga, jumlah):
        item = Item(nama, harga, jumlah)
        self.items.append(item)
```

```

def display_items(self):
    if not self.items:
        print("Keranjang masih kosong.")
    else:
        print("Daftar Barang dalam Keranjang:")
        for index, item in enumerate(self.items, start=1):
            print(f"{index}. {item.nama} - {item.jumlah} x Rp{item.harga}
= Rp{item.total_harga()}")

def calculate_total(self):
    return sum(item.total_harga() for item in self.items)

def print_receipt(self):
    print("\nStruk Belanja")
    print("=" * 20)
    self.display_items()
    print("=" * 20)
    print(f"Total Harga: Rp{self.calculate_total()}")
    print("Terima kasih sudah berbelanja!\n")

```

Kelas `Keranjang` berfungsi sebagai keranjang belanja, yang dapat menyimpan beberapa barang.

- **Atribut `items`:** Menyimpan daftar objek `Item` yang telah ditambahkan ke dalam keranjang.
- **Metode `add_item`:** Digunakan untuk menambahkan barang baru ke dalam keranjang.
- **Metode `display_items`:** Menampilkan seluruh daftar barang dalam keranjang. Jika keranjang kosong, program akan menampilkan pesan "Keranjang masih kosong". Jika tidak kosong, daftar barang akan ditampilkan lengkap dengan nama barang, harga per unit, jumlah barang, dan total harga per barang.
- **Metode `calculate_total`:** Menghitung total harga semua barang dalam keranjang menggunakan metode `get_total_price` dari setiap objek `Item`. Hasil perhitungan dijumlahkan menggunakan fungsi `sum`.
- **Metode `print_receipt`:** Mencetak struk belanja yang berisi daftar barang dan total harga seluruh barang dalam keranjang. Metode ini juga menampilkan ucapan terima kasih.

### 3. Fungsi “Main()”

```
def main():  
    cart = Keranjang()
```

- **Tujuan:** main() adalah fungsi yang menjalankan seluruh alur program kasir di terminal
- **Atribut:**
  - cart: Objek Keranjang digunakan untuk menyimpan barang-barang yang ditambahkan.

Loop Utama while True

```
while True:  
    print("\nPilih opsi:")  
    print("1. Tambah barang ke keranjang")  
    print("2. Tampilkan daftar barang")  
    print("3. Hitung total harga")  
    print("4. Cetak struk")  
    print("5. Keluar")  
  
    choice = input("Masukkan pilihan (1-5): ")
```

- **Tujuan:** Menampilkan menu pilihan kepada pengguna dan menerima input dari pengguna.
- **Opsi menu:**
  1. Tambah barang ke keranjang.
  2. Tampilkan daftar barang di keranjang.
  3. Hitung total harga barang di keranjang.
  4. Cetak struk belanja.
  5. Keluar dari program.

### Proses Berdasarkan Pilihan Pengguna

```
if choice == '1':  
    nama = input("Masukkan nama barang: ")  
    harga = float(input("Masukkan harga barang: "))  
    jumlah = int(input("Masukkan jumlah barang: "))  
    cart.add_item(nama, harga, jumlah)  
    print(f"{nama} berhasil ditambahkan ke keranjang.")
```

#### 1. Tambah barang (Pilihan 1)

- Meminta input nama barang, harga, dan jumlah.
- Memanggil add\_item dari Keranjang untuk menambahkan barang ke keranjang.

Contoh Output:

```
Pilih opsi:
1. Tambah barang ke keranjang
2. Tampilkan daftar barang
3. Hitung total harga
4. Cetak struk
5. Keluar
Masukkan pilihan (1-5): 1
Masukkan nama barang: Indomie
Masukkan harga barang: 3500
Masukkan jumlah barang: 6
Indomie berhasil ditambahkan ke keranjang.

Pilih opsi:
1. Tambah barang ke keranjang
2. Tampilkan daftar barang
3. Hitung total harga
4. Cetak struk
5. Keluar
Masukkan pilihan (1-5): 1
Masukkan nama barang: Spagetti WOW
Masukkan harga barang: 3000
Masukkan jumlah barang: 6
Spagetti WOW berhasil ditambahkan ke keranjang.
```

```
elif choice == '2':
    cart.display_items()
```

## 2. Tambahkan Daftar Pilihan Barang (Pilihan 2)

- Memanggil `display_items` untuk menampilkan barang-barang dalam keranjang.

Contoh Output:

```
Pilih opsi:
1. Tambah barang ke keranjang
2. Tampilkan daftar barang
3. Hitung total harga
4. Cetak struk
5. Keluar
Masukkan pilihan (1-5): 2
Daftar Barang dalam Keranjang:
1. Indomie - 6 x Rp3500.0 = Rp21000.0
2. Spagetti WOW - 6 x Rp3000.0 = Rp18000.0
```

```
elif choice == '3':
    total = cart.calculate_total()
    print(f"Total harga: Rp{total}")
```

### 3. Hitung Total Harga (Pilihan 3)

- Menghitung dan menampilkan total harga semua barang di keranjang menggunakan `calculate_total`.

Contoh Output:

```
Pilih opsi:
1. Tambah barang ke keranjang
2. Tampilkan daftar barang
3. Hitung total harga
4. Cetak struk
5. Keluar
Masukkan pilihan (1-5): 3
Total harga: Rp39000.0
```

```
elif choice == '4':
    cart.print_receipt()
```

### 4. Cetak Struk (pilihan 4)

- Mencetak struk belanja yang berisi daftar barang dan total harga menggunakan `print_receipt`.

Contoh Output:

```
Pilih opsi:
1. Tambah barang ke keranjang
2. Tampilkan daftar barang
3. Hitung total harga
4. Cetak struk
5. Keluar
Masukkan pilihan (1-5): 4

Struk Belanja
=====
Daftar Barang dalam Keranjang:
1. Indomie - 6 x Rp3500.0 = Rp21000.0
2. Spagetti WOW - 6 x Rp3000.0 = Rp18000.0
=====
Total Harga: Rp39000.0
Terima kasih sudah berbelanja!
```

```
elif choice == '5':  
    print("Terima kasih! Program berakhir.")  
    break
```

#### 5. Keluar (Pilihan 5)

- Menampilkan pesan perpisahan dan mengakhiri loop `while`, sehingga program selesai.

Contoh Output:

```
Pilih opsi:  
1. Tambah barang ke keranjang  
2. Tampilkan daftar barang  
3. Hitung total harga  
4. Cetak struk  
5. Keluar  
Masukkan pilihan (1-5): 5  
Terima kasih! Program berakhir.
```

```
else:  
    print("Pilihan tidak valid. Silakan coba lagi.")
```

Fungsi:

- Validasi Input: Menampilkan pesan kesalahan jika pengguna memasukkan pilihan di luar opsi 1 hingga 5.
- Instruksi Ulang: Pesan "Pilihan tidak valid. Silakan coba lagi." muncul, dan program kembali ke menu, memberi pengguna kesempatan untuk mengoreksi input tanpa keluar dari program.

Contoh Output:

```
Pilih opsi:  
1. Tambah barang ke keranjang  
2. Tampilkan daftar barang  
3. Hitung total harga  
4. Cetak struk  
5. Keluar  
Masukkan pilihan (1-5): 0  
Pilihan tidak valid. Silakan coba lagi.
```