

## E-COMMERCE



**Disusun Oleh:**

### **KELOMPOK 11**

**MUHAMMAD RIZKY FEBRIANTO                    2409116045**

**ABDURAHMAN AL FARISY                    2409116055**

**DINDA AULIA RIZKY                    2409116076**

**Asisten Laboratorium:**

**MUHAMMAD ARIFIN DAFA**

**2309116059**

**MUHAMAD AIDIL MIRZA**

**2309116040**

## **Kata Pengantar**

Dengan nama Allah Yang Maha Pengasih dan Maha Penyayang, segala puji dan syukur kami panjatkan ke hadirat-Nya. Shalawat serta salam semoga selalu tercurah kepada junjungan kita, Nabi Muhammad SAW, keluarga, sahabat, dan para pengikut setianya.

Kami dengan rendah hati menyusun laporan proyek akhir ini sebagai salah satu syarat kelulusan ujian akhir semester di Universitas Mulawarman, Kalimantan Timur. Laporan ini dibuat dalam rangka mengeksplorasi dan mendokumentasikan proyek pemrograman kami yang berfokus pada *e-commerce*.

Penyusunan laporan ini tidak mungkin terwujud tanpa bimbingan, dukungan, serta dorongan dari berbagai pihak. Kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bang Daffa selaku Asisten Laboratorium Konsul yang telah memberikan arahan, saran, serta pengawasan sepanjang perjalanan pelaksanaan PA.
2. Para abang selaku Asisten Laboratorium Sistem Informasi 2024
3. Seluruh teman mahasiswa yang sudah berkenan memberikan dukungan dan bantuan

Kami menyadari bahwa laporan ini tidak sempurna, dan segala kritik dan saran yang membangun sangat kami hargai untuk perbaikan di masa mendatang. Semoga laporan tugas akhir ini dapat memberikan pemahaman yang baik tentang proyek kami dan bermanfaat bagi semua pihak yang membacanya.

November, 2024

## Daftar Isi

<b>Kata Pengantar .....</b>	<b>ii</b>
<b>Daftar Isi .....</b>	<b>iii</b>
<b>Daftar Gambar .....</b>	<b>iv</b>
<b>Daftar Tabel.....</b>	<b>vi</b>
<b>BAB I .....</b>	<b>7</b>
1.1    Deskripsi Masalah.....	7
1.2    Rumusan Masalah.....	7
1.3    Batasan Masalah .....	8
1.4    Tujuan .....	8
1.5    Manfaat .....	8
<b>BAB II.....</b>	<b>9</b>
2.1 Analisis Program.....	9
2.2 Flowchart.....	12
<b>BAB III.....</b>	<b>24</b>
3.1 Implementasi Program .....	24
3.2 Alur Program.....	101
3.3 Source Code .....	118
<b>BAB IV .....</b>	<b>169</b>
4.1 Kesimpulan .....	169
4.2 Saran.....	169
<b>DAFTAR PUSTAKA .....</b>	<b>171</b>
<b>LAMPIRAN.....</b>	<b>172</b>

## Daftar Gambar

<b>Gambar 2. 1</b> Flowchart Login .....	12
<b>Gambar 2. 2</b> Flowchart Menu User.....	13
<b>Gambar 2. 3</b> Flowchart Menu Makanan (User) .....	14
<b>Gambar 2. 4</b> Flowchart Opsi Order Makanan (User) .....	15
<b>Gambar 2. 5</b> Flowchart Menu Barang (User) .....	16
<b>Gambar 2. 6</b> Flowchart Opsi Order Barang (User).....	17
<b>Gambar 2. 7</b> Flowchart Cek Keranjang (User) .....	18
<b>Gambar 2. 8</b> Flowchart Menu Admin .....	19
<b>Gambar 2. 9</b> Flowchart Menu Admin .....	20
<b>Gambar 2. 10</b> Flowchart Menu Makanan (Admin) .....	21
<b>Gambar 2. 11</b> Flowchart Menu Barang (Admin).....	22
<b>Gambar 2. 12</b> Flowchart Menu Voucher (Admin).....	23
<b>Gambar 3. 1</b> Registrasi User.....	101
<b>Gambar 3. 2</b> Kesalahan Registrasi User .....	102
<b>Gambar 3. 3</b> Login User.....	102
<b>Gambar 3. 4</b> Kesalahan Login User .....	102
<b>Gambar 3. 5</b> Keluar Program .....	102
<b>Gambar 3. 6</b> Menu Layanan User .....	103
<b>Gambar 3. 7</b> Order Makanan Fitur Sorting (asc) .....	103
<b>Gambar 3. 8</b> Order Makanan Fitur Sorting (desc) .....	103
<b>Gambar 3. 9</b> Order Makanan Fitur Searching .....	104
<b>Gambar 3. 10</b> Order Makanan Transaksi Order Sekarang.....	104
<b>Gambar 3. 11</b> Order Makanan Fitur Voucher .....	105
<b>Gambar 3. 12</b> Invoice order makanan .....	105
<b>Gambar 3. 13</b> Order Makanan Masukkan Ke Keranjang .....	106
<b>Gambar 3. 14</b> Order barang sorting (asc).....	106

<b>Gambar 3. 15</b> Order Barang Sorting (desc) .....	106
<b>Gambar 3. 16</b> Order Barang Fitur Searching .....	107
<b>Gambar 3. 17</b> Order Barang Beli Sekarang .....	107
<b>Gambar 3. 18</b> Order Barang Fitur Voucher .....	108
<b>Gambar 3. 19</b> Invoice Order Barang .....	108
<b>Gambar 3. 20</b> Order Barang Masukkan Ke Keranjang .....	108
<b>Gambar 3. 21</b> Cek Voucher .....	109
<b>Gambar 3. 22</b> Cek Keranjang Check Out (semua).....	109
<b>Gambar 3. 23</b> Cek Keranjang Check Out (pilih) .....	109
<b>Gambar 3. 24</b> Cek Saldo .....	110
<b>Gambar 3. 25</b> Top Up Saldo .....	110
<b>Gambar 3. 26</b> Kembali .....	110
<b>Gambar 3.27 27</b> Login sebagai Admin .....	111
<b>Gambar 3. 28</b> Menu Layanan Admin .....	111
<b>Gambar 3.2929</b> Menu Makanan.....	111
<b>Gambar 3. 30</b> Create Makanan .....	112
<b>Gambar 3. 31</b> Read Makanan.....	112
<b>Gambar 3. 32</b> Update makanan.....	113
<b>Gambar 3. 33</b> Delete Makanan .....	113
<b>Gambar 3. 34</b> Menu barang.....	114
<b>Gambar 3. 35</b> Create barang .....	114
<b>Gambar 3. 36</b> Read barang.....	114
<b>Gambar 3. 37</b> Update barang .....	115
<b>Gambar 3. 38</b> Delete barang .....	115
<b>Gambar 3. 39</b> Menu voucher.....	116
<b>Gambar 3. 40</b> Create voucher .....	116
<b>Gambar 3. 41</b> Read voucher.....	116
<b>Gambar 3. 42</b> Update voucher .....	117
<b>Gambar 3. 43</b> Delete voucher .....	117
<b>Gambar 3. 44</b> Keluar program .....	118

## **Daftar Tabel**

<b>Tabel 3.1</b> Source Code .....	53
<b>Tabel 1.1</b> Tabel Kontribusi .....	74

## **BAB I**

### **PANDAHULUAN**

#### **1.1 Deskripsi Masalah**

Seperti yang kita ketahui Python merupakan bahasa pemrograman dinamis yang mendukung pemrograman berbasis objek. Dengan bahasa phyton ini kami kaitkan dengan kemajuan teknologi yang turut mempengaruhi semua bidang kehidupan untuk menjadi semakin lebih mudah dan cepat dalam melakukan sebuah pekerjaan.

Berdasarkan fakta yang beredar bahwa, masih banyak toko yang menggunakan sistem pelayanan dengan cara yang manual. Dengan sistem tersebut terkadang menyebabkan masalah yang dapat terjadi seperti pada saat melakukan pemesanan atau transaksi pembayaran dan hal tersebut menyebabkan ketidakpuasan pelanggan terhadap sistem yang masih dilakukan dengan cara manual. Maka berdasarkan latar belakang tersebut kami merancang sebuah projek berlatar belakang “*E-Commerce*” yang disusun menggunakan bahasa dan struktur dari pemrograman phyton yang akan mempermudah suatu transaksi serta pelayanan dalam penjualan.

Dari permasalahan tersebut, kelompok kami telah merancang sebuah Program *E-Commerce* yang akan dapat membantu para UMKM dan juga konsumen. Dengan adanya program ini, para UMKM dan konsumen agar dapat melakukan pencatatan, pengelolaan, dan penghitungan transaksi secara praktis dan efisien. Selain itu, program ini juga telah kami rancang sedemikian rupa sehingga tidak hanya memudahkan para UMKM untuk melakukan pengelolaan, tetapi juga memudahkan para konsumen untuk melakukan pembelian barang.

#### **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang diuraikan dapat kami ambil rumusan masalah sebagai berikut:

1. Bagaimana merancang Program E-Commerce?
2. Bagaimana menjalankan Program E-Commerce bagi Admin?

3. Bagaimana menjalankan Program E-Commerce bagi konsumen?

### **1.3 Batasan Masalah**

Dari permasalahan yang telah dijabarkan diatas, terdapat batasan masalah dalam penyusunan laporan program ini:

1. Program aplikasi ini masih berupa program saja dan belum dapat diterapkan sepenuhnya dalam kehidupan nyata
2. Program ini dibuat menggunakan bahasa pemrograman Python

### **1.4 Tujuan**

Adapun tujuan penyusunan dan pembuatan program ini adalah:

1. Mengikuti perkembangan ke arah yang lebih maju sesuai zaman dan teknologi yang ada saat ini
2. Pencatatan transaksi yang rapi, mudah dipahami, dan tersimpan dengan baik dalam penjualan.
3. Memudahkan penjual dalam menampilkan stok minuman dan memproses pesanan yang ada
4. Memudahkan konsumen dalam melakukan pembelian pesanan yang ada
5. Menciptakan sebuah program aplikasi layanan penjualan yang berbasis system untuk mempermudah pengoperasian sebuah toko secara otomatis melalui komputer agar lebih efektif dan efisien.

### **1.5 Manfaat**

Adapun manfaat dari penyusunan dan pembuatan program ini, yaitu:

1. Mempermudah pendataan pesanan dan stok minuman yang ada
2. Mempersingkat waktu tunggu dari setiap pelanggan yang mengantre
3. Memudahkan sistem pelayanan dalam proses transaksi penjualan
4. Membantu pengelolaan dalam sebuah toko berbasis digital

## **BAB II**

## **PERANCANGAN**

### **2.1 Analisis Program**

Pada program berbasis python ini kami menggunakan beberapa library yang mendukung yakni:

#### 1. Csv (Comma-Separated Values)

Format file sederhana yang digunakan untuk dalam bentuk teks yang terstruktur dengan tabel yang digunakan sebagai tempat penyimpanan seperti menyimpan data layanan.

#### 2. Pwinput

Library ini digunakan untuk memudahkan penggunaan masukan berbentuk kata sandi (password) dengan tampilan yang lebih aman.

Cara kerja: pwinput akan menyembunyikan masukan password yang diketik oleh pengguna sehingga tidak terlihat dalam teks biasa di layar. Ini meningkatkan keamanan karena kata sandi tidak akan terbaca oleh mata orang lain yang melihat layar. Pengguna akan diminta untuk memasukkan kata sandi, dan nilai yang dimasukkan akan disaring untuk keamanan. Contoh penggunaan: password = pwinput("Masukkan kata sandi: ")

#### 3. Pretty table

Library dalam python untuk mengeluarkan data seperti kode, nama layanan, harga dan deskripsi layanandalam bentuk tabel agar memiliki tampilan yang rapid an mudah dibaca oleh user.

#### 4. If, Elif Else

If digunakan untuk membuat percabangan dan menguji suatu kondisi dan kemudian mengambil keputusan, elif digunakan untuk menguji kondisi kedua

jika tidak memenuhi kondisi if. Dan else digunakan jika kondisi if dan elif tidak terpenuhi.

#### 5. def

Kata kunci yang digunakan untuk mendefinisikan atau membuat fungsi. Fungsi sendiri adalah blok kode yang dapat dipanggil secara berulang untuk melakukan tugas tertentu. Seperti def login () yang berisikan kode login yang saat dipanggil nanti akan menjalankan blok kode login.

#### 6. Loop “While True”

Loop “While True” digunakan dengan tujuan agar saat user salah meng-input atau tidak sesuai dengan ketentuan, maka akan terjadi perulangan. “While True” biasanya diakhiri dengan “break” untuk mengakhiri perulangan apabila diinginkan.

#### 7. Try Except

Digunakan untuk mengatasi bug, contohnya saat kita meng-input huruf pada bagan yang telah dideklarasikan dengan “int”.

#### 8. Date time

Digunakan untuk mengelola data tanggal dan waktu dalam membuat struk belanja.

#### 9. Sorting

Sorting (pengurutan) adalah proses mengatur data dalam suatu urutan tertentu, seperti urutan naik (ascending) atau turun (descending). Fungsi utama dari sorting adalah untuk memudahkan pencarian, analisis, dan pengelolaan data, terutama saat kita bekerja dengan dataset besar atau data yang perlu diurutkan untuk tujuan tertentu.

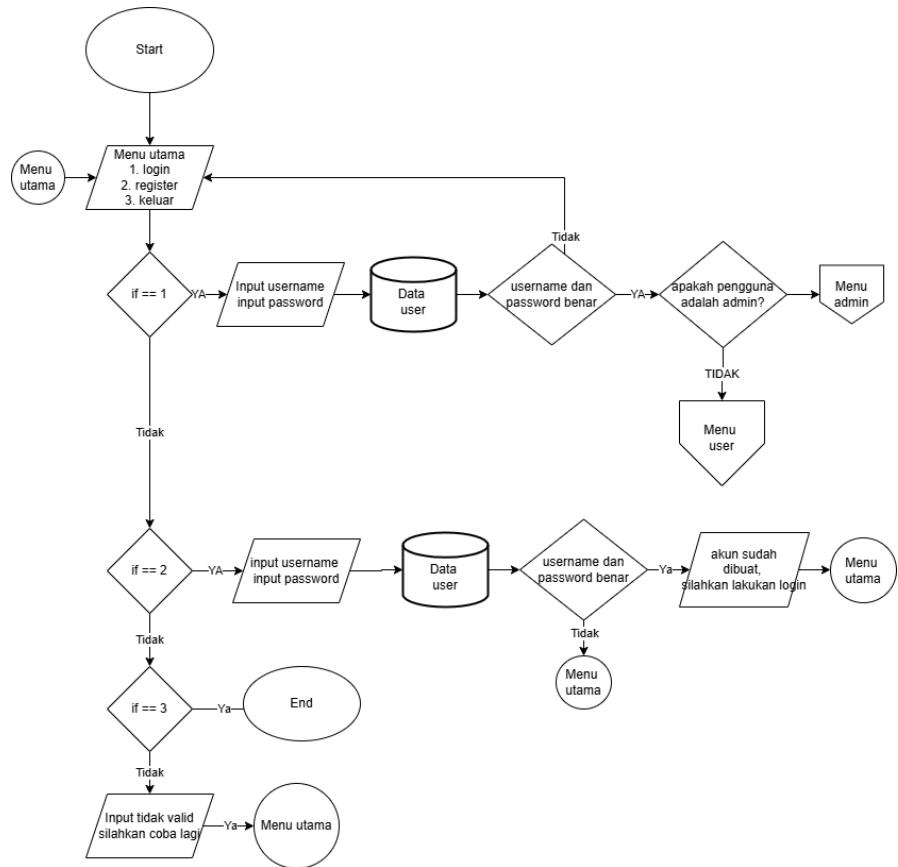
## 10. Searching

Searching (pencarian) adalah proses menemukan data tertentu dalam suatu kumpulan data atau struktur data berdasarkan kriteria yang diberikan. Fungsi utama searching adalah untuk mendapatkan informasi yang spesifik secara cepat dan efisien dari kumpulan data yang besar. Searching digunakan dalam berbagai aplikasi seperti pencarian teks, data dalam database, maupun pencarian file.

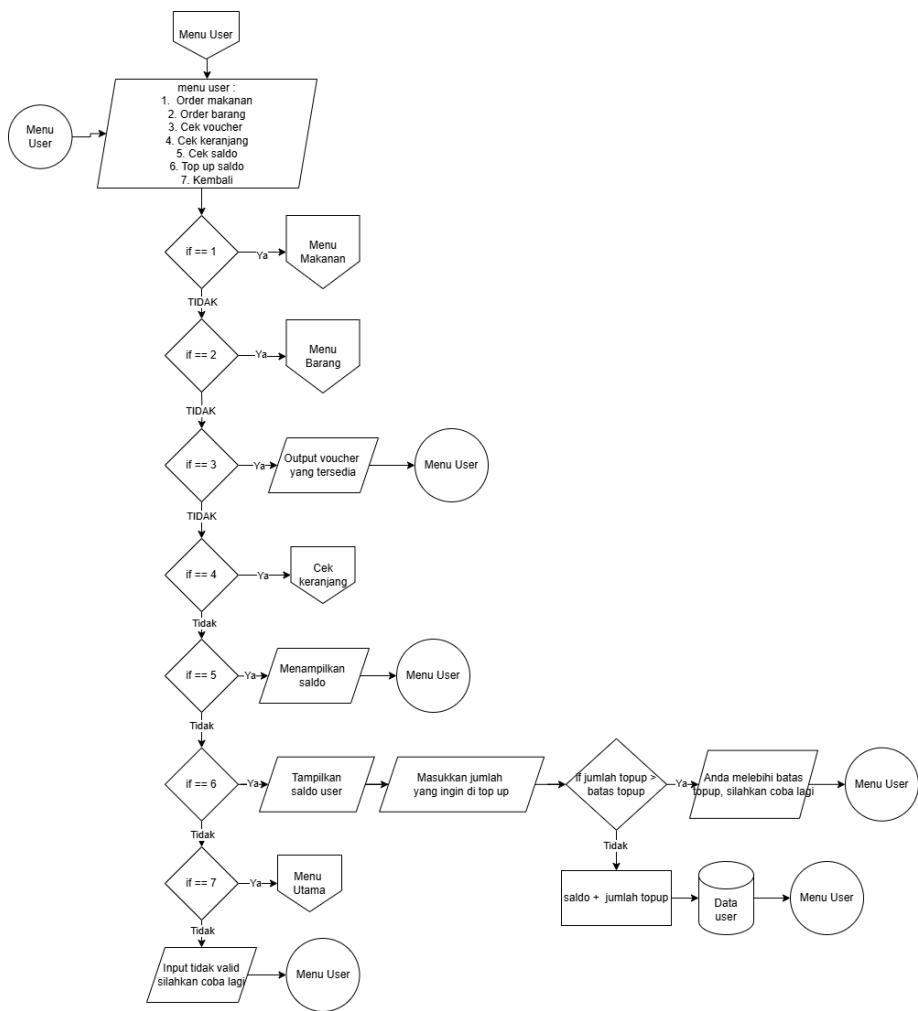
## 11. CRUD (Create, Read, Update, Delete)

CRUD adalah singkatan dari Create, Read, Update, dan Delete, yaitu empat operasi dasar yang dilakukan pada data dalam aplikasi atau sistem manajemen database. CRUD adalah konsep inti dalam pengembangan aplikasi yang melibatkan pengelolaan data, seperti aplikasi berbasis web, sistem informasi, dan database.

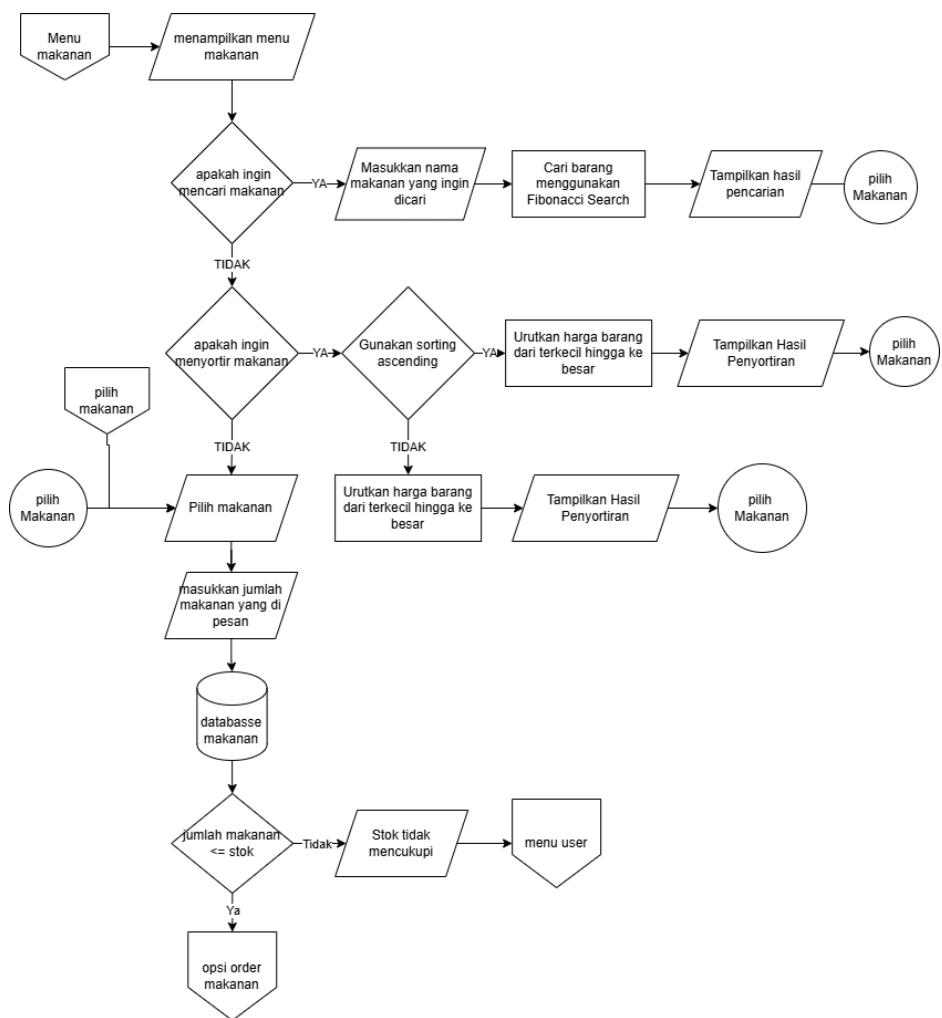
## 2.2 Flowchart



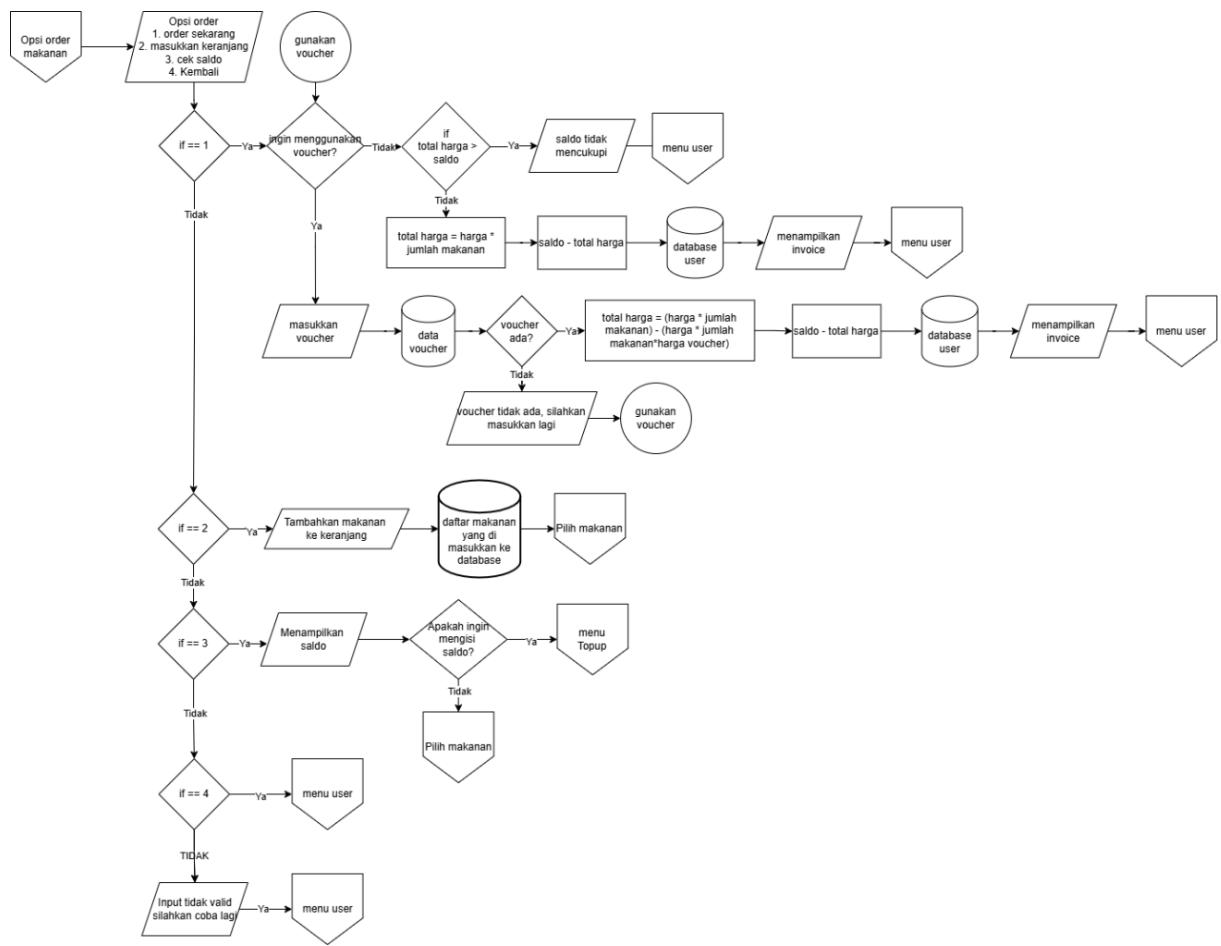
**Gambar 2. 1** Flowchart Login



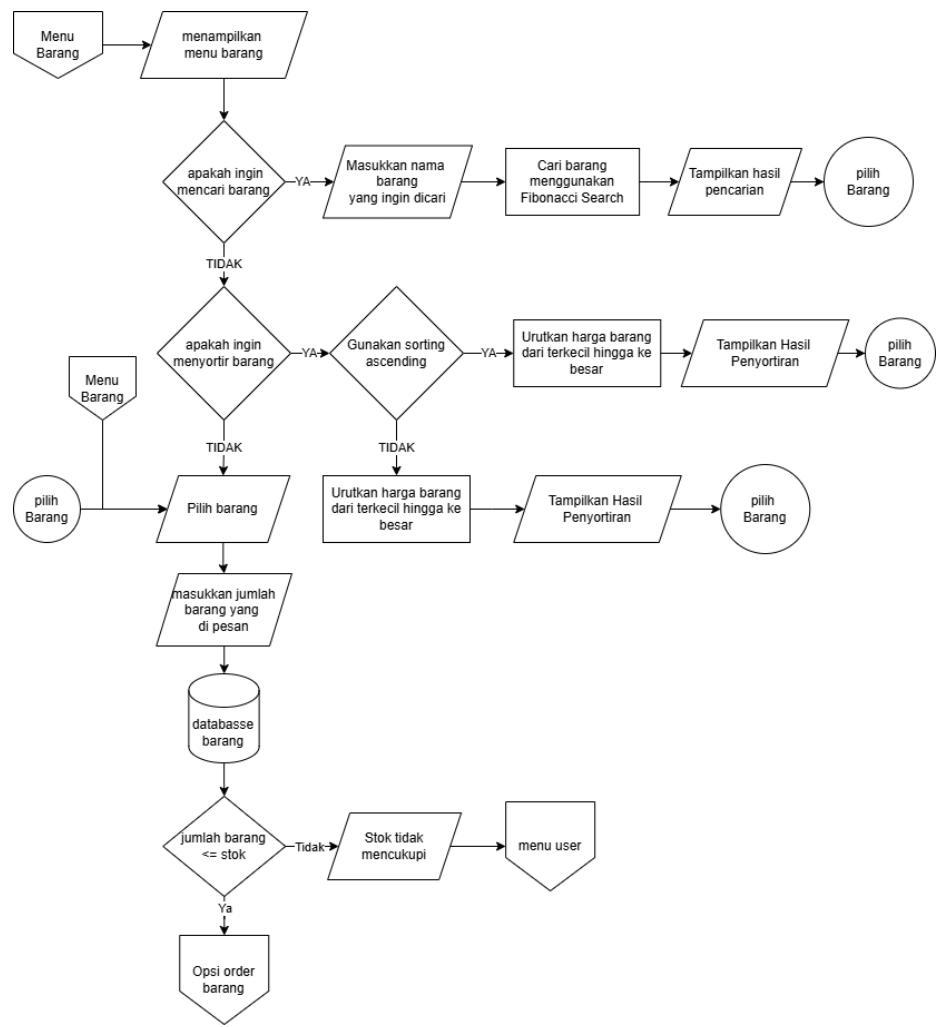
**Gambar 2. 2 Flowchart Menu User**



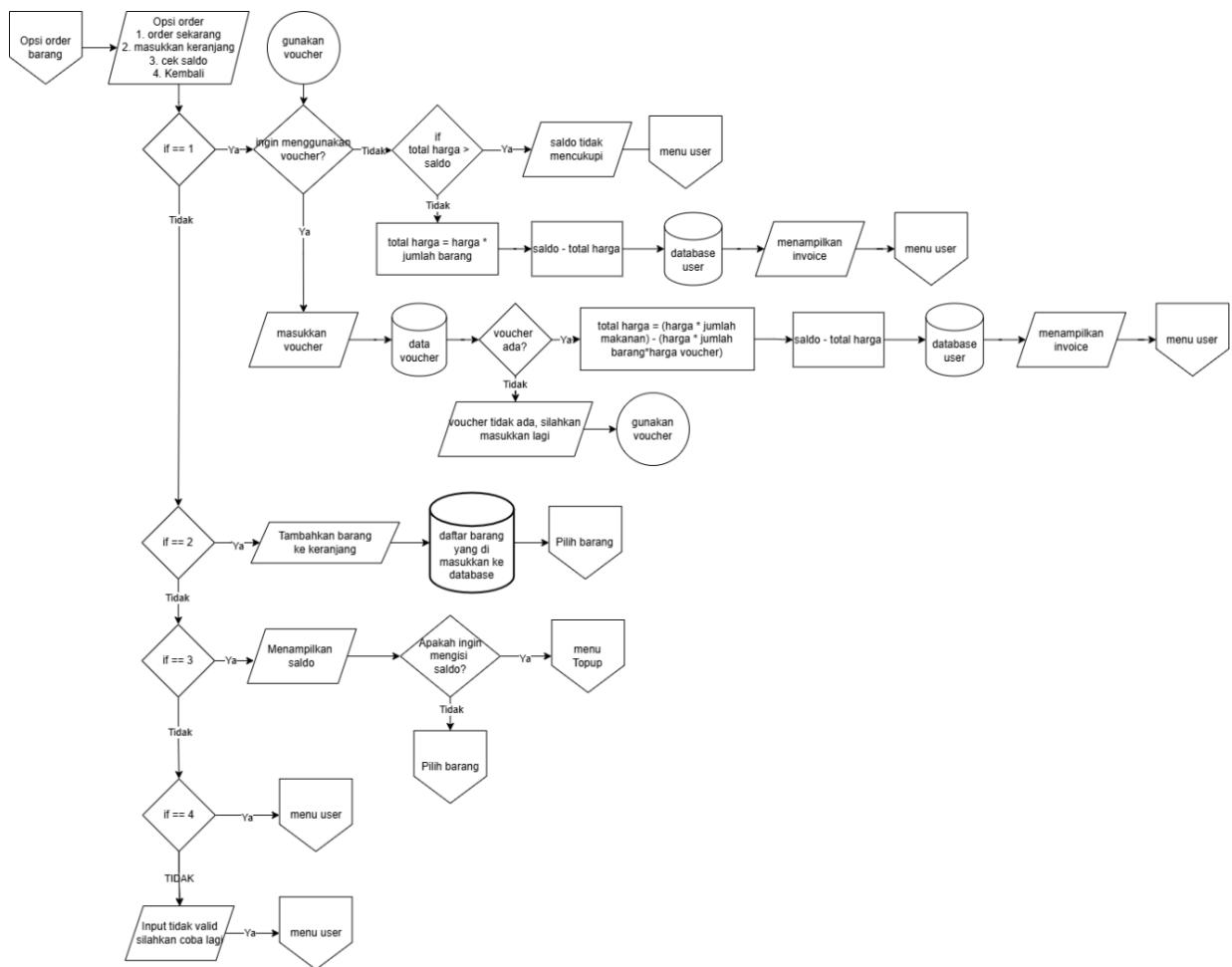
**Gambar 2. 3** Flowchart Menu Makanan (User)



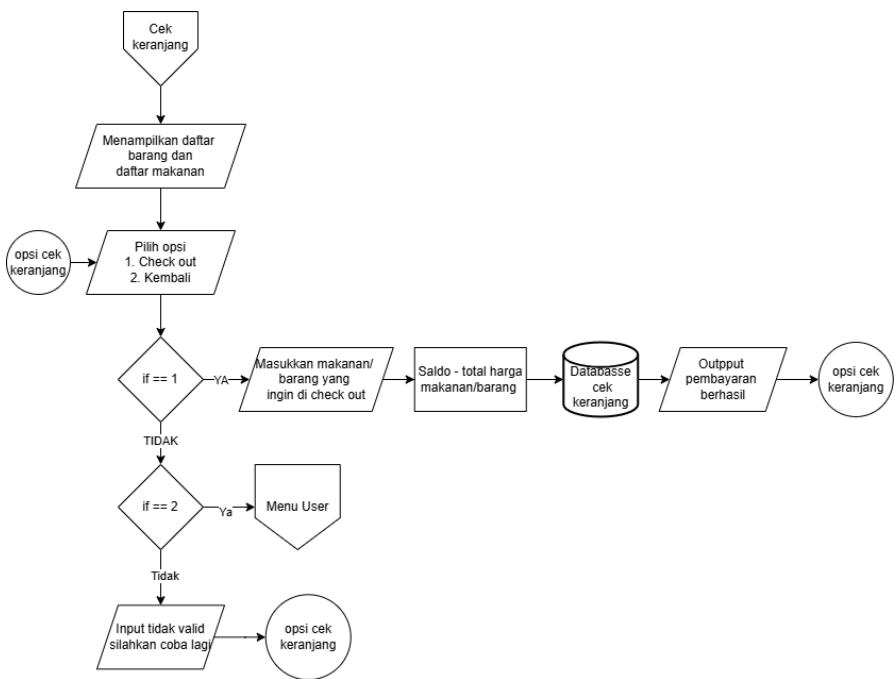
**Gambar 2. 4** Flowchart Opsi Order Makanan (User)



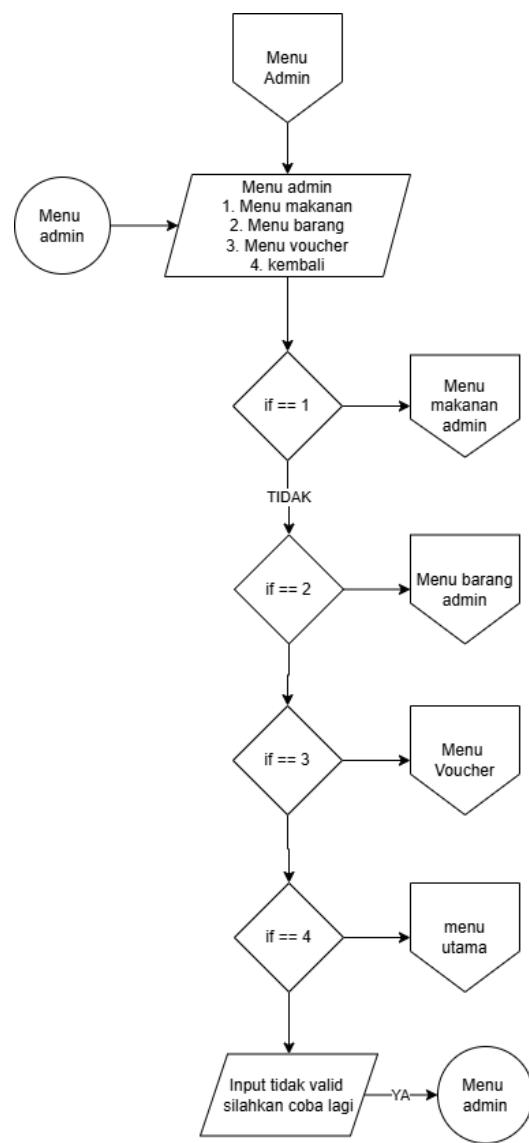
**Gambar 2. 5 Flowchart Menu Barang (User)**



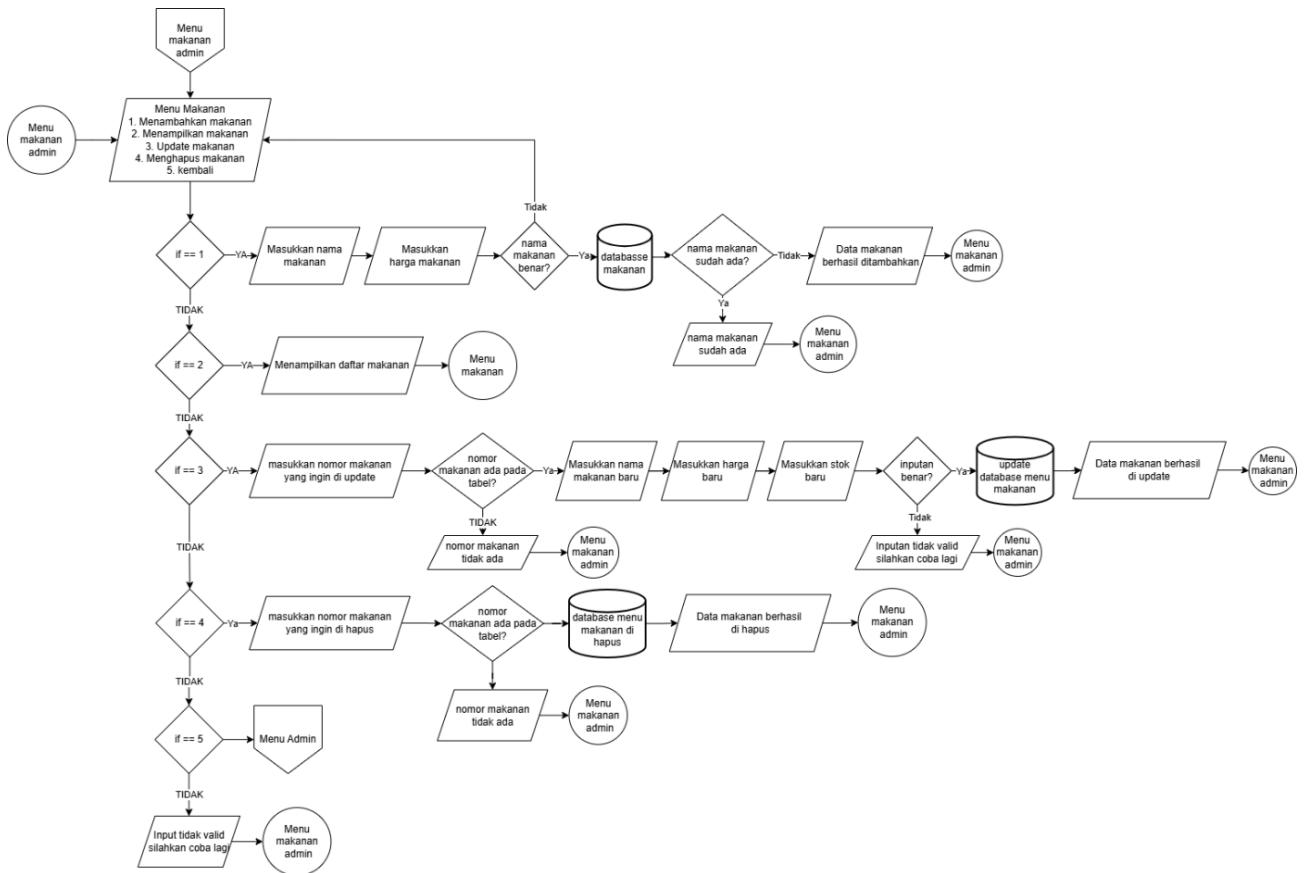
**Gambar 2. 6 Flowchart Opsi Order Barang (User)**



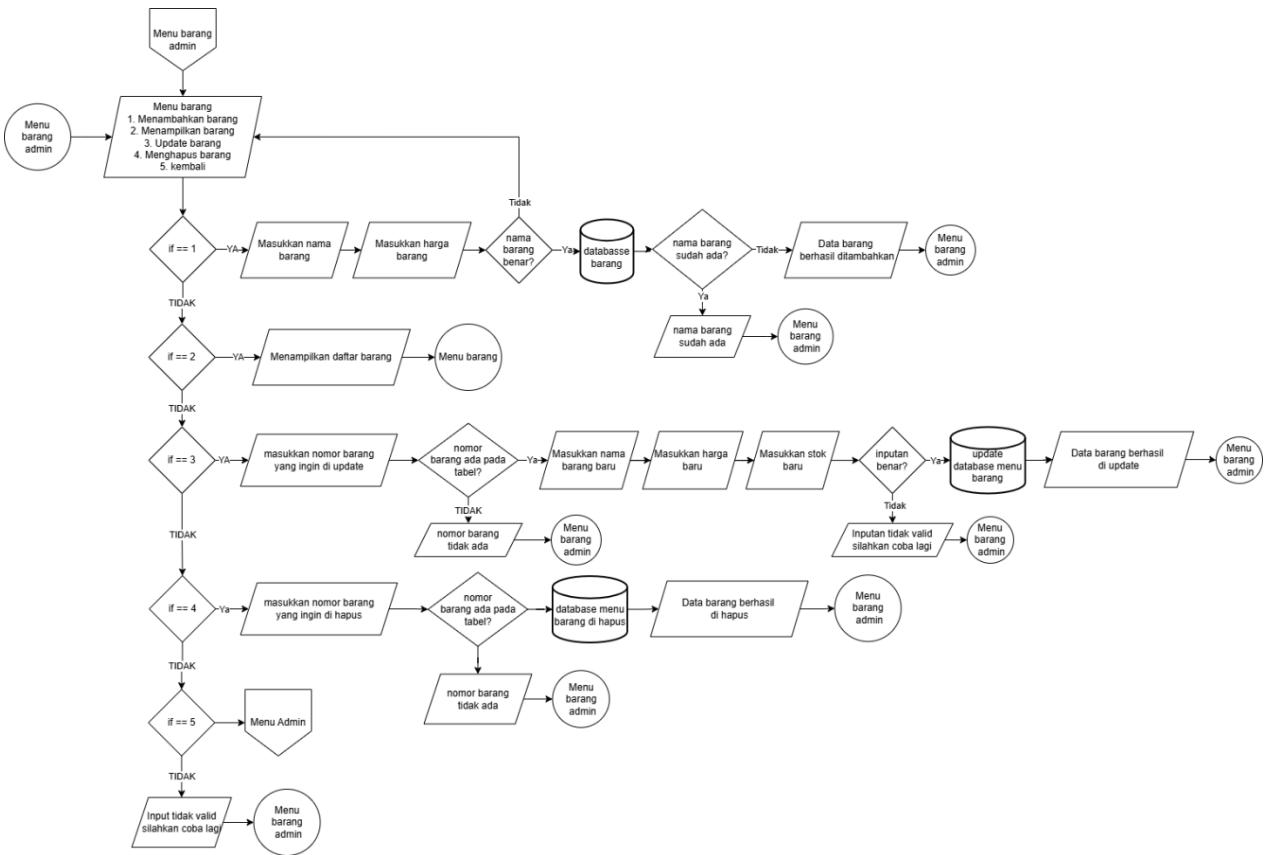
**Gambar 2. 7 Flowchart Cek Keranjang (User)**



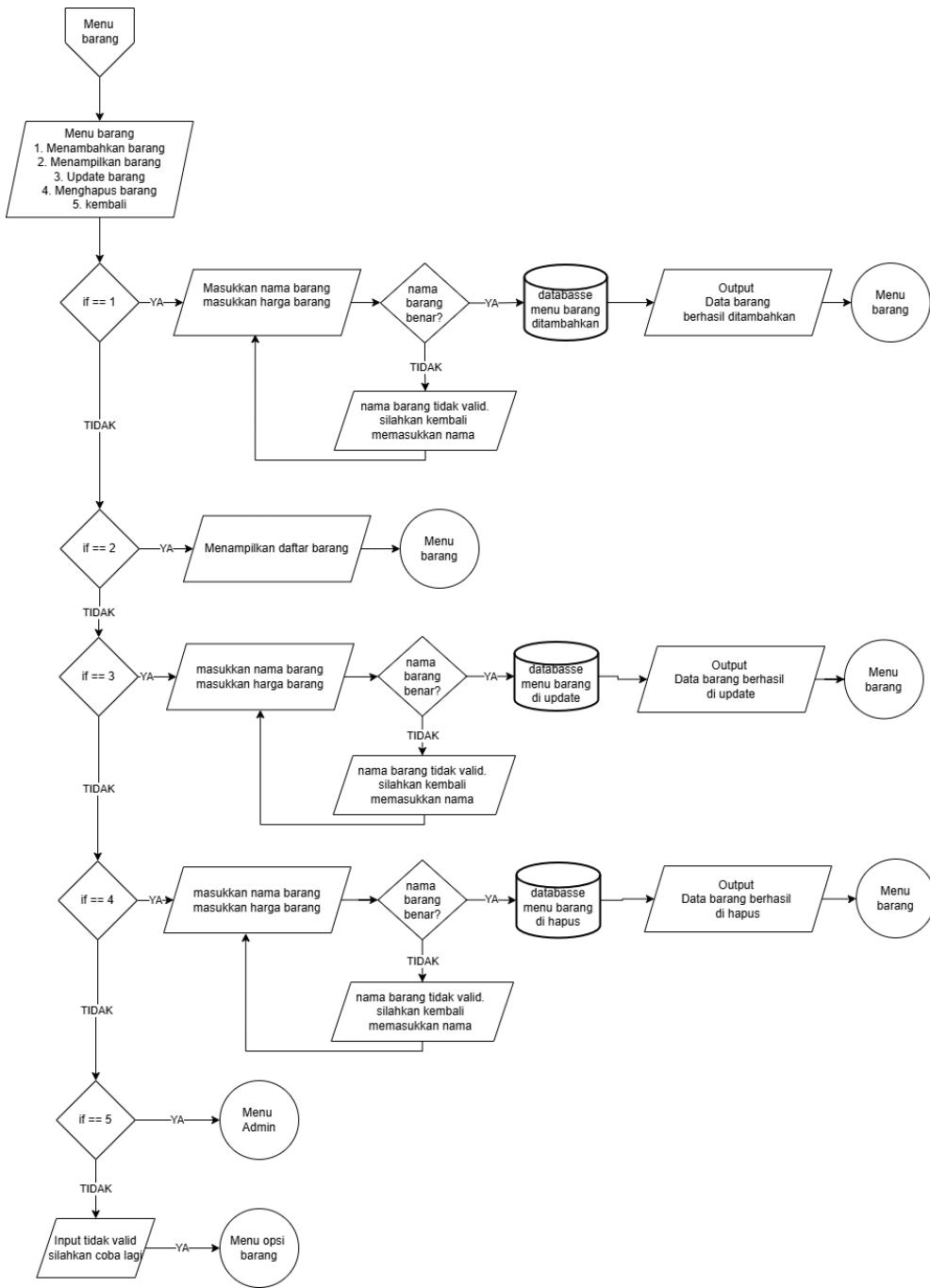
**Gambar 2. 8** Flowchart Menu Admin



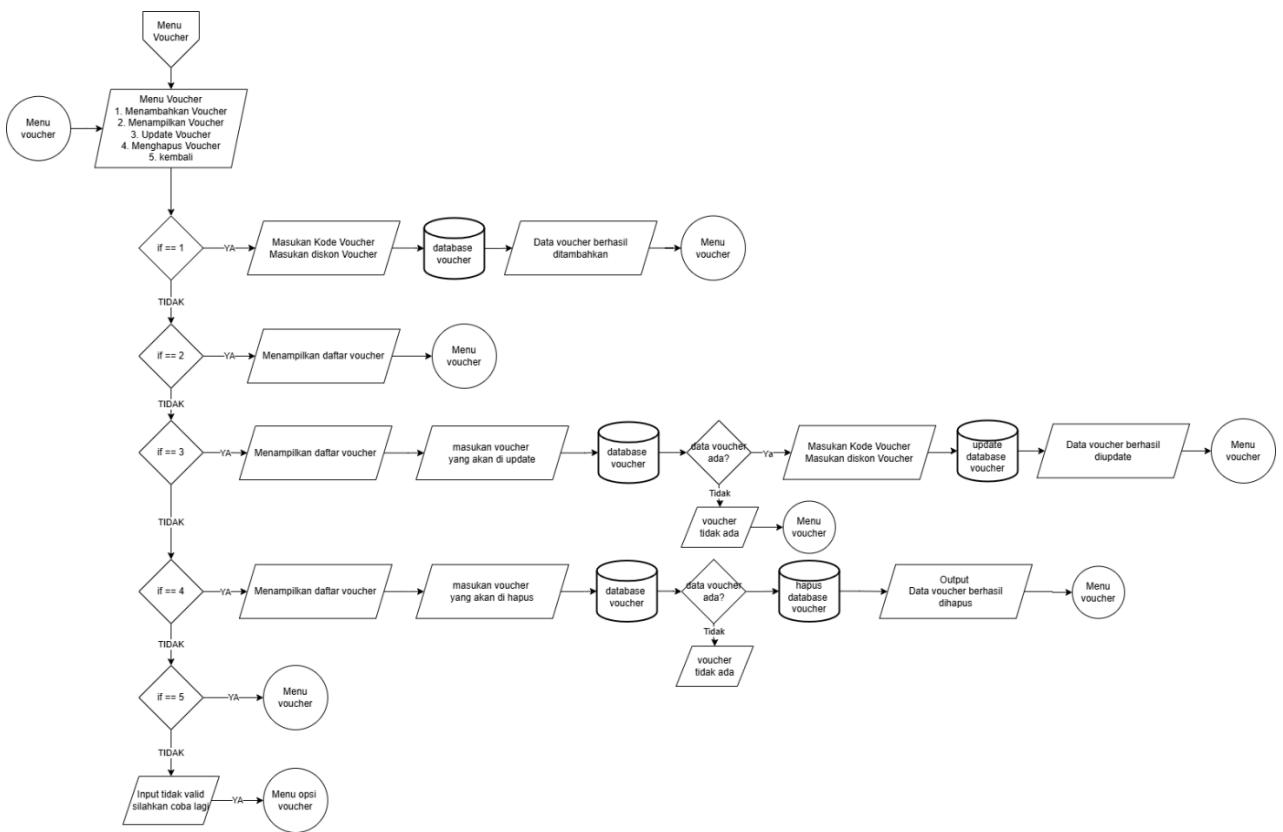
**Gambar 2.9** Flowchart Menu Admin



**Gambar 2. 10** Flowchart Menu Makanan (Admin)



**Gambar 2. 11 Flowchart Menu Barang (Admin)**



**Gambar 2. 12** Flowchart Menu Voucher (Admin)

## BAB III

### HASIL DAN PEMBAHASAN

#### 3.1 Implementasi Program

Dalam perancangan program ini kami mengimplementasikan semua modul-modul materi yang telah dipelajari dalam pratikum.

##### 1. Csv

```
# Fungsi untuk menyimpan data keranjang ke CSV
def simpan_keranjang_ke_csv(keranjang):
    with open("keranjang.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["username", "nama_item", "total_harga", "status"])
        writer.writeheader()
        for item in keranjang:
            writer.writerow(item)

def save_keranjang(username, keranjang):
    existing_data = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            existing_data = list(reader)
    except FileNotFoundError:
        pass

    for item in keranjang:
        existing_data.append({
            "username": username,
            "nama_item": item["nama_item"],
            "total_harga": item["total_harga"],
            "status": item["status"]
        })

def load_keranjang(username):
    keranjang = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            for row in reader:
                if row['username'] == username:
                    keranjang.append({
                        "nama_item": row['nama_item'],
                        "total_harga": row['total_harga'],
                        "status": row['status']
                    })
    except FileNotFoundError:
        print("File keranjang tidak ditemukan.")
    return keranjang

def simpan_keranjang_ke_csv(existing_data):
    with open("keranjang.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["username", "nama_item", "total_harga", "status"])
        writer.writeheader()
        for item in existing_data:
            writer.writerow(item)

with open('keranjang.csv', mode='w', newline='') as file:
    fieldnames = ['username', 'nama_item', 'total_harga', 'status']
    writer = csv.DictWriter(file, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerows(existing_data)
```

```

# Fungsi untuk menyimpan data keranjang ke CSV
def simpan_keranjang_ke_csv(keranjang):
    with open("keranjang.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["username", "nama_item", "total_harga", "status"])
        writer.writeheader()
        for item in keranjang:
            writer.writerow(item)

def save_keranjang(username, keranjang):
    existing_data = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            existing_data = list(reader)
    except FileNotFoundError:
        pass

    for item in keranjang:
        existing_data.append({
            "username": username,
            "nama_item": item["nama_item"],
            "total_harga": item["total_harga"],
            "status": item["status"]
        })

# Fungsi untuk memuat data barang
def load_data_barang():
    barang = []
    with open("stok_barang.csv", mode="r") as file:
        reader = csv.DictReader(file)
        for row in reader:
            row["no"] = int(row["no"])
            row["harga"] = int(row["harga"])
            row["stok"] = int(row["stok"])
            barang.append(row)
    return barang

# Fungsi untuk menyimpan data barang ke CSV
def simpan_barang_ke_csv(data_barang):
    with open("stok_barang.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "nama_barang", "harga", "stok"])
        writer.writeheader()
        for barang in data_barang:
            writer.writerow(barang)

def load_keranjang(username):
    keranjang = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            for row in reader:
                if row['username'] == username:
                    keranjang.append({
                        "nama_item": row['nama_item'],
                        "total_harga": row['total_harga'],
                        "status": row['status']
                    })
    except FileNotFoundError:
        print("File keranjang tidak ditemukan.")
    return keranjang

def simpan_keranjang_ke_csv(existing_data):
    with open("keranjang.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["username", "nama_item", "total_harga", "status"])
        writer.writeheader()
        for item in existing_data:
            writer.writerow(item)

```

```

def update_voucher_status(voucher_used):
    vouchers = load_data_voucher()
    for voucher in vouchers:
        if voucher['kode'] == voucher_used['kode']:
            voucher['used_by'] = voucher_used['used_by']

    # Menyimpan kembali ke file CSV
    with open("voucher.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "kode", "diskon", "used_by"])
        writer.writeheader()
        writer.writerows(vouchers)

def simpan_voucher_ke_csv(data_vouchers):
    with open("voucher.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "kode", "diskon", "used_by"])
        writer.writeheader()
        for voucher in data_vouchers:
            writer.writerow(voucher)

# Fungsi untuk memuat data user di file csv
def data_user():
    users = []
    with open("user.csv", mode="r") as file:
        reader = csv.DictReader(file)
        for row in reader:
            row["saldo"] = int(float(row["saldo"])) # Convert saldo to integer
            users.append(row)
    return users

# Fungsi untuk menyimpan data user ke file csv
def simpan_data_user_ke_csv(data_users):
    with open("user.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["username", "password", "saldo"])
        writer.writeheader()
        for user in data_users:
            user["saldo"] = int(user["saldo"])
        writer.writerow(user)

# Fungsi untuk memuat data makanan
def load_data_makanan():
    makanan = []
    with open("stok_makanan.csv", mode="r") as file:
        reader = csv.DictReader(file)
        for row in reader:
            row["no"] = int(row["no"])
            row["harga"] = int(row["harga"])
            row["stok"] = int(row["stok"])
            makanan.append(row)
    return makanan

# Fungsi untuk menyimpan data makanan ke CSV
def simpan_makanan_ke_csv(data_makanan):
    with open("stok_makanan.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "nama_makanan", "harga", "stok"])
        writer.writeheader()
        for makanan in data_makanan:
            writer.writerow(makanan)

```

```

# Fungsi untuk memuat data makanan
def load_data_makanan():
    makanan = []
    with open("stok_makanan.csv", mode="r") as file:
        reader = csv.DictReader(file)
        for row in reader:
            row["no"] = int(row["no"])
            row["harga"] = int(row["harga"])
            row["stok"] = int(row["stok"])
            makanan.append(row)
    return makanan

# Fungsi untuk menyimpan data makanan ke CSV
def simpan_makanan_ke_csv(data_makanan):
    with open("stok_makanan.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "nama_makanan", "harga", "stok"])
        writer.writeheader()
        for makanan in data_makanan:
            writer.writerow(makanan)

```

Program ini menggunakan file CSV untuk menyimpan dan memuat data pengguna, makanan, barang, dan voucher. Dengan menggunakan modul CSV, kami dapat membaca dan menulis data ke dalam format yang mudah diakses dan dikelola.

## 2. isdigit

```

# Fungsi untuk top-up saldo
def top_up(username):
    cek_saldo = data_user()
    for user in cek_saldo:
        if user["username"] == username:
            print(f"Saldo anda saat ini adalah = Rp {user['saldo']}")
            dana = input("Berapakah yang ingin anda topup? : Rp")
            if dana.isdigit():
                dana = int(dana)
                if dana > 2000000:
                    print("Top up melampaui batas maksimal, batas maksimal : Rp 2.000.000")
                else:
                    user["saldo"] = int(user["saldo"]) + dana # Pastikan saldo tetap sebagai integer
                    simpan_data_user_ke_csv(cek_saldo)
                    print(f"Saldo anda telah ditambahkan. Saldo sekarang = Rp {user['saldo']}")
            else:
                print("Masukkan nominal yang valid.")
    break # Keluar dari loop setelah memproses pengguna

```

```

# Fungsi untuk mengupdate data barang
def update_barang():
    lihat_barang()
    data_barang = load_data_barang()

    try:
        item_no = int(input("Masukkan nomor barang yang ingin diupdate: "))
        item_to_update = next((barang for barang in data_barang if barang["no"] == item_no), None)
        if not item_to_update:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            return
        nama_baru_barang = input(f"Masukkan nama baru untuk {item_to_update['nama_barang']} : ")
        harga_baru_barang = input(f"Masukkan harga baru untuk {item_to_update['nama_barang']} : ")
        stok_baru_barang = input(f"Masukkan stok baru untuk {item_to_update['nama_barang']} : ")

        if nama_baru_barang:
            item_to_update["nama_barang"] = nama_baru_barang
        if harga_baru_barang.isdigit():
            item_to_update["harga"] = int(harga_baru_barang)
        if stok_baru_barang.isdigit():
            item_to_update["stok"] = int(stok_baru_barang)

        simpan_barang_ke_csv(data_barang)
        print("Barang berhasil diperbarui.")
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
    return update_barang()

def update_barang():
    lihat_barang()
    data_barang = load_data_barang()

    try:
        item_no = int(input("Masukkan nomor barang yang ingin diupdate: "))
        item_to_update = next((barang for barang in data_barang if barang["no"] == item_no), None)
        if not item_to_update:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            return
        nama_baru_barang = input(f"Masukkan nama baru untuk {item_to_update['nama_barang']} : ")
        harga_baru_barang = input(f"Masukkan harga baru untuk {item_to_update['nama_barang']} : ")
        stok_baru_barang = input(f"Masukkan stok baru untuk {item_to_update['nama_barang']} : ")

        if nama_baru_barang:
            item_to_update["nama_barang"] = nama_baru_barang
        if harga_baru_barang.isdigit():
            item_to_update["harga"] = int(harga_baru_barang)
        if stok_baru_barang.isdigit():
            item_to_update["stok"] = int(stok_baru_barang)

        simpan_barang_ke_csv(data_barang)
        print("Barang berhasil diperbarui.")
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
    return update_barang()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return update_barang()

```

```

def tambah_voucher():
    while True:
        data_vouchers = load_data_voucher()
        try:
            kode = input("Masukkan kode voucher: ")
            diskon = input("Masukkan besaran diskon (dalam persen): ")

            if not diskon.isdigit():
                print("Diskon harus berupa angka.")
                return

            diskon = int(diskon)
            no = max([voucher["no"] for voucher in data_vouchers], default=0) + 1
            voucher_baru = {"no": no, "kode": kode, "diskon": diskon}
            data_vouchers.append(voucher_baru)

            simpan_voucher_ke_csv(data_vouchers)
            print("Voucher baru berhasil ditambahkan.")
            break # Keluar dari loop setelah berhasil menambah voucher
        except ValueError:
            print("Input tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except EOFError:
            print("\nInput tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return # Keluar dari fungsi jika ada interupsi

```

```

def update_barang():
    lihat_barang()
    data_barang = load_data_barang()

    try:
        item_no = int(input("Masukkan nomor barang yang ingin diupdate: "))
        item_to_update = next((barang for barang in data_barang if barang["no"] == item_no), None)
        if not item_to_update:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            return
        nama_baru_barang = input(f"Masukkan nama baru untuk {item_to_update['nama_barang']} : ")
        harga_baru_barang = input(f"Masukkan harga baru untuk {item_to_update['nama_barang']} : ")
        stok_baru_barang = input(f"Masukkan stok baru untuk {item_to_update['nama_barang']} : ")

        if nama_baru_barang:
            item_to_update["nama_barang"] = nama_baru_barang
        if harga_baru_barang.isdigit():
            item_to_update["harga"] = int(harga_baru_barang)
        if stok_baru_barang.isdigit():
            item_to_update["stok"] = int(stok_baru_barang)

        simpan_barang_ke_csv(data_barang)
        print("Barang berhasil diperbarui.")
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
        return update_barang()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return update_barang()

```

isdigit() digunakan dalam berbagai fungsi untuk memastikan bahwa input yang diterima dari pengguna adalah angka sebelum melakukan konversi atau operasi lebih lanjut.

### 3. Pretty table

```
def menu_user(username):
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang di BELIBELI"]
        tabel.add_row(["[1] Order Makanan\n[2] Order Barang\n[3] Cek Voucher\n[4] Cek Keranjang\n[5]"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda : "))
            if opsi == 1:
                order_makanan(username)
            elif opsi == 2:
                order_barang(username)
            elif opsi == 3:
                lihat_voucher()
            elif opsi == 4:
                cek_keranjang(username)
            elif opsi == 5:
                cek_saldo(username)
            elif opsi == 6:
                top_up(username)
            elif opsi == 7:
                break
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
    return menu_user(username)

def lihat_voucher():
    data_vouchers = load_data_voucher()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Kode Voucher", "Diskon (%)"]

    for voucher in data_vouchers:
        tabel.add_row([voucher["no"], voucher["kode"], voucher["diskon"]])
    print(tabel)

def lihat_barang():
    data_barang = load_data_barang()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]

    for barang in data_barang:
        tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"], barang["stok"]])
    print(tabel)
```

```

def invoice_makanan(username, nama_makanan, jumlah_makanan, total_harga):
    struk = PrettyTable()
    struk.field_names = ["Invoice"]
    struk.add_row([f"Nama Customer : {username}")]
    struk.add_row([f"Tanggal Transaksi : {datetime.now().strftime("%d-%m-%y %H:%M:%S")}]")
    struk.add_row([f"Nama Produk : {nama_makanan}"])
    struk.add_row([f"Jumlah Item : {jumlah_makanan}"])
    struk.add_row([f"Total Bayar : {total_harga}"])
    struk.add_row(["Terima kasih sudah bertransaksi"])
    print(struk)

# Fungsi untuk menampilkan Invoice makanan
def invoice_barang(username, nama_barang, jumlah_barang, total_harga):
    struk = PrettyTable()
    struk.field_names = ["Invoice"]
    struk.add_row([f"Nama Customer : {username}")]
    struk.add_row([f"Tanggal Transaksi : {datetime.now().strftime("%d-%m-%y %H:%M:%S")}]")
    struk.add_row([f"Nama Produk : {nama_barang}"])
    struk.add_row([f"Jumlah Item : {jumlah_barang}"])
    struk.add_row([f"Total Bayar : {total_harga}"])
    struk.add_row(["Terima kasih sudah bertransaksi"])
    print(struk)

def menu_admin():
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang admin"]
        tabel.add_row(["[1] Menu Makanan\n[2] Menu Barang\n[3] Menu Voucher\n[4] Keluar Program\n"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda: "))
            if opsi == 1:
                menu_makanan_admin()
            elif opsi == 2:
                menu_barang_admin()
            elif opsi == 3:
                menu_voucher()
            elif opsi == 4:
                print("Program berakhir")
                login()
                break
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            return menu_admin()
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            return menu_admin()
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return menu_admin()

```

Digunakan untuk menampilkan data dalam format tabel yang rapi di terminal

#### 4. If, Elif, Else

```
elif opsi == 2:
    username2 = input("Masukkan username anda : ")
    password2 = pwinput.pwinput("Masukkan Password anda : ")
    user_sudah_ada = data_user()
    login_success = False

    if username2 == "admin" and password2 == "admin123":
        print("== Login berhasil, Selamat datang Admin ==")
        menu_admin()
        login_success = True

    else:
        for user in user_sudah_ada:
            if user["username"] == username2 and user["password"] == password2:
                print(f"== Login berhasil, Selamat datang {username2}, selamat")
                login_success = True
                menu_user(username2)
                break

        if not login_success:
            print("== Akun tidak ada atau password salah ===")

elif opsi == 3:
    print("== Terima kasih telah menggunakan BeliBeli! ==")
    break # Keluar dari fungsi login

else:
    print("Opsi tidak ada")

def login():
    while True:
        print("== Selamat datang di BeliBeli ==")
        print("[1] Register")
        print("[2] Login")
        print("[3] Keluar")
        try:
            opsi = int(input("Masukkan pilihan anda : "))

            if opsi == 1:
                username1 = input("Masukkan username anda : ")
                if not username1:
                    print("Username tidak boleh kosong")
                    continue

                password1 = input("Masukkan Password anda : ")
                if not password1:
                    print("Password tidak boleh kosong")
                    continue

                user_sudah_ada = data_user()
                if any(user["username"] == username1 for user in user_sudah_ada):
                    print("Username sudah ada, masukkan username lain")
                    continue

                # Menambahkan pengguna baru
                tambah_user_baru = {"username": username1, "password": password1, "saldo": "0"}
                user_sudah_ada.append(tambah_user_baru)
                simpan_data_user_ke_csv(user_sudah_ada)
```

```

def menu_user(username):
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang di BELIBELI"]
        tabel.add_row(["[1] Order Makanan\n[2] Order Barang\n[3] Cek Voucher\n[4] Cek Keranjang"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda : "))
            if opsi == 1:
                order_makanan(username)
            elif opsi == 2:
                order_barang(username)
            elif opsi == 3:
                lihat_voucher()
            elif opsi == 4:
                cek_keranjang(username)
            elif opsi == 5:
                cek_saldo(username)
            elif opsi == 6:
                top_up(username)
            elif opsi == 7:
                break
            else:
                print('input tidak valid')
                menu_user(username)
        except ValueError:
            print(" input harus berupa angka. Silakan coba lagi.")
            return menu_user(username)

def menu_admin():
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang admin"]
        tabel.add_row(["[1] Menu Makanan\n[2] Menu Barang\n[3] Menu Voucher\n[4] Keluar Program\n"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda: "))
            if opsi == 1:
                menu_makanan_admin()
            elif opsi == 2:
                menu_barang_admin()
            elif opsi == 3:
                menu_voucher()
            elif opsi == 4:
                print("Program berakhir")
                login()
                break
            else:
                print("Input harus berupa angka. Silakan coba lagi.")
                return menu_admin()
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            return menu_admin()
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            return menu_admin()

```

```

def merge_sort(data, key, reverse=False):
    if len(data) <= 1:
        return data

    mid = len(data) // 2
    left_half = merge_sort(data[:mid], key, reverse)
    right_half = merge_sort(data[mid:], key, reverse)

    return merge(left_half, right_half, key, reverse)

def merge(left, right, key, reverse):
    sorted_list = []
    while left and right:
        if (left[0][key] <= right[0][key] and not reverse) or (left[0][key] > right[0][key] and reverse):
            sorted_list.append(left.pop(0))
        else:
            sorted_list.append(right.pop(0))

```

```

def search_item(data, keyword):
    results = [
        item for item in data
        if (item.get("nama_makanan") and keyword.lower() in item["nama_makanan"].lower()) or
           (item.get("nama_barang") and keyword.lower() in item["nama_barang"].lower())
    ]
    return results

```

```

def order_makanan(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_makanan()
        saldo = user["saldo"]
        data_makanan = load_data_makanan() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan makanan berdasarkan harga? (ya/tidak): ").strip()
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
        for makanan in data_makanan:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)

```

```

search_option = input("Apakah Anda ingin mencari makanan? (ya/tidak): ").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_makanan = search_item(data_barang, keyword)

    # Menampilkan hasil pencarian
    if data_makanan:
        print("Hasil pencarian makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama makanan", "Harga", "Stok"]
        for barang in data_barang:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)
    else:
        print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
        continue # Kembali ke awal loop untuk mencoba lagi

    # Memilih item untuk dipesan
    while True:
        try:
            item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
            item_to_order = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)

            if not item_to_order:
                print("Barang dengan nomor tersebut tidak ditemukan.")
                continue # Kembali ke awal loop untuk mencoba lagi

            if item_to_order["stok"] <= 0:
                print("Maaf, stok tidak tersedia untuk item ini.")
                continue # Kembali ke awal loop untuk mencoba lagi

            quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
            if quantity > item_to_order["stok"]:
                print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
                continue # Kembali ke awal loop untuk mencoba lagi

            # Pilihan untuk langsung beli atau masukkan ke keranjang
            action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")
            if action == '1':
                # Proses pembelian langsung
                total_harga = item_to_order["harga"] * quantity

                # Menggunakan voucher
                use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()
                if use_voucher == 'ya':
                    lihat_voucher()
                    kode_voucher = input("Masukkan kode voucher: ")
                    diskon, voucher = gunakan_voucher(kode_voucher)
                    if diskon > 0:
                        a = total_harga
                        b = diskon
                        total_harga_setelah_diskon = a - (a * b / 100)
                        total_harga = total_harga_setelah_diskon
                        print(f"Voucher digunakan! Diskon: {diskon}%, Total harga setelah diskon: {total_harga:.2f}")
                        voucher['used_by'] = username # Update voucher status

```

```

        else:
            print("Voucher tidak valid atau sudah digunakan.")

        if saldo < total_harga:
            print("Saldo Anda tidak cukup untuk membeli jumlah yang diminta.")
            continue # Kembali ke awal loop untuk mencoba lagi

        # Memperbarui saldo dan stok
        user["saldo"] -= total_harga
        item_to_order["stok"] -= quantity
        simpan_data_user_ke_csv(pengguna)
        simpan_makanan_ke_csv(data_makanan)

        print(f"Pembelian berhasil! Saldo Anda sekarang: Rp {user['saldo']:.2f}")

        invoice_makanan(username, item_to_order["nama_makanan"], quantity, total_harga)

elif action == '2':
    # Menambahkan makanan ke keranjang
    tambah_ke_keranjang(username, item_to_order["nama_makanan"], item_to_order["harga"], quantity)
    print(f"{quantity} {item_to_order['nama_makanan']} telah ditambahkan ke keranjang.")
else:
    print("Opsi tidak valid. Silakan coba lagi.")

# Menanyakan apakah pengguna ingin memesan makanan lain
continue_order = input("Apakah Anda ingin memesan makanan lain? (ya/tidak): ").strip().lower()
if continue_order != 'ya':
    print("Terima kasih telah berbelanja!")

```

Untuk menjalankan blok kode tertentu berdasarkan kondisi yang diberikan, sehingga memberikan fleksibilitas dalam alur eksekusi program. Dengan if, kita bisa memeriksa kondisi awal; dengan elif, kita bisa memberikan beberapa kondisi alternatif; dan dengan else, kita menyediakan opsi terakhir jika semua kondisi sebelumnya tidak terpenuhi.

## 5. Def

```
# Opsi pencarian
search_option = input("Apakah Anda ingin mencari makanan? (ya/tidak): ").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_makanan = search_item(data_barang, keyword)

# Menampilkan hasil pencarian
if data_makanan:
    print("Hasil pencarian makanan:")
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama makanan", "Harga", "Stok"]
    for barang in data_barang:
        tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
    print(tabel)
else:
    print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
    continue # Kembali ke awal loop untuk mencoba lagi

# Memilih item untuk dipesan
while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)
    except ValueError:
        print("Input tidak valid. Silakan masukkan angka.")

    def data_user():
        users = []
        with open("user.csv", mode="r") as file:
            reader = csv.DictReader(file)
            for row in reader:
                row["saldo"] = int(float(row["saldo"])) # Convert saldo to integer
            users.append(row)
        return users

    # Fungsi untuk menyimpan data user ke file csv
    def simpan_data_user_ke_csv(data_users):
        with open("user.csv", mode="w", newline="") as file:
            writer = csv.DictWriter(file, fieldnames=["username", "password", "saldo"])
            writer.writeheader()
            for user in data_users:
                user["saldo"] = int(user["saldo"])
            writer.writerow(user)
```

```

def login():
    while True:
        print("== Selamat datang di BeliBeli ==")
        print("[1] Register")
        print("[2] Login")
        print("[3] Keluar")
        try:
            opsi = int(input("Masukkan pilihan anda : "))

            if opsi == 1:
                username1 = input("Masukkan username anda : ")
                if not username1:
                    print("Username tidak boleh kosong")
                    continue

                password1 = input("Masukkan Password anda : ")
                if not password1:
                    print("Password tidak boleh kosong")
                    continue

                user_sudah_ada = data_user()
                if any(user["username"] == username1 for user in user_sudah_ada):
                    print("Username sudah ada, masukkan username lain")
                    continue

# Menambahkan pengguna baru
            tambah_user_baru = {"username": username1, "password": password1, "saldo": "0"}
            user_sudah_ada.append(tambah_user_baru)
            simpan_data_user_ke_csv(user_sudah_ada)
            print("==== Akun anda telah dibuat ===")

            elif opsi == 2:
                username2 = input("Masukkan username anda : ")
                password2 = pwinput.pwinput("Masukkan Password anda : ")
                user_sudah_ada = data_user()
                login_success = False

                if username2 == "admin" and password2 == "admin123":
                    print("== Login berhasil, Selamat datang Admin ==")
                    menu_admin()
                    login_success = True

                else:
                    for user in user_sudah_ada:
                        if user["username"] == username2 and user["password"] == password2:
                            print(f"== Login berhasil, Selamat datang {username2}, selamat berbelanja")
                            login_success = True
                            menu_user(username2)
                            break

```

```

        else:
            for user in user_sudah_ada:
                if user["username"] == username2 and user["password"] == password2:
                    print(f"--- Login berhasil, Selamat datang {username2}, selamat berbelanja ---")
                    login_success = True
                    menu_user(username2)
                    break

            if not login_success:
                print("--- Akun tidak ada atau password salah ---")

        elif opsi == 3:
            print("--- Terima kasih telah menggunakan BeliBeli! ---")
            break # Keluar dari fungsi login

        else:
            print("Opsi tidak ada")
    except ValueError:
        print("Nomor harus berupa angka. Silakan coba lagi.")
        return login()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return login()
    except KeyboardInterrupt:
        print("Input tidak valid. Silakan coba lagi.")
        return login()

def menu_user(username):
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang di BELIBELI"]
        tabel.add_row(["[1] Order Makanan\n[2] Order Barang\n[3] Cek Voucher\n[4] Cek Keranjang\n[5]"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda : "))
            if opsi == 1:
                order_makanan(username)
            elif opsi == 2:
                order_barang(username)
            elif opsi == 3:
                lihat_voucher()
            elif opsi == 4:
                cek_keranjang(username)
            elif opsi == 5:
                cek_saldo(username)
            elif opsi == 6:
                top_up(username)
            elif opsi == 7:
                break
            else:
                print('input tidak valid')
                menu_user(username)
        except ValueError:
            print(" input harus berupa angka. Silakan coba lagi.")
            return menu_user(username)

```

```
except ValueError:
    print(" input harus berupa angka. Silakan coba lagi.")
    return menu_user(username)
except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    return menu_user(username)
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return menu_user(username)

def menu_admin():
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang admin"]
        tabel.add_row(["[1] Menu Makanan\n[2] Menu Barang\n[3] Menu Voucher\n[4] Keluar Program\n"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda: "))
            if opsi == 1:
                menu_makanan_admin()
            elif opsi == 2:
                menu_barang_admin()
            elif opsi == 3:
                menu_voucher()
            elif opsi == 4:
                print("Program berakhir")
                login()
                break
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            return menu_admin()
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            return menu_admin()
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return menu_admin()
```

```

def merge_sort(data, key, reverse=False):
    if len(data) <= 1:
        return data

    mid = len(data) // 2
    left_half = merge_sort(data[:mid], key, reverse)
    right_half = merge_sort(data[mid:], key, reverse)

    return merge(left_half, right_half, key, reverse)

def merge(left, right, key, reverse):
    sorted_list = []
    while left and right:
        if (left[0][key] <= right[0][key] and not reverse) or (left[0][key] > right[0][key] and reverse):
            sorted_list.append(left.pop(0))
        else:
            sorted_list.append(right.pop(0))

    sorted_list.extend(left)
    sorted_list.extend(right)

    return sorted_list

```

```

def order_makanan(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_makanan()
        saldo = user["saldo"]
        data_makanan = load_data_makanan() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan makanan berdasarkan harga? (ya/tidak): ").strip()
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah: ")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
        for makanan in data_makanan:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)

```

```

        elif action == '2':
            # Menambahkan makanan ke keranjang
            tambah_ke_keranjang(username, item_to_order["nama_barang"], item_to_order["harga"], quantity)
            print(f"{quantity} {item_to_order['nama_barang']} telah ditambahkan ke keranjang.")
        else:
            print("Opsi tidak valid. Silakan coba lagi.")

        # Menanyakan apakah pengguna ingin memesan makanan lain
        continue_order = input("Apakah Anda ingin memesan barang lain? (ya/tidak): ").strip().lower()
        if continue_order != 'ya':
            print("Terima kasih telah berbelanja!")
            return
    except ValueError:
        print("Input tidak valid. Harap masukkan angka yang benar.")
        continue # Kembali ke awal loop untuk mencoba lagi

if not item_to_order:
    print("Barang dengan nomor tersebut tidak ditemukan.")
    continue # Kembali ke awal loop untuk mencoba lagi

if item_to_order["stok"] <= 0:
    print("Maaf, stok tidak tersedia untuk item ini.")
    continue # Kembali ke awal loop untuk mencoba lagi

quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
if quantity > item_to_order["stok"]:
    print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
    continue # Kembali ke awal loop untuk mencoba lagi

# Pilihan untuk langsung beli atau masukkan ke keranjang
action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2):")
if action == '1':
    # Proses pembelian langsung
    total_harga = item_to_order["harga"] * quantity

    # Menggunakan voucher
    use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()
    if use_voucher == 'ya':
        lihat_voucher()
        kode_voucher = input("Masukkan kode voucher: ")
        diskon, voucher = gunakan_voucher(kode_voucher)
        if diskon > 0:

```

```

    a = total_harga
    b = diskon
    total_harga_setelah_diskon = a - (a * b / 100)
    total_harga = total_harga_setelah_diskon
    print(f"Voucher digunakan! Diskon: {diskon}%, Total harga setelah diskon: {total_harga:.2f}")
    voucher['used_by'] = username # Update voucher status
    update_voucher_status(voucher) # Update voucher status di CSV
else:
    print("Voucher tidak valid atau sudah digunakan.")

if saldo < total_harga:
    print("Saldo Anda tidak cukup untuk membeli jumlah yang diminta.")
    continue # Kembali ke awal loop untuk mencoba lagi

# Memperbarui saldo dan stok
user["saldo"] -= total_harga
item_to_order["stok"] -= quantity
simpan_data_user_ke_csv(pengguna)
simpan_makanan_ke_csv(data_makanan)

print(f"Pembelian berhasil! Saldo Anda sekarang: Rp {user['saldo']:.2f}")

invoice_makanan(username, item_to_order["nama_makanan"], quantity, total_harga)

elif action == '2':
    # Menambahkan makanan ke keranjang
    tambah_ke_keranjang(username, item_to_order["nama_makanan"], item_to_order["harga"], quantity)
    print(f"{quantity} {item_to_order['nama_makanan']} telah ditambahkan ke keranjang.")
else:
    print("Opsi tidak valid. Silakan coba lagi.")

# Menanyakan apakah pengguna ingin memesan makanan lain
continue_order = input("Apakah Anda ingin memesan makanan lain? (ya/tidak): ").strip().lower()
if continue_order != 'ya':
    print("Terima kasih telah berbelanja!")
    return

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi

```

```

def order_barang(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_barang()
        saldo = user["saldo"]
        data_barang = load_data_barang() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan barang berdasarkan harga? (ya/tidak): ")
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan barang:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]

        search_option = input("Apakah Anda ingin mencari barang? (ya/tidak): ").strip().lower()
        if search_option == 'ya':
            keyword = input("Masukkan kata kunci pencarian: ")
            data_barang = search_item(data_barang, keyword)

            # Menampilkan hasil pencarian
            if data_barang:
                print("Hasil pencarian barang:")
                tabel = PrettyTable()
                tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
                for barang in data_barang:
                    tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"], barang["stok"]])
                print(tabel)
            else:
                print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
                continue # Kembali ke awal loop untuk mencoba lagi

```

```

while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((barang for barang in data_barang if barang["no"] == item_no), None)

        if not item_to_order:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            continue # Kembali ke awal loop untuk mencoba lagi

        if item_to_order["stok"] <= 0:
            print("Maaf, stok tidak tersedia untuk item ini.")
            continue # Kembali ke awal loop untuk mencoba lagi

        quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
        if quantity > item_to_order["stok"]:
            print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
            continue # Kembali ke awal loop untuk mencoba lagi

        # Pilihan untuk langsung beli atau masukkan ke keranjang
        action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")
        if action == '1':
            # Proses pembelian langsung
            total_harga = item_to_order["harga"] * quantity

            # Menggunakan voucher
            use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()
            if use_voucher == 'ya':
                keyword = input("Masukkan kata kunci pencarian: ")
                data_barang = search_item(data_barang, keyword)

            # Menampilkan hasil pencarian
            if data_barang:
                print("Hasil pencarian barang:")
                tabel = PrettyTable()
                tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
                for barang in data_barang:
                    tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"], barang["stok"]])
                print(tabel)
            else:
                print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
                continue # Kembali ke awal loop untuk mencoba lagi

        # Memilih item untuk dipesan
        while True:
            try:
                item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
                item_to_order = next((barang for barang in data_barang if barang["no"] == item_no), None)

                if not item_to_order:
                    print("Barang dengan nomor tersebut tidak ditemukan.")
                    continue # Kembali ke awal loop untuk mencoba lagi

```

```

if item_to_order["stok"] <= 0:
    print("Maaf, stok tidak tersedia untuk item ini.")
    continue # Kembali ke awal loop untuk mencoba lagi

quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
if quantity > item_to_order["stok"]:
    print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
    continue # Kembali ke awal loop untuk mencoba lagi

# Pilihan untuk langsung beli atau masukkan ke keranjang
action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")
if action == '1':
    # Proses pembelian langsung
    total_harga = item_to_order["harga"] * quantity

    # Menggunakan voucher
    use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()
    if use_voucher == 'ya':
        lihat_voucher()
        kode_voucher = input("Masukkan kode voucher: ")
        diskon, voucher = gunakan_voucher(kode_voucher)
        if diskon > 0:
            a = total_harga
            b = diskon
            total_harga_setelah_diskon = a - (a * b / 100)

            total_harga_setelah_diskon = a - (a * b / 100)
            total_harga = total_harga_setelah_diskon
            print(f"Voucher digunakan! Diskon: {diskon}%, Total harga setelah diskon: {total_harga:.2f}")
            voucher['used_by'] = username # Update voucher status
            update_voucher_status(voucher) # Update voucher status di CSV
        else:
            print("Voucher tidak valid atau sudah digunakan.")

    if saldo < total_harga:
        print("Saldo Anda tidak cukup untuk membeli jumlah yang diminta.")
        continue # Kembali ke awal loop untuk mencoba lagi

    # Memperbarui saldo dan stok
    user["saldo"] -= total_harga
    item_to_order["stok"] -= quantity
    simpan_data_user_ke_csv(pengguna)
    simpan_barang_ke_csv(data_barang)

print(f"Pembelian berhasil! Saldo Anda sekarang: Rp {user['saldo']:.2f}")

invoice_barang(username, item_to_order["nama_barang"], quantity, total_harga)

```

```

def checkout_selected_items(username, keranjang):
    pengguna = data_user()
    user = next((user for user in pengguna if user["username"] == username), None)

    if not user:
        print("Pengguna tidak ditemukan.")
        return

    selected_items = []
    while True:
        try:
            item_no = int(input("Masukkan nomor item yang ingin Anda checkout (0 untuk selesai): "))
            if item_no == 0:
                break
            if 1 <= item_no <= len(keranjang):
                selected_items.append(keranjang[item_no - 1])
            else:
                print("Nomor item tidak valid. Silakan coba lagi.")
        except ValueError:
            print("Input tidak valid. Silakan coba lagi.")

```

```

# Fungsi untuk menampilkan Invoice makanan
def invoice_makanan(username, nama_makanan, jumlah_makanan, total_harga):
    struk = PrettyTable()
    struk.field_names = ["Invoice"]
    struk.add_row([f"Nama Customer : {username}"])
    struk.add_row([f"Tanggal Transaksi : {datetime.now().strftime('%d-%m-%y %H:%M:%S')}"])
    struk.add_row([f"Nama Produk : {nama_makanan}"])
    struk.add_row([f"Jumlah Item : {jumlah_makanan}"])
    struk.add_row([f"Total Bayar : {total_harga}"])
    struk.add_row(["Terima kasih sudah bertransaksi"])
    print(struk)

# Fungsi untuk menampilkan Invoice barang
def invoice_barang(username, nama_barang, jumlah_barang, total_harga):
    struk = PrettyTable()
    struk.field_names = ["Invoice"]
    struk.add_row([f"Nama Customer : {username}"])
    struk.add_row([f"Tanggal Transaksi : {datetime.now().strftime('%d-%m-%y %H:%M:%S')}"])
    struk.add_row([f"Nama Produk : {nama_barang}"])
    struk.add_row([f"Jumlah Item : {jumlah_barang}"])
    struk.add_row([f"Total Bayar : {total_harga}"])
    struk.add_row(["Terima kasih sudah bertransaksi"])
    print(struk)

```

```

# Fungsi untuk menampilkan data voucher
def lihat_voucher():
    data_vouchers = load_data_voucher()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Kode Voucher", "Diskon (%)"]

    for voucher in data_vouchers:
        tabel.add_row([voucher["no"], voucher["kode"], voucher["diskon"]])
    print(tabel)

def cek_keranjang(username):
    keranjang = load_keranjang(username)

    if not keranjang:
        print("Keranjang Anda kosong.")
        return

    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Item", "Total Harga", "Status"]

    for index, item in enumerate(keranjang, start=1):
        tabel.add_row([index, item["nama_item"], item["total_harga"], item["status"]])

    print(tabel)

    checkout_option = input("Apakah Anda ingin checkout semua item di keranjang (ketik 'semua')")

    if checkout_option == 'semua':
        checkout_keranjang(username, keranjang)
    elif checkout_option == 'pilih':
        checkout_selected_items(username, keranjang)
    else:
        print("Opsi tidak valid. Silakan coba lagi.")

```

```

# Fungsi untuk menampilkan saldo
def cek_saldo(username):
    cek_saldo = data_user()
    for user in cek_saldo:
        if user["username"] == username:
            print(f"Saldo anda adalah = Rp {user['saldo']}")

# Fungsi untuk top-up saldo
def top_up(username):
    cek_saldo = data_user()
    for user in cek_saldo:
        if user["username"] == username:
            print(f"Saldo anda saat ini adalah = Rp {user['saldo']}")
            dana = input("Berapakah yang ingin anda topup? : Rp")
            if dana.isdigit():
                dana = int(dana)
                if dana > 2000000:
                    print("Top up melampaui batas maksimal, batas maksimal : Rp 2.000.000")
                else:
                    user["saldo"] = int(user["saldo"]) + dana # Pastikan saldo tetap sebagai integer
                    simpan_data_user_ke_csv(cek_saldo)
                    print(f"Saldo anda telah ditambahkan. Saldo sekarang = Rp {user['saldo']}") 
            else:
                print("Masukkan nominal yang valid.")
            break # Keluar dari loop setelah memproses pengguna

```

```

# Fungsi untuk memuat data makanan
def load_data_makanan():
    makanan = []
    with open("stok_makanan.csv", mode="r") as file:
        reader = csv.DictReader(file)
        for row in reader:
            row["no"] = int(row["no"])
            row["harga"] = int(row["harga"])
            row["stok"] = int(row["stok"])
            makanan.append(row)
    return makanan

# Fungsi untuk menyimpan data makanan ke CSV
def simpan_makanan_ke_csv(data_makanan):
    with open("stok_makanan.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "nama_makanan", "harga", "stok"])
        writer.writeheader()
        for makanan in data_makanan:
            writer.writerow(makanan)

```

```
# Fungsi untuk memuat data barang
def load_data_barang():
    barang = []
    with open("stok_barang.csv", mode="r") as file:
        reader = csv.DictReader(file)
        for row in reader:
            row["no"] = int(row["no"])
            row["harga"] = int(row["harga"])
            row["stok"] = int(row["stok"])
            barang.append(row)
    return barang

# Fungsi untuk menyimpan data barang ke CSV
def simpan_barang_ke_csv(data_barang):
    with open("stok_barang.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "nama_barang", "harga", "stok"])
        writer.writeheader()
        for barang in data_barang:
            writer.writerow(barang)
```

```
# Fungsi untuk menyimpan data keranjang ke CSV
def simpan_keranjang_ke_csv(keranjang):
    with open("keranjang.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["username", "nama_item", "total_harga", "status"])
        writer.writeheader()
        for item in keranjang:
            writer.writerow(item)

def save_keranjang(username, keranjang):
    existing_data = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            existing_data = list(reader)
    except FileNotFoundError:
        pass

    for item in keranjang:
        existing_data.append({
            "username": username,
            "nama_item": item["nama_item"],
            "total_harga": item["total_harga"],
            "status": item["status"]
        })
```

```
def checkout_keranjang(username, keranjang):
    pengguna = data_user()
    user = next((user for user in pengguna if user["username"] == username), None)

    if not user:
        print("Pengguna tidak ditemukan.")
        return

    saldo = user["saldo"]
    total_belanja = sum(int(item["total_harga"]) for item in keranjang)

    if saldo < total_belanja:
        print("Saldo Anda tidak cukup untuk checkout.")
        return

    user["saldo"] -= total_belanja
    for item in keranjang:
        item["status"] = "sudah dibayar"

    simpan_data_user_ke_csv(pengguna)
    simpan_keranjang_ke_csv(keranjang)
    print(f"Checkout berhasil! Saldo Anda sekarang: Rp {user['saldo']}")
```

```
def tambah_ke_keranjang(username, nama_item, harga, quantity):
    keranjang = load_keranjang(username)
    total_harga = harga * quantity
    keranjang.append({
        "nama_item": nama_item,
        "total_harga": total_harga,
        "status": "belum dibayar"
    })
    save_keranjang(username, keranjang)

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    menu_makanan_admin()
    return
```

```

if not selected_items:
    print("Tidak ada item yang dipilih untuk checkout.")
    return
saldo = user["saldo"]
total_belanja = sum(int(item["total_harga"]) for item in selected_items)
if saldo < total_belanja:
    print("Saldo Anda tidak cukup untuk checkout.")
    return
user["saldo"] -= total_belanja
for item in selected_items:
    item["status"] = "sudah dibayar"

simpan_data_user_ke_csv(pengguna)
simpan_keranjang_ke_csv(keranjang)
print(f"Checkout berhasil! Saldo Anda sekarang: Rp {user['saldo']}")

def load_keranjang(username):
    keranjang = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            for row in reader:
                if row['username'] == username:
                    keranjang.append({
                        "nama_item": row['nama_item'],
                        "total_harga": row['total_harga'],
                        "status": row['status']
                    })
    except FileNotFoundError:
        print("File keranjang tidak ditemukan.")
    return keranjang

def simpan_keranjang_ke_csv(existing_data):
    with open("keranjang.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["username", "nama_item", "total_harga", "status"])
        writer.writeheader()
        for item in existing_data:
            writer.writerow(item)

```

```

def save_keranjang(username, keranjang):
    existing_data = load_all_keranjang()

    existing_data = [item for item in existing_data if item['username'] != username]

    for item in keranjang:
        existing_data.append({
            "username": username,
            "nama_item": item["nama_item"],
            "total_harga": item["total_harga"],
            "status": item["status"]
        })

    simpan_keranjang_ke_csv(existing_data)

def load_all_keranjang():
    keranjang_data = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            for row in reader:
                keranjang_data.append(row)
    except FileNotFoundError:
        print("File keranjang tidak ditemukan.")
    return keranjang_data

def lihat_barang():
    data_barang = load_data_barang()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]

    for barang in data_barang:
        tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"], barang["stok"]])
    print(tabel)

# Fungsi untuk mengupdate data barang
def update_barang():
    lihat_barang()
    data_barang = load_data_barang()

    try:
        item_no = int(input("Masukkan nomor barang yang ingin diupdate: "))
        item_to_update = next((barang for barang in data_barang if barang["no"] == item_no), None)
        if not item_to_update:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            return
        nama_baru_barang = input(f"Masukkan nama baru untuk {item_to_update['nama_barang']} : ")
        harga_baru_barang = input(f"Masukkan harga baru untuk {item_to_update['nama_barang']} : ")
        stok_baru_barang = input(f"Masukkan stok baru untuk {item_to_update['nama_barang']} : ")
    
```

```

if nama_baru_barang:
    item_to_update["nama_barang"] = nama_baru_barang
if harga_baru_barang.isdigit():
    item_to_update["harga"] = int(harga_baru_barang)
if stok_baru_barang.isdigit():
    item_to_update["stok"] = int(stok_baru_barang)

simpan_barang_ke_csv(data_barang)
print("Barang berhasil diperbarui.")
except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")
    return update_barang()
except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    return update_barang()
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return update_barang()

```

```

def hapus_barang():
    data_barang = load_data_barang()
    lihat_barang()
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dihapus: "))

        item_to_delete = next((barang for barang in data_barang if barang["no"] == item_no), None)
        if not item_to_delete:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            return
        data_barang = [barang for barang in data_barang if barang["no"] != item_no]
        for index, barang in enumerate(data_barang, start=1):
            barang["no"] = index
        simpan_barang_ke_csv(data_barang)
        print("Barang berhasil dihapus.")
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
        return hapus_barang()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return hapus_barang()
    except KeyboardInterrupt:
        print("Input tidak valid. Silakan coba lagi.")
        return hapus_barang()

```

```

def menu_barang_admin():
    while True:
        print("==== Menu Barang ===")
        print("[1] Tambah Barang")
        print("[2] Lihat Barang")
        print("[3] Update Barang")
        print("[4] Hapus Barang")
        print("[5] Kembali")
        try:
            opsi = int(input("Masukkan pilihan anda : "))

            if opsi == 1:
                tambah_barang()
            elif opsi == 2:
                lihat_barang()
            elif opsi == 3:
                update_barang()
            elif opsi == 4:
                hapus_barang()
            elif opsi == 5:
                print("keluar dari menu barang")
                break
            else:
                print("Opsi tidak ada.")

        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            menu_barang_admin()
            return
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            menu_barang_admin()
            return
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            menu_barang_admin()
            return

```

```

def menu_makanan_admin():
    while True:
        print("== Menu Makanan ==")
        print("[1] Tambah Makanan")
        print("[2] Lihat Makanan")
        print("[3] Update Makanan")
        print("[4] Hapus Makanan")
        print("[5] Kembali")
        try:
            opsi = int(input("Masukkan pilihan anda : "))
            if opsi == 1:
                tambah_makanan()
            elif opsi == 2:
                lihat_makanan()
            elif opsi == 3:
                update_makanan()
            elif opsi == 4:
                hapus_makanan()
            elif opsi == 5:
                break
            else:
                print("Opsi tidak ada.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            menu_makanan_admin()
            return
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")

def tambah_voucher():
    while True:
        data_vouchers = load_data_voucher()
        try:
            kode = input("Masukkan kode voucher: ")
            diskon = input("Masukkan besaran diskon (dalam persen): ")

            if not diskon.isdigit():
                print("Diskon harus berupa angka.")
                return

            diskon = int(diskon)
            no = max([voucher["no"] for voucher in data_vouchers], default=0) + 1
            voucher_baru = {"no": no, "kode": kode, "diskon": diskon}
            data_vouchers.append(voucher_baru)

            simpan_voucher_ke_csv(data_vouchers)
            print("Voucher baru berhasil ditambahkan.")
            break # Keluar dari loop setelah berhasil menambah voucher
        except ValueError:
            print("Input tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except EOFError:
            print("\nInput tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return # Keluar dari fungsi jika ada interupsi

```

```

# Fungsi untuk menampilkan menu makanan
def load_data_makanan():
    makanan = []
    with open("stok_makanan.csv", mode="r") as file:
        reader = csv.DictReader(file)
        for row in reader:
            row["no"] = int(row["no"])
            row["harga"] = int(row["harga"])
            row["stok"] = int(row["stok"])
            makanan.append(row)
    return makanan

# Fungsi untuk menyimpan menu makanan ke dalam CSV
def simpan_makanan_ke_csv(data_makanan):
    with open("stok_makanan.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "nama_makanan", "harga", "stok"])
        writer.writeheader()
        for makanan in data_makanan:
            writer.writerow(makanan)

def tambah_makanan():
    data_makanan = load_data_makanan()
    try:
        no = len(data_makanan) + 1

        nama_makanan = input("Masukkan nama makanan: ")
        harga = int(input("Masukkan harga makanan: "))
        stok = int(input("Masukkan stok makanan: "))

        makanan_baru = {"no": no, "nama_makanan": nama_makanan, "harga": harga, "stok": stok}
        data_makanan.append(makanan_baru)

        simpan_makanan_ke_csv(data_makanan)
        print("Makanan baru berhasil ditambahkan.")
    except:
        print("\nPerhatikan input")
    return tambah_makanan()

# Fungsi untuk melihat menu makanan
def lihat_makanan():
    data_makanan = load_data_makanan()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Makanan", "Harga", "Stok"]

    for makanan in data_makanan:
        tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
    print(tabel)

def lihat_makanan():
    data_makanan = load_data_makanan()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Makanan", "Harga", "Stok"]

    for makanan in data_makanan:
        tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
    print(tabel)

```

```

def update_makanan():
    while True:
        lihat_makanan()
        data_makanan = load_data_makanan()
        try:
            item_no = int(input("Masukkan nomor makanan yang ingin diupdate: "))
            item_to_update = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)
            if not item_to_update:
                print("Barang dengan nomor tersebut tidak ditemukan.")
                return
            nama_baru_makanan = input(f"Masukkan nama baru untuk {item_to_update['nama_makanan']} : ")
            harga_baru_makanan= input(f"Masukkan harga baru untuk {item_to_update['nama_makanan']} : ")
            stok_baru_makanan = input(f"Masukkan stok baru untuk {item_to_update['nama_makanan']} : ")

            if nama_baru_makanan:
                item_to_update["nama_makanan"] = nama_baru_makanan
            if harga_baru_makanan.isdigit():
                item_to_update["harga"] = int(harga_baru_makanan)
            if harga_baru_makanan.isdigit():
                item_to_update["stok"] = int(stok_baru_makanan)

            simpan_makanan_ke_csv(data_makanan)
            print("Barang berhasil diperbarui.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            return update_makanan()
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            return update_makanan()
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return update_makanan()
        break

# Fungsi untuk mengupdate stok makanan
def update_stok(menu_makanan):
    with open("stok_makanan.csv", "w", newline="") as file:
        fieldnames = ["no", "nama_makanan", "harga", "stok"]
        writer = csv.DictWriter(file, fieldnames=fieldnames)
        writer.writeheader()
        for menu in menu_makanan:
            writer.writerow({"no":menu["no"], "nama_makanan": menu["nama_makanan"], "harga": menu["harga"], "stok": menu["stok"]})

```

```

def hapus_makanan():
    data_makanan = load_data_makanan()
    lihat_makanan()
    try:
        item_no = int(input("Masukkan nomor makanan yang ingin dihapus: "))

        item_to_delete = next(makanan for makanan in data_makanan if makanan["no"] == item_no, None)
        if not item_to_delete:
            print("makanan dengan nomor tersebut tidak ditemukan.")
            return
        data_makanan = [makanan for makanan in data_makanan if makanan["no"] != item_no]
        for index, makanan in enumerate(data_makanan, start=1):
            makanan["no"] = index
        simpan_makanan_ke_csv(data_makanan)
        print("makanan berhasil dihapus.")
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
        return hapus_makanan()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return hapus_makanan()
    except KeyboardInterrupt:
        print("Input tidak valid. Silakan coba lagi.")
        return hapus_makanan()

def load_data_voucher():
    data_vouchers = []
    with open('voucher.csv', mode='r') as file:
        reader = csv.DictReader(file)
        for row in reader:
            # Pastikan 'no' adalah integer
            row['no'] = int(row['no']) # Konversi ke integer
            row['diskon'] = int(row['diskon']) # Pastikan diskon juga integer
            data_vouchers.append(row)
    return data_vouchers

# Fungsi untuk menggunakan voucher
def gunakan_voucher(kode_voucher):
    vouchers = load_data_voucher()
    for voucher in vouchers:
        if voucher['kode'] == kode_voucher and not voucher['used_by']:
            return int(voucher['diskon']), voucher
    return 0, None

```

```

def update_voucher_status(voucher_used):
    vouchers = load_data_voucher()
    for voucher in vouchers:
        if voucher['kode'] == voucher_used['kode']:
            voucher['used_by'] = voucher_used['used_by']

    # Menyimpan kembali ke file CSV
    with open("voucher.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "kode", "diskon", "used_by"])
        writer.writeheader()
        writer.writerows(vouchers)

def simpan_voucher_ke_csv(data_vouchers):
    with open("voucher.csv", mode="w", newline="") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "kode", "diskon", "used_by"])
        writer.writeheader()
        for voucher in data_vouchers:
            writer.writerow(voucher)

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    menu_voucher()
    return
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    menu_voucher()
    return

def lihat_voucher():
    data_vouchers = load_data_voucher()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Kode Voucher", "Diskon (%)"]

    for voucher in data_vouchers:
        tabel.add_row([voucher["no"], voucher["kode"], voucher["diskon"]])
    print(tabel)

```

```

def update_voucher():
    while True:
        lihat_voucher() # Menampilkan daftar voucher
        data_vouchers = load_data_voucher() # Memuat data voucher
        try:
            item_no = input("Masukkan nomor voucher yang ingin diupdate: ").strip() # Menghapus spasi di awal dan akhir
            print(f"Input yang diterima: '{item_no}'") # Debugging: Menampilkan input yang diterima

            if not item_no.isdigit():
                print("Input harus berupa angka. Silakan coba lagi.")
                continue # Kembali ke awal loop untuk mencoba lagi

            item_no = int(item_no) # Konversi ke integer
            print(f"Nomor voucher yang dicari: {item_no}") # Debugging: Menampilkan nomor yang dicari

            item_to_update = next((voucher for voucher in data_vouchers if int(voucher["no"]) == item_no),
            if not item_to_update:
                print("Voucher dengan nomor tersebut tidak ditemukan.")
                continue # Kembali ke awal loop untuk mencoba lagi

            # Meminta input baru untuk kode dan diskon
            kode_baru_voucher = input(f"Masukkan kode baru untuk {item_to_update['kode']} : ")
            diskon_baru_voucher = input(f"Masukkan diskon baru untuk {item_to_update['diskon']} : ")

            # Memperbarui kode jika ada input baru
            if kode_baru_voucher:
                item_to_update["kode"] = kode_baru_voucher

            # Memperbarui diskon jika ada input baru
            if diskon_baru_voucher.isdigit():
                item_to_update["diskon"] = int(diskon_baru_voucher)

            # Menyimpan kembali data voucher ke CSV
            simpan_voucher_ke_csv(data_vouchers)
            print(f"Voucher '{item_to_update['kode']}' berhasil diperbarui.")
            break # Keluar dari loop setelah berhasil memperbarui

        except ValueError:
            print("Input tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except EOFError:
            print("\nInput tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return # Keluar dari fungsi jika ada interupsi

```

```

def hapus_voucher():
    while True:
        try:
            lihat_voucher()
            data_vouchers = load_data_voucher()
            try:
                no = int(input("Masukkan nomor voucher yang ingin dihapus: "))
            except ValueError:
                print("Nomor harus berupa angka. Silakan coba lagi.")
                return
            voucher_to_delete = next((voucher for voucher in data_vouchers if voucher["no"] == no), None)

            if not voucher_to_delete:
                print("Voucher dengan nomor tersebut tidak ditemukan.")
                return
            data_vouchers = [voucher for voucher in data_vouchers if voucher["no"] != no]
            for index, voucher in enumerate(data_vouchers, start=1):
                voucher["no"] = index
            simpan_voucher_ke_csv(data_vouchers)
            print(f"Voucher dengan nomor {no} berhasil dihapus.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")

        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            hapus_voucher()
            return
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            hapus_voucher()
            return
        break

```

```
def menu_voucher():
    while True:
        print("== Menu Voucher ==")
        print("[1] Tambah Voucher")
        print("[2] Lihat Voucher")
        print("[3] Update Voucher")
        print("[4] Hapus Voucher")
        print("[5] Kembali")
        try:
            opsi = int(input("Masukkan pilihan anda : "))

            if opsi == 1:
                tambah_voucher()
            elif opsi == 2:
                lihat_voucher()
            elif opsi == 3:
                update_voucher()
            elif opsi == 4:
                hapus_voucher()
            elif opsi == 5:
                break
            else:
                print("Opsi tidak ada.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
    menu_voucher()
    return
```

Fungsi-fungsi dalam sistem manajemen toko online berperan penting dalam mengelola berbagai aspek operasional, mulai dari data pengguna hingga transaksi penjualan. Mereka memungkinkan pengguna untuk melakukan pemesanan, mengakses informasi produk, dan mengelola akun dengan mudah. Secara keseluruhan, fungsi-fungsi ini berkontribusi pada pengalaman berbelanja yang efisien dan terorganisir.

## 6. Loop while True

```
def login():
    while True:
        print("== Selamat datang di BeliBeli ==")
        print("[1] Register")
        print("[2] Login")
        print("[3] Keluar")
        try:
            opsi = int(input("Masukkan pilihan anda : "))

            if opsi == 1:
                username1 = input("Masukkan username anda : ")
                if not username1:
                    print("Username tidak boleh kosong")
                    continue

                password1 = input("Masukkan Password anda : ")
                if not password1:
                    print("Password tidak boleh kosong")
                    continue

                user_sudah_ada = data_user()
                if any(user["username"] == username1 for user in user_sudah_ada):
                    print("Username sudah ada, masukkan username lain")
                    continue

            # Menambahkan pengguna baru
            tambah_user_baru = {"username": username1, "password": password1, "saldo": "0"}
            user_sudah_ada.append(tambah_user_baru)
            simpan_data_user_ke_csv(user_sudah_ada)
            print("== Akun anda telah dibuat ==")

            elif opsi == 2:
                username2 = input("Masukkan username anda : ")
                password2 = pwinput.pwinput("Masukkan Password anda : ")
                user_sudah_ada = data_user()
                login_success = False

                if username2 == "admin" and password2 == "admin123":
                    print("== Login berhasil, Selamat datang Admin ==")
                    menu_admin()
                    login_success = True

                else:
                    for user in user_sudah_ada:
                        if user["username"] == username2 and user["password"] == password2:
                            print(f"== Login berhasil, Selamat datang {username2}, selamat berbelanja")
                            login_success = True
                            menu_user(username2)
                            break
```

```

        else:
            for user in user_sudah_ada:
                if user["username"] == username2 and user["password"] == password2:
                    print(f"--- Login berhasil, Selamat datang {username2}, selamat berbelanja ---")
                    login_success = True
                    menu_user(username2)
                    break

            if not login_success:
                print("== Akun tidak ada atau password salah ==")

        elif opsi == 3:
            print("== Terima kasih telah menggunakan BeliBeli! ==")
            break # Keluar dari fungsi login

        else:
            print("Opsi tidak ada")
    except ValueError:
        print("Nomor harus berupa angka. Silakan coba lagi.")
        return login()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return login()
    except KeyboardInterrupt:
        print("Input tidak valid. Silakan coba lagi.")
        return login()

def menu_user(username):
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang di BELIBELI"]
        tabel.add_row(["[1] Order Makanan\n[2] Order Barang\n[3] Cek Voucher\n[4] Cek Keranjang\n[5]"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda : "))
            if opsi == 1:
                order_makanan(username)
            elif opsi == 2:
                order_barang(username)
            elif opsi == 3:
                lihat_voucher()
            elif opsi == 4:
                cek_keranjang(username)
            elif opsi == 5:
                cek_saldo(username)
            elif opsi == 6:
                top_up(username)
            elif opsi == 7:
                break
            else:
                print('input tidak valid')
                menu_user(username)
        except ValueError:
            print(" input harus berupa angka. Silakan coba lagi.")
            return menu_user(username)

```

```

except ValueError:
    print(" input harus berupa angka. Silakan coba lagi.")
    return menu_user(username)
except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    return menu_user(username)
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return menu_user(username)

def menu_admin():
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang admin"]
        tabel.add_row(["[1] Menu Makanan\n[2] Menu Barang\n[3] Menu Voucher\n[4] Keluar Program\n"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda: "))
            if opsi == 1:
                menu_makanan_admin()
            elif opsi == 2:
                menu_barang_admin()
            elif opsi == 3:
                menu_voucher()
            elif opsi == 4:
                print("Program berakhir")
                login()
                break
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            return menu_admin()
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            return menu_admin()
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return menu_admin()

```

```
def menu_barang_admin():
    while True:
        print("==== Menu Barang ===")
        print("[1] Tambah Barang")
        print("[2] Lihat Barang")
        print("[3] Update Barang")
        print("[4] Hapus Barang")
        print("[5] Kembali")
        try:
            opsi = int(input("Masukkan pilihan anda : "))

            if opsi == 1:
                tambah_barang()
            elif opsi == 2:
                lihat_barang()
            elif opsi == 3:
                update_barang()
            elif opsi == 4:
                hapus_barang()
            elif opsi == 5:
                print("keluar dari menu barang")
                break
            else:
                print("Opsi tidak ada.")

        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            menu_barang_admin()
            return
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            menu_barang_admin()
            return
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            menu_barang_admin()
            return
```

```

def order_barang(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_barang()
        saldo = user["saldo"]
        data_barang = load_data_barang() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan barang berdasarkan harga? (ya/tidak): ")
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan barang:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]

def checkout_selected_items(username, keranjang):
    pengguna = data_user()
    user = next((user for user in pengguna if user["username"] == username), None)

    if not user:
        print("Pengguna tidak ditemukan.")
        return

    selected_items = []
    while True:
        try:
            item_no = int(input("Masukkan nomor item yang ingin Anda checkout (0 untuk selesai): "))
            if item_no == 0:
                break
            if 1 <= item_no <= len(keranjang):
                selected_items.append(keranjang[item_no - 1])
            else:
                print("Nomor item tidak valid. Silakan coba lagi.")
        except ValueError:
            print("Input tidak valid. Silakan coba lagi.")

```

```

        elif action == '2':
            # Menambahkan makanan ke keranjang
            tambah_ke_keranjang(username, item_to_order["nama_makanan"], item_to_order["harga"], quantity)
            print(f"{quantity} {item_to_order['nama_makanan']} telah ditambahkan ke keranjang.")
        else:
            print("Opsi tidak valid. Silakan coba lagi.")

        # Menanyakan apakah pengguna ingin memesan makanan lain
        continue_order = input("Apakah Anda ingin memesan makanan lain? (ya/tidak): ").strip().lower()
        if continue_order != 'ya':
            print("Terima kasih telah berbelanja!")
            return
    except ValueError:
        print("Input tidak valid. Harap masukkan angka yang benar.")
        continue # Kembali ke awal loop untuk mencoba lagi

while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((barang for barang in data_barang if barang["no"] == item_no), None)

        if not item_to_order:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            continue # Kembali ke awal loop untuk mencoba lagi

        if item_to_order["stok"] <= 0:
            print("Maaf, stok tidak tersedia untuk item ini.")
            continue # Kembali ke awal loop untuk mencoba lagi

        quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
        if quantity > item_to_order["stok"]:
            print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
            continue # Kembali ke awal loop untuk mencoba lagi

        # Pilihan untuk langsung beli atau masukkan ke keranjang
        action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")
        if action == '1':
            # Proses pembelian langsung
            total_harga = item_to_order["harga"] * quantity

            # Menggunakan voucher
            use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()

```

```

search_option = input("Apakah Anda ingin mencari barang? (ya/tidak): ").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_barang = search_item(data_barang, keyword)

    # Menampilkan hasil pencarian
    if data_barang:
        print("Hasil pencarian barang:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
        for barang in data_barang:
            tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"], barang["stok"]])
        print(tabel)
    else:
        print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
        continue # Kembali ke awal loop untuk mencoba lagi

except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")
    return menu_user(username)
except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    return menu_user(username)
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return menu_user(username)

def order_makanan(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_makanan()
        saldo = user["saldo"]
        data_makanan = load_data_makanan() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan makanan berdasarkan harga? (ya/tidak): ").strip()
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
        for makanan in data_makanan:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)

```

```

# Opsi pencarian
search_option = input("Apakah Anda ingin mencari makanan? (ya/tidak): ").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_makanan = search_item(data_barang, keyword)

    # Menampilkan hasil pencarian
    if data_makanan:
        print("Hasil pencarian makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama makanan", "Harga", "Stok"]
        for barang in data_barang:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)
    else:
        print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
        continue # Kembali ke awal loop untuk mencoba lagi

# Memilih item untuk dipesan
while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)
        if not item_to_order:
            print("Item tidak ditemukan. Silakan coba lagi")
            continue
    except ValueError:
        print("Input tidak valid. Silakan masukkan angka")
        continue

# Opsi pencarian
search_option = input("Apakah Anda ingin mencari makanan? (ya/tidak): ").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_makanan = search_item(data_barang, keyword)

    # Menampilkan hasil pencarian
    if data_makanan:
        print("Hasil pencarian makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama makanan", "Harga", "Stok"]
        for barang in data_barang:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)
    else:
        print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
        continue # Kembali ke awal loop untuk mencoba lagi

# Memilih item untuk dipesan
while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)
        if not item_to_order:
            print("Item tidak ditemukan. Silakan coba lagi")
            continue
    except ValueError:
        print("Input tidak valid. Silakan masukkan angka")
        continue

```

```

        a = total_harga
        b = diskon
        total_harga_setelah_diskon = a - (a * b / 100)
        total_harga = total_harga_setelah_diskon
        print(f"Voucher digunakan! Diskon: {diskon}%, Total harga setelah diskon: {total_harga:.2f}")
        voucher['used_by'] = username # Update voucher status
        update_voucher_status(voucher) # Update voucher status di CSV
    else:
        print("Voucher tidak valid atau sudah digunakan.")

if saldo < total_harga:
    print("Saldo Anda tidak cukup untuk membeli jumlah yang diminta.")
    continue # Kembali ke awal loop untuk mencoba lagi

# Memperbarui saldo dan stok
user["saldo"] -= total_harga
item_to_order["stok"] -= quantity
simpan_data_user_ke_csv(pengguna)
simpan_makanan_ke_csv(data_makanan)

print(f"Pembelian berhasil! Saldo Anda sekarang: Rp {user['saldo']:.2f}")

invoice_makanan(username, item_to_order["nama_makanan"], quantity, total_harga)

elif action == '2':
    # Menambahkan makanan ke keranjang
    tambah_ke_keranjang(username, item_to_order["nama_makanan"], item_to_order["harga"], quantity)
    print(f"{quantity} {item_to_order['nama_makanan']} telah ditambahkan ke keranjang.")
else:
    print("Opsi tidak valid. Silakan coba lagi.")

# Menanyakan apakah pengguna ingin memesan makanan lain
continue_order = input("Apakah Anda ingin memesan makanan lain? (ya/tidak): ").strip().lower()
if continue_order != 'ya':
    print("Terima kasih telah berbelanja!")
    return
except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi

```

Pernyataan if dan elif dalam pemrograman digunakan untuk mengontrol alur eksekusi berdasarkan kondisi tertentu. Dengan menggunakan if, program dapat mengevaluasi suatu kondisi dan menjalankan blok kode yang sesuai jika kondisi tersebut benar, sementara elif memungkinkan penambahan kondisi tambahan yang akan dievaluasi jika kondisi sebelumnya salah

## 7. Try except

```
def login():
    while True:
        print("== Selamat datang di BeliBeli ==")
        print("[1] Register")
        print("[2] Login")
        print("[3] Keluar")
        try:
            opsi = int(input("Masukkan pilihan anda : "))

            if opsi == 1:
                username1 = input("Masukkan username anda : ")
                if not username1:
                    print("Username tidak boleh kosong")
                    continue

                password1 = input("Masukkan Password anda : ")
                if not password1:
                    print("Password tidak boleh kosong")
                    continue

                user_sudah_ada = data_user()
                if any(user["username"] == username1 for user in user_sudah_ada):
                    print("Username sudah ada, masukkan username lain")
                    continue

                tambah_user_baru = {"username": username1, "password": password1, "saldo": "0"}
                user_sudah_ada.append(tambah_user_baru)
                simpan_data_user_ke_csv(user_sudah_ada)
                print("== Akun anda telah dibuat ==")

            elif opsi == 2:
                username2 = input("Masukkan username anda : ")
                password2 = pwininput.pwininput("Masukkan Password anda : ")
                user_sudah_ada = data_user()
                login_success = False

                if username2 == "admin" and password2 == "admin123":
                    print("== Login berhasil, Selamat datang Admin ==")
                    menu_admin()
                    login_success = True

                else:
                    for user in user_sudah_ada:
                        if user["username"] == username2 and user["password"] == password2:
                            print(f"== Login berhasil, Selamat datang {username2}, selamat berbelanja ==")
                            login_success = True
                            menu_user(username2)
                            break

                if not login_success:
                    print("== Akun tidak ada atau password salah ==")
```

```

        else:
            print("Opsi tidak ada")
    except ValueError:
        print("Nomor harus berupa angka. Silakan coba lagi.")
        return login()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return login()
    except KeyboardInterrupt:
        print("Input tidak valid. Silakan coba lagi.")
        return login()

def menu_user(username):
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang di BELIBELI"]
        tabel.add_row(["[1] Order Makanan\n[2] Order Barang\n[3] Cek Voucher\n[4] Cek Keranjang\n[5] Lihat Saldo\n[6] Top Up\n[7] Keluar"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda : "))
            if opsi == 1:
                order_makanan(username)
            elif opsi == 2:
                order_barang(username)
            elif opsi == 3:
                lihat_voucher()
            elif opsi == 4:
                cek_keranjang(username)
            elif opsi == 5:
                cek_saldo(username)
            elif opsi == 6:
                top_up(username)
            elif opsi == 7:
                break
            else:
                print('input tidak valid')
                menu_user(username)
        except ValueError:
            print(" input harus berupa angka. Silakan coba lagi.")
            return menu_user(username)

```

```

def menu_admin():
    while True:
        tabel = PrettyTable()
        tabel.field_names = ["Selamat datang admin"]
        tabel.add_row(["[1] Menu Makanan\n[2] Menu Barang\n[3] Menu Voucher\n[4] Keluar Program\n"])
        print(tabel)
        try:
            opsi = int(input("Masukan Pilihan Anda: "))
            if opsi == 1:
                menu_makanan_admin()
            elif opsi == 2:
                menu_barang_admin()
            elif opsi == 3:
                menu_voucher()
            elif opsi == 4:
                print("Program berakhir")
                login()
                break
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            return menu_admin()
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            return menu_admin()
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return menu_admin()

```

```

def order_makanan(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_makanan()
        saldo = user["saldo"]
        data_makanan = load_data_makanan() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan makanan berdasarkan harga? (ya/tidak): ").strip()
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke ter")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]

```

```

    elif action == '2':
        # Menambahkan makanan ke keranjang
        tambah_ke_keranjang(username, item_to_order["nama_makanan"], item_to_order["harga"], quantity)
        print(f"{quantity} {item_to_order['nama_makanan']} telah ditambahkan ke keranjang.")
    else:
        print("Opsi tidak valid. Silakan coba lagi.")

    # Menanyakan apakah pengguna ingin memesan makanan lain
    continue_order = input("Apakah Anda ingin memesan makanan lain? (ya/tidak): ").strip().lower()
    if continue_order != 'ya':
        print("Terima kasih telah berbelanja!")
        return
except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi

# Menggunakan voucher
use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()
if use_voucher == 'ya':
    lihat_voucher()
    kode_voucher = input("Masukkan kode voucher: ")
    diskon, voucher = gunakan_voucher(kode_voucher)
    if diskon > 0:
        a = total_harga
        b = diskon
        total_harga_setelah_diskon = a - (a * b / 100)
        total_harga = total_harga_setelah_diskon
        print(f"Voucher digunakan! Diskon: {diskon}%, Total harga setelah diskon: {total_harga:.2f}")
        voucher['used_by'] = username # Update voucher status
        update_voucher_status(voucher) # Update voucher status di CSV
    else:
        print("Voucher tidak valid atau sudah digunakan.")

    if saldo < total_harga:
        print("Saldo Anda tidak cukup untuk membeli jumlah yang diminta.")
        continue # Kembali ke awal loop untuk mencoba lagi

# Memperbarui saldo dan stok
user["saldo"] -= total_harga
item_to_order["stok"] -= quantity
simpan_data_user_ke_csv(pengguna)
simpan_makanan_ke_csv(data_makanan)

```

```

while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)

        if not item_to_order:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            continue # Kembali ke awal loop untuk mencoba lagi

        if item_to_order["stok"] <= 0:
            print("Maaf, stok tidak tersedia untuk item ini.")
            continue # Kembali ke awal loop untuk mencoba lagi

        quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
        if quantity > item_to_order["stok"]:
            print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
            continue # Kembali ke awal loop untuk mencoba lagi

        # Pilihan untuk langsung beli atau masukkan ke keranjang
        action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")
        if action == '1':
            # Proses pembelian langsung
            total_harga = item_to_order["harga"] * quantity

        elif action == '2':
            # Menggunakan voucher
            use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()

            for makanan in data_makanan:
                tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
            print(tabel)

# Opsi pencarian
search_option = input("Apakah Anda ingin mencari makanan? (ya/tidak): ").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_makanan = search_item(data_barang, keyword)

# Menampilkan hasil pencarian
if data_makanan:
    print("Hasil pencarian makanan:")
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama makanan", "Harga", "Stok"]
    for barang in data_barang:
        tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
    print(tabel)
else:
    print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
    continue # Kembali ke awal loop untuk mencoba lagi

```

```

def order_barang(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_barang()
        saldo = user["saldo"]
        data_barang = load_data_barang() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan barang berdasarkan harga? (ya/tidak): ").strip()
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah: ")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan barang:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
        for barang in data_barang:
            tabel.add_row([barang["id"], barang["nama_barang"], barang["harga"], barang["stok"]])

        # Memperbarui saldo dan stok
        user["saldo"] -= total_harga
        item_to_order["stok"] -= quantity
        simpan_data_user_ke_csv(pengguna)
        simpan_barang_ke_csv(data_barang)

        print(f"Pembelian berhasil! Saldo Anda sekarang: Rp {user['saldo']:.2f}")

        invoice_barang(username, item_to_order["nama_barang"], quantity, total_harga)

    elif action == '2':
        # Menambahkan makanan ke keranjang
        tambah_ke_keranjang(username, item_to_order["nama_barang"], item_to_order["harga"], quantity)
        print(f"{quantity} {item_to_order['nama_barang']} telah ditambahkan ke keranjang.")
    else:
        print("Opsi tidak valid. Silakan coba lagi.")

    # Menanyakan apakah pengguna ingin memesan makanan lain
    continue_order = input("Apakah Anda ingin memesan barang lain? (ya/tidak): ").strip().lower()
    if continue_order != 'ya':
        print("Terima kasih telah berbelanja!")
        return

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi

```

```

# Menggunakan voucher
use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()
if use_voucher == 'ya':
    lihat_voucher()
    kode_voucher = input("Masukkan kode voucher: ")
    diskon, voucher = gunakan_voucher(kode_voucher)
    if diskon > 0:
        a = total_harga
        b = diskon
        total_harga_setelah_diskon = a - (a * b / 100)
        total_harga = total_harga_setelah_diskon
        print(f"Voucher digunakan! Diskon: {diskon}%, Total harga setelah diskon: {total_harga:.2f}")
        voucher['used_by'] = username # Update voucher status
        update_voucher_status(voucher) # Update voucher status di CSV
    else:
        print("Voucher tidak valid atau sudah digunakan.")

    if saldo < total_harga:
        print("Saldo Anda tidak cukup untuk membeli jumlah yang diminta.")
        continue # Kembali ke awal loop untuk mencoba lagi

while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((barang for barang in data_barang if barang["no"] == item_no), None)

        if not item_to_order:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            continue # Kembali ke awal loop untuk mencoba lagi

        if item_to_order["stok"] <= 0:
            print("Maaf, stok tidak tersedia untuk item ini.")
            continue # Kembali ke awal loop untuk mencoba lagi

        quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
        if quantity > item_to_order["stok"]:
            print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
            continue # Kembali ke awal loop untuk mencoba lagi

        # Pilihan untuk langsung beli atau masukkan ke keranjang
        action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")
        if action == '1':
            # Proses pembelian langsung
            total_harga = item_to_order["harga"] * quantity

```

```

        tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"], barang["stok"]])
    print(tabel)

# Opsi pencarian
search_option = input("Apakah Anda ingin mencari barang? (ya/tidak): ").strip().lower()
if search_option == "ya":
    keyword = input("Masukkan kata kunci pencarian: ")
    data_barang = search_item(data_barang, keyword)

# Menampilkan hasil pencarian
if data_barang:
    print("Hasil pencarian barang:")
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
    for barang in data_barang:
        tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"], barang["stok"]])
    print(tabel)
else:
    print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
    continue # Kembali ke awal loop untuk mencoba lagi

```

```

def save_keranjang(username, keranjang):
    existing_data = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            existing_data = list(reader)
    except FileNotFoundError:
        pass

    for item in keranjang:
        existing_data.append({
            "username": username,
            "nama_item": item["nama_item"],
            "total_harga": item["total_harga"],
            "status": item["status"]
        })

```

```
def load_keranjang(username):
    keranjang = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            for row in reader:
                if row['username'] == username:
                    keranjang.append({
                        "nama_item": row['nama_item'],
                        "total_harga": row['total_harga'],
                        "status": row['status']
                    })
    except FileNotFoundError:
        print("File keranjang tidak ditemukan.")
    return keranjang
```

```
def load_all_keranjang():
    keranjang_data = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            for row in reader:
                keranjang_data.append(row)
    except FileNotFoundError:
        print("File keranjang tidak ditemukan.")
    return keranjang_data
```

```

def tambah_barang():
    while True:
        data_barang = load_data_barang()
        no = len(data_barang) + 1
        try:
            nama_barang = input("Masukkan nama barang: ")
            harga = int(input("Masukkan harga barang: "))
            stok = int(input("Masukkan stok barang: "))

            barang_baru = {"no": no, "nama_barang": nama_barang, "harga": harga, "stok": stok}
            data_barang.append(barang_baru)

            simpan_barang_ke_csv(data_barang)
            print("Barang baru berhasil ditambahkan.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            return tambah_barang()
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            return tambah_barang()
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return tambah_barang()
        break

def update_barang():
    lihat_barang()
    data_barang = load_data_barang()

    try:
        item_no = int(input("Masukkan nomor barang yang ingin diupdate: "))
        item_to_update = next((barang for barang in data_barang if barang["no"] == item_no), None)
        if not item_to_update:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            return
        nama_baru_barang = input(f"Masukkan nama baru untuk {item_to_update['nama_barang']} : ")
        harga_baru_barang = input(f"Masukkan harga baru untuk {item_to_update['nama_barang']} : ")
        stok_baru_barang = input(f"Masukkan stok baru untuk {item_to_update['nama_barang']} : ")

        if nama_baru_barang:
            item_to_update["nama_barang"] = nama_baru_barang
        if harga_baru_barang.isdigit():
            item_to_update["harga"] = int(harga_baru_barang)
        if stok_baru_barang.isdigit():
            item_to_update["stok"] = int(stok_baru_barang)

        simpan_barang_ke_csv(data_barang)
        print("Barang berhasil diperbarui.")
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
        return update_barang()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return update_barang()

```

```

def hapus_barang():
    data_barang = load_data_barang()
    lihat_barang()
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dihapus: "))

        item_to_delete = next((barang for barang in data_barang if barang["no"] == item_no), None)
        if not item_to_delete:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            return
        data_barang = [barang for barang in data_barang if barang["no"] != item_no]
        for index, barang in enumerate(data_barang, start=1):
            barang["no"] = index
        simpan_barang_ke_csv(data_barang)
        print("Barang berhasil dihapus.")
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
        return hapus_barang()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return hapus_barang()
    except KeyboardInterrupt:
        print("Input tidak valid. Silakan coba lagi.")
        return hapus_barang()

```

```

# Langkah awal memperbaiki menu barang kelas Admin
def menu_barang_admin():
    while True:
        print("== Menu Barang ==")
        print("[1] Tambah Barang")
        print("[2] Lihat Barang")
        print("[3] Update Barang")
        print("[4] Hapus Barang")
        print("[5] Kembali")
        try:
            opsi = int(input("Masukkan pilihan anda : "))

            if opsi == 1:
                tambah_barang()
            elif opsi == 2:
                lihat_barang()
            elif opsi == 3:
                update_barang()
            elif opsi == 4:
                hapus_barang()
            elif opsi == 5:
                print("keluar dari menu barang")
                break
            else:
                print("Opsi tidak ada.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            menu_barang_admin()

```

```
except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    menu_barang_admin()
    return
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    menu_barang_admin()
    return

def menu_makanan():
    while True:
        print("== Menu Makanan ==")
        print("[1] Tambah Makanan")
        print("[2] Lihat Makanan")
        print("[3] Update Makanan")
        print("[4] Hapus Makanan")
        print("[5] Kembali")
        try:
            opsi = int(input("Masukkan pilihan anda : "))
            if opsi == 1:
                tambah_makanan()
            elif opsi == 2:
                lihat_makanan()
            elif opsi == 3:
                update_makanan()
            elif opsi == 4:
                hapus_makanan()
            elif opsi == 5:
                break
            else:
                print("Opsi tidak ada.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            menu_makanan_admin()
            return
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            menu_makanan_admin()
```

```

#FUNGSI UNTUK menambah menu makanan
def tambah_makanan():
    data_makanan = load_data_makanan()
    try:
        no = len(data_makanan) + 1

        nama_makanan = input("Masukkan nama makanan: ")
        harga = int(input("Masukkan harga makanan: "))
        stok = int(input("Masukkan stok makanan: "))

        makanan_baru = {"no": no, "nama_makanan": nama_makanan, "harga": harga, "stok": stok}
        data_makanan.append(makanan_baru)

        simpan_makanan_ke_csv(data_makanan)
        print("Makanan baru berhasil ditambahkan.")
    except:
        print("\nPerhatikan input")
        return tambah_makanan()

def update_makanan():
    while True:
        lihat_makanan()
        data_makanan = load_data_makanan()
        try:
            item_no = int(input("Masukkan nomor makanan yang ingin diupdate: "))
            item_to_update = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)
            if not item_to_update:
                print("Barang dengan nomor tersebut tidak ditemukan.")
                return
            nama_baru_makanan = input(f"Masukkan nama baru untuk {item_to_update['nama_makanan']} : ")
            harga_baru_makanan= input(f"Masukkan harga baru untuk {item_to_update['nama_makanan']} : ")
            stok_baru_makanan = input(f"Masukkan stok baru untuk {item_to_update['nama_makanan']} : ")

            if nama_baru_makanan:
                item_to_update["nama_makanan"] = nama_baru_makanan
            if harga_baru_makanan.isdigit():
                item_to_update["harga"] = int(harga_baru_makanan)
            if harga_baru_makanan.isdigit():
                item_to_update["stok"] = int(stok_baru_makanan)

            simpan_makanan_ke_csv(data_makanan)
            print("Barang berhasil diperbarui.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            return update_makanan()
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            return update_makanan()

```

```

def hapus_makanan():
    data_makanan = load_data_makanan()
    lihat_makanan()
    try:
        item_no = int(input("Masukkan nomor makanan yang ingin dihapus: "))

        item_to_delete = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)
        if not item_to_delete:
            print("makanan dengan nomor tersebut tidak ditemukan.")
            return
        data_makanan = [makanan for makanan in data_makanan if makanan["no"] != item_no]
        for index, makanan in enumerate(data_makanan, start=1):
            makanan["no"] = index
        simpan_makanan_ke_csv(data_makanan)
        print("makanan berhasil dihapus.")
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
        return hapus_makanan()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return hapus_makanan()
    except KeyboardInterrupt:
        print("Input tidak valid. Silakan coba lagi.")
        return hapus_makanan()

def tambah_voucher():
    while True:
        data_vouchers = load_data_voucher()
        try:
            kode = input("Masukkan kode voucher: ")
            diskon = input("Masukkan besaran diskon (dalam persen): ")

            if not diskon.isdigit():
                print("Diskon harus berupa angka.")
                return

            diskon = int(diskon)
            no = max([voucher["no"] for voucher in data_vouchers], default=0) + 1
            voucher_baru = {"no": no, "kode": kode, "diskon": diskon}
            data_vouchers.append(voucher_baru)

            simpan_voucher_ke_csv(data_vouchers)
            print("Voucher baru berhasil ditambahkan.")
            break # Keluar dari loop setelah berhasil menambah voucher
        except ValueError:
            print("Input tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except EOFError:
            print("\nInput tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return # Keluar dari fungsi jika ada interupsi

```

```

def update_voucher():
    while True:
        lihat_voucher() # Menampilkan daftar voucher
        data_vouchers = load_data_voucher() # Memuat data voucher
        try:
            item_no = input("Masukkan nomor voucher yang ingin diupdate: ").strip() # Menghapus spasi di depan dan di belakang
            print(f"Input yang diterima: '{item_no}'") # Debugging: Menampilkan input yang diterima

            if not item_no.isdigit():
                print("Input harus berupa angka. Silakan coba lagi.")
                continue # Kembali ke awal loop untuk mencoba lagi

            item_no = int(item_no) # Konversi ke integer
            print(f"Nomor voucher yang dicari: {item_no}") # Debugging: Menampilkan nomor yang dicari

            item_to_update = next((voucher for voucher in data_vouchers if int(voucher["no"]) == item_no),
                                  None)
            if not item_to_update:
                print("Voucher dengan nomor tersebut tidak ditemukan.")
                continue # Kembali ke awal loop untuk mencoba lagi

            # Meminta input baru untuk kode dan diskon
            kode_baru_voucher = input(f"Masukkan kode baru untuk {item_to_update['kode']} : ")
            diskon_baru_voucher = input(f"Masukkan diskon baru untuk {item_to_update['diskon']} : ")

def hapus_voucher():
    while True:
        try:
            lihat_voucher()
            data_vouchers = load_data_voucher()
            try:
                no = int(input("Masukkan nomor voucher yang ingin dihapus: "))
            except ValueError:
                print("Nomor harus berupa angka. Silakan coba lagi.")
                return
            voucher_to_delete = next((voucher for voucher in data_vouchers if voucher["no"] == no), None)

            if not voucher_to_delete:
                print("Voucher dengan nomor tersebut tidak ditemukan.")
                return
            data_vouchers = [voucher for voucher in data_vouchers if voucher["no"] != no]
            for index, voucher in enumerate(data_vouchers, start=1):
                voucher["no"] = index
            simpan_voucher_ke_csv(data_vouchers)
            print(f"Voucher dengan nomor {no} berhasil dihapus.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            hapus_voucher()
            return
        except EOFError:
            break

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    hapus_voucher()
    return
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    hapus_voucher()
    return
break

```

```

def menu_voucher():
    while True:
        print("== Menu Voucher ==")
        print("[1] Tambah Voucher")
        print("[2] Lihat Voucher")
        print("[3] Update Voucher")
        print("[4] Hapus Voucher")
        print("[5] Kembali")
        try:
            opsi = int(input("Masukkan pilihan anda : "))
            if opsi == 1:
                tambah_voucher()
            elif opsi == 2:
                lihat_voucher()
            elif opsi == 3:
                update_voucher()
            elif opsi == 4:
                hapus_voucher()
            elif opsi == 5:
                break
            else:
                print("Opsi tidak ada.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
    menu_voucher()

```

```

# Memperbarui kode jika ada input baru
if kode_baru_voucher:
    item_to_update["kode"] = kode_baru_voucher

# Memperbarui diskon jika ada input baru
if diskon_baru_voucher.isdigit():
    item_to_update["diskon"] = int(diskon_baru_voucher)

# Menyimpan kembali data voucher ke CSV
simpan_voucher_ke_csv(data_vouchers)
print(f"Voucher '{item_to_update['kode']}' berhasil diperbarui.")
break # Keluar dari loop setelah berhasil memperbarui

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi
except EOFError:
    print("\nInput tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return # Keluar dari fungsi jika ada interupsi

```

```

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    menu_voucher()
    return
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    menu_voucher()
    return

```

Pernyataan **try** dan **except** dalam pemrograman digunakan untuk menangani kesalahan atau pengecualian yang mungkin terjadi selama eksekusi kode. Dengan menggunakan **try**, programmer dapat menempatkan kode yang berpotensi menyebabkan kesalahan, sementara **except** memungkinkan penanganan kesalahan tersebut dengan cara yang terkontrol, sehingga program tidak berhenti secara tiba-tiba.

## 8. Date time

```

# Fungsi untuk menampilkan Invoice makanan
def invoice_makanan(username, nama_makanan, jumlah_makanan, total_harga):
    struk = PrettyTable()
    struk.field_names = ["Invoice"]
    struk.add_row([f"Nama Customer : {username}")]
    struk.add_row([f"Tanggal Transaksi : {datetime.now().strftime('%d-%m-%Y %H:%M:%S')}"])
    struk.add_row([f"Nama Produk : {nama_makanan}"])
    struk.add_row([f"Jumlah Item : {jumlah_makanan}"])
    struk.add_row([f"Total Bayar : {total_harga}"])
    struk.add_row(["Terima kasih sudah bertransaksi"])
    print(struk)

# Fungsi untuk menampilkan Invoice barang
def invoice_barang(username, nama_barang, jumlah_barang, total_harga):
    struk = PrettyTable()
    struk.field_names = ["Invoice"]
    struk.add_row([f"Nama Customer : {username}")]
    struk.add_row([f"Tanggal Transaksi : {datetime.now().strftime('%d-%m-%Y %H:%M:%S')}"])
    struk.add_row([f"Nama Produk : {nama_barang}"])
    struk.add_row([f"Jumlah Item : {jumlah_barang}"])
    struk.add_row([f"Total Bayar : {total_harga}"])
    struk.add_row(["Terima kasih sudah bertransaksi"])
    print(struk)

```

Modul **datetime** di Python ini berfungsi untuk menangani berbagai operasi yang berkaitan dengan tanggal dan waktu seperti invoice

## 9. Sorting, Searching, dan CRUD

```
def order_makanan(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_makanan()
        saldo = user["saldo"]
        data_makanan = load_data_makanan() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan makanan berdasarkan harga? (ya/tidak): ").strip()
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah: ")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
        for makanan in data_makanan:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)

search_option = input("Apakah Anda ingin mencari makanan? (ya/tidak): ").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_makanan = search_item(data_barang, keyword)

    # Menampilkan hasil pencarian
    if data_makanan:
        print("Hasil pencarian makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama makanan", "Harga", "Stok"]
        for barang in data_barang:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)
    else:
        print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
        continue # Kembali ke awal loop untuk mencoba lagi

# Memilih item untuk dipesan
while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)

        if not item_to_order:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            continue # Kembali ke awal loop untuk mencoba lagi
```

```

if item_to_order["stok"] <= 0:
    print("Maaf, stok tidak tersedia untuk item ini.")
    continue # Kembali ke awal loop untuk mencoba lagi

quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
if quantity > item_to_order["stok"]:
    print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
    continue # Kembali ke awal loop untuk mencoba lagi

# Pilihan untuk langsung beli atau masukkan ke keranjang
action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")
if action == '1':
    # Proses pembelian langsung
    total_harga = item_to_order["harga"] * quantity

    # Menggunakan voucher
    use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()
    if use_voucher == 'ya':
        lihat_voucher()
        kode_voucher = input("Masukkan kode voucher: ")
        diskon, voucher = gunakan_voucher(kode_voucher)
        if diskon > 0:
            a = total_harga
            b = diskon
            total_harga_setelah_diskon = a - (a * b / 100)
            total_harga = total_harga_setelah_diskon
            print(f"Voucher digunakan! Diskon: {diskon}%, Total harga setelah diskon: {total_harga:.2f}")
            voucher['used_by'] = username # Update voucher status
        else:
            print("Voucher tidak valid atau sudah digunakan.")

    if saldo < total_harga:
        print("Saldo Anda tidak cukup untuk membeli jumlah yang diminta.")
        continue # Kembali ke awal loop untuk mencoba lagi

    # Memperbarui saldo dan stok
    user["saldo"] -= total_harga
    item_to_order["stok"] -= quantity
    simpan_data_user_ke_csv(pengguna)
    simpan_makanan_ke_csv(data_makanan)

    print(f"Pembelian berhasil! Saldo Anda sekarang: Rp {user['saldo']:.2f}")

    invoice_makanan(username, item_to_order["nama_makanan"], quantity, total_harga)

elif action == '2':
    # Menambahkan makanan ke keranjang
    tambah_ke_keranjang(username, item_to_order["nama_makanan"], item_to_order["harga"], quantity)
    print(f"{quantity} {item_to_order['nama_makanan']} telah ditambahkan ke keranjang.")
else:
    print("Opsi tidak valid. Silakan coba lagi.")

# Menanyakan apakah pengguna ingin memesan makanan lain
continue_order = input("Apakah Anda ingin memesan makanan lain? (ya/tidak): ").strip().lower()
if continue_order != 'ya':
    print("Terima kasih telah berbelanja!")

```

```

def order_makanan(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_makanan()
        saldo = user["saldo"]
        data_makanan = load_data_makanan() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan makanan berdasarkan harga? (ya/tidak): ").strip()
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah: ")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
        for makanan in data_makanan:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)

        print(f"Pembelian berhasil! Saldo Anda sekarang: Rp {user['saldo']:.2f}")

        invoice_makanan(username, item_to_order["nama_makanan"], quantity, total_harga)

        elif action == '2':
            # Menambahkan makanan ke keranjang
            tambah_ke_keranjang(username, item_to_order["nama_makanan"], item_to_order["harga"], quantity)
            print(f"{quantity} {item_to_order['nama_makanan']} telah ditambahkan ke keranjang.")
        else:
            print("Opsi tidak valid. Silakan coba lagi.")

        # Menanyakan apakah pengguna ingin memesan makanan lain
        continue_order = input("Apakah Anda ingin memesan makanan lain? (ya/tidak): ").strip().lower()
        if continue_order != 'ya':
            print("Terima kasih telah berbelanja!")
            return
    except ValueError:
        print("Input tidak valid. Harap masukkan angka yang benar.")
        continue # Kembali ke awal loop untuk mencoba lagi

# Menggunakan voucher
use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()
if use_voucher == 'ya':
    lihat_voucher()
    kode_voucher = input("Masukkan kode voucher: ")
    diskon, voucher = gunakan_voucher(kode_voucher)
    if diskon > 0:
        a = total_harga
        b = diskon
        total_harga_setelah_diskon = a - (a * b / 100)
        total_harga = total_harga_setelah_diskon
        print(f"Voucher digunakan! Diskon: {diskon}%, Total harga setelah diskon: {total_harga:.2f}")
        voucher['used_by'] = username # Update voucher status
        update_voucher_status(voucher) # Update voucher status di CSV
    else:
        print("Voucher tidak valid atau sudah digunakan.")

if saldo < total_harga:
    print("Saldo Anda tidak cukup untuk membeli jumlah yang diminta.")
    continue # Kembali ke awal loop untuk mencoba lagi

# Memperbarui saldo dan stok
user["saldo"] -= total_harga
item_to_order["stok"] -= quantity
simpan_data_user_ke_csv(pengguna)
simpan_makanan_ke_csv(data_makanan)

```

```

# Memilih item untuk dipesan
while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)

        if not item_to_order:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            continue # Kembali ke awal loop untuk mencoba lagi

        if item_to_order["stok"] <= 0:
            print("Maaf, stok tidak tersedia untuk item ini.")
            continue # Kembali ke awal loop untuk mencoba lagi

        quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
        if quantity > item_to_order["stok"]:
            print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
            continue # Kembali ke awal loop untuk mencoba lagi

        # Pilihan untuk langsung beli atau masukkan ke keranjang
        action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")
        if action == '1':
            # Proses pembelian langsung
            total_harga = item_to_order["harga"] * quantity
    
```

---

```

# Opsi pencarian
search_option = input("Apakah Anda ingin mencari makanan? (ya/tidak): ").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_makanan = search_item(data_barang, keyword)

    # Menampilkan hasil pencarian
    if data_makanan:
        print("Hasil pencarian makanan:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama makanan", "Harga", "Stok"]
        for barang in data_barang:
            tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
        print(tabel)
    else:
        print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
        continue # Kembali ke awal loop untuk mencoba lagi

```

```

def order_barang(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username), None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_barang()
        saldo = user["saldo"]
        data_barang = load_data_barang() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan barang berdasarkan harga? (ya/tidak): ").strip()
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke ter")
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan barang:")
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
        for barang in data_barang:
            tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"], barang["stok"]])
        print(tabel)

        # Memperbarui saldo dan stok
        user["saldo"] -= total_harga
        item_to_order["stok"] -= quantity
        simpan_data_user_ke_csv(pengguna)
        simpan_barang_ke_csv(data_barang)

        print(f"Pembelian berhasil! Saldo Anda sekarang: Rp {user['saldo']:.2f}")

        invoice_barang(username, item_to_order["nama_barang"], quantity, total_harga)

    elif action == '2':
        # Menambahkan makanan ke keranjang
        tambah_ke_keranjang(username, item_to_order["nama_barang"], item_to_order["harga"], quantity)
        print(f"{quantity} {item_to_order['nama_barang']} telah ditambahkan ke keranjang.")
    else:
        print("Opsi tidak valid. Silakan coba lagi.")

    # Menanyakan apakah pengguna ingin memesan makanan lain
    continue_order = input("Apakah Anda ingin memesan barang lain? (ya/tidak): ").strip().lower()
    if continue_order != 'ya':
        print("Terima kasih telah berbelanja!")
        return

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi

```

```

# Menggunakan voucher
use_voucher = input("Apakah Anda ingin menggunakan voucher? (ya/tidak): ").strip().lower()
if use_voucher == 'ya':
    lihat_voucher()
    kode_voucher = input("Masukkan kode voucher: ")
    diskon, voucher = gunakan_voucher(kode_voucher)
    if diskon > 0:
        a = total_harga
        b = diskon
        total_harga_setelah_diskon = a - (a * b / 100)
        total_harga = total_harga_setelah_diskon
        print(f"Voucher digunakan! Diskon: {diskon}%, Total harga setelah diskon: {total_harga:.2f}")
        voucher['used_by'] = username # Update voucher status
        update_voucher_status(voucher) # Update voucher status di CSV
    else:
        print("Voucher tidak valid atau sudah digunakan.")

if saldo < total_harga:
    print("Saldo Anda tidak cukup untuk membeli jumlah yang diminta.")
    continue # Kembali ke awal loop untuk mencoba lagi

while True:
    try:
        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))
        item_to_order = next((barang for barang in data_barang if barang["no"] == item_no), None)

        if not item_to_order:
            print("Barang dengan nomor tersebut tidak ditemukan.")
            continue # Kembali ke awal loop untuk mencoba lagi

        if item_to_order["stok"] <= 0:
            print("Maaf, stok tidak tersedia untuk item ini.")
            continue # Kembali ke awal loop untuk mencoba lagi

        quantity = int(input("Masukkan jumlah yang ingin dibeli: "))
        if quantity > item_to_order["stok"]:
            print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")
            continue # Kembali ke awal loop untuk mencoba lagi

        # Pilihan untuk langsung beli atau masukkan ke keranjang
        action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")
        if action == '1':
            # Proses pembelian langsung
            total_harga = item_to_order["harga"] * quantity

```

```

# Opsi pencarian
search_option = input("Apakah Anda ingin mencari barang? (ya/tidak): ").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_barang = search_item(data_barang, keyword)

# Menampilkan hasil pencarian
if data_barang:
    print("Hasil pencarian barang:")
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
    for barang in data_barang:
        tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"], barang["stok"]])
    print(tabel)
else:
    print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")
    continue # Kembali ke awal loop untuk mencoba lagi

if not voucher_to_delete:
    print("Voucher dengan nomor tersebut tidak ditemukan.")
    return
data_vouchers = [voucher for voucher in data_vouchers if voucher["no"] != no]
for index, voucher in enumerate(data_vouchers, start=1):
    voucher["no"] = index
    simpan_voucher_ke_csv(data_vouchers)
    print(f"Voucher dengan nomor {no} berhasil dihapus.")
except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")
    hapus_voucher()
    return
except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    hapus_voucher()
    return
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    hapus_voucher()
    return
break

# Fungsi untuk menghapus data voucher
def hapus_voucher():
    while True:
        try:
            lihat_voucher()
            data_vouchers = load_data_voucher()
            try:
                no = int(input("Masukkan nomor voucher yang ingin dihapus: "))
            except ValueError:
                print("Nomor harus berupa angka. Silakan coba lagi.")
                return
            voucher_to_delete = next((voucher for voucher in data_vouchers if voucher["no"] == no), None)

```

```

# Memperbarui kode jika ada input baru
if kode_baru_voucher:
    item_to_update["kode"] = kode_baru_voucher

# Memperbarui diskon jika ada input baru
if diskon_baru_voucher.isdigit():
    item_to_update["diskon"] = int(diskon_baru_voucher)

# Menyimpan kembali data voucher ke CSV
simpan_voucher_ke_csv(data_vouchers)
print(f"Voucher '{item_to_update['kode']}' berhasil diperbarui.")
break # Keluar dari loop setelah berhasil memperbarui

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi
except EOFError:
    print("\nInput tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return # Keluar dari fungsi jika ada interupsi

def update_voucher():
    while True:
        lihat_voucher() # Menampilkan daftar voucher
        data_vouchers = load_data_voucher() # Memuat data voucher
        try:
            item_no = input("Masukkan nomor voucher yang ingin diupdate: ").strip() # Menghapus spasi di
            print(f"Input yang diterima: '{item_no}'") # Debugging: Menampilkan input yang diterima

            if not item_no.isdigit():
                print("Input harus berupa angka. Silakan coba lagi.")
                continue # Kembali ke awal loop untuk mencoba lagi

            item_no = int(item_no) # Konversi ke integer
            print(f"Nomor voucher yang dicari: {item_no}") # Debugging: Menampilkan nomor yang dicari

            item_to_update = next((voucher for voucher in data_vouchers if int(voucher["no"]) == item_no))
            if not item_to_update:
                print("Voucher dengan nomor tersebut tidak ditemukan.")
                continue # Kembali ke awal loop untuk mencoba lagi

            # Meminta input baru untuk kode dan diskon
            kode_baru_voucher = input(f"Masukkan kode baru untuk {item_to_update['kode']} : ")
            diskon_baru_voucher = input(f"Masukkan diskon baru untuk {item_to_update['diskon']} : ")

```

```

# Fungsi untuk memperbarui data voucher
def lihat_voucher():
    data_vouchers = load_data_voucher()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Kode Voucher", "Diskon (%)"]

    for voucher in data_vouchers:
        tabel.add_row([voucher["no"], voucher["kode"], voucher["diskon"]])
    print(tabel)

# Fungsi untuk menambah data voucher
def tambah_voucher():
    while True:
        data_vouchers = load_data_voucher()
        try:
            kode = input("Masukkan kode voucher: ")
            diskon = input("Masukkan besaran diskon (dalam persen): ")

            if not diskon.isdigit():
                print("Diskon harus berupa angka.")
                return

            diskon = int(diskon)
            no = max([voucher["no"] for voucher in data_vouchers], default=0) + 1
            voucher_baru = {"no": no, "kode": kode, "diskon": diskon}
            data_vouchers.append(voucher_baru)

            simpan_voucher_ke_csv(data_vouchers)
            print("Voucher baru berhasil ditambahkan.")
            break # Keluar dari loop setelah berhasil menambah voucher
        except ValueError:
            print("Input tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except EOFError:
            print("\nInput tidak valid. Harap masukkan angka yang benar.")
            continue # Kembali ke awal loop untuk mencoba lagi
        except KeyboardInterrupt:
            print("Input tidak valid. Silakan coba lagi.")
            return # Keluar dari fungsi jika ada interupsi

```

```

def hapus_makanan():
    data_makanan = load_data_makanan()
    lihat_makanan()
    try:
        item_no = int(input("Masukkan nomor makanan yang ingin dihapus: "))

        item_to_delete = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)
        if not item_to_delete:
            print("makanan dengan nomor tersebut tidak ditemukan.")
            return
        data_makanan = [makanan for makanan in data_makanan if makanan["no"] != item_no]
        for index, makanan in enumerate(data_makanan, start=1):
            makanan["no"] = index
        simpan_makanan_ke_csv(data_makanan)
        print("makanan berhasil dihapus.")
    except ValueError:
        print("Input harus berupa angka. Silakan coba lagi.")
        return hapus_makanan()
    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
        return hapus_makanan()
    except KeyboardInterrupt:
        print("Input tidak valid. Silakan coba lagi.")
        return hapus_makanan()

def update_makanan():
    while True:
        lihat_makanan()
        data_makanan = load_data_makanan()
        try:
            item_no = int(input("Masukkan nomor makanan yang ingin diupdate: "))
            item_to_update = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)
            if not item_to_update:
                print("Barang dengan nomor tersebut tidak ditemukan.")
                return
            nama_baru_makanan = input(f"Masukkan nama baru untuk {item_to_update['nama_makanan']} : ")
            harga_baru_makanan= input(f"Masukkan harga baru untuk {item_to_update['nama_makanan']} : ")
            stok_baru_makanan = input(f"Masukkan stok baru untuk {item_to_update['nama_makanan']} : ")

            if nama_baru_makanan:
                item_to_update["nama_makanan"] = nama_baru_makanan
            if harga_baru_makanan.isdigit():
                item_to_update["harga"] = int(harga_baru_makanan)
            if harga_baru_makanan.isdigit():
                item_to_update["stok"] = int(stok_baru_makanan)

            simpan_makanan_ke_csv(data_makanan)
            print("Barang berhasil diperbarui.")
        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
            return update_makanan()
        except EOFError:
            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
            return update_makanan()

# Fungsi untuk melihat menu makanan
def lihat_makanan():
    data_makanan = load_data_makanan()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Makanan", "Harga", "Stok"]

    for makanan in data_makanan:
        tabel.add_row([makanan["no"], makanan["nama_makanan"], makanan["harga"], makanan["stok"]])
    print(tabel)

```

```

#Fungsi untuk menambah menu makanan
def tambah_makanan():
    data_makanan = load_data_makanan()
    try:
        no = len(data_makanan) + 1

        nama_makanan = input("Masukkan nama makanan: ")
        harga = int(input("Masukkan harga makanan: "))
        stok = int(input("Masukkan stok makanan: "))

        makanan_baru = {"no": no, "nama_makanan": nama_makanan, "harga": harga, "stok": stok}
        data_makanan.append(makanan_baru)

        simpan_makanan_ke_csv(data_makanan)
        print("Makanan baru berhasil ditambahkan.")
    except:
        print("\nPerhatikan input")
        return tambah_makanan()

```

```

# Memperbarui kode jika ada input baru
if kode_baru_voucher:
    item_to_update["kode"] = kode_baru_voucher

# Memperbarui diskon jika ada input baru
if diskon_baru_voucher.isdigit():
    item_to_update["diskon"] = int(diskon_baru_voucher)

# Menyimpan kembali data voucher ke CSV
simpan_voucher_ke_csv(data_vouchers)
print(f"Voucher '{item_to_update['kode']}' berhasil diperbarui.")
break # Keluar dari loop setelah berhasil memperbarui

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi
except EOFError:
    print("\nInput tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return # Keluar dari fungsi jika ada interupsi

```

```

if not voucher_to_delete:
    print("Voucher dengan nomor tersebut tidak ditemukan.")
    return
data_vouchers = [voucher for voucher in data_vouchers if voucher["no"] != no]
for index, voucher in enumerate(data_vouchers, start=1):
    voucher["no"] = index
simpan_voucher_ke_csv(data_vouchers)
print(f"Voucher dengan nomor {no} berhasil dihapus.")
except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")
    hapus_voucher()
    return
except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    hapus_voucher()
    return
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    hapus_voucher()
    return
break

# Fungsi untuk menghapus data voucher
def hapus_voucher():
    while True:
        try:
            lihat_voucher()
            data_vouchers = load_data_voucher()
            try:
                no = int(input("Masukkan nomor voucher yang ingin dihapus: "))
            except ValueError:
                print("Nomor harus berupa angka. Silakan coba lagi.")
                return
            voucher_to_delete = next((voucher for voucher in data_vouchers if voucher["no"] == no), None)

```

Menggunakan CRUD sebagai wadah untuk admin/pengguna dalam membuat, merubah, melihat dan menghapus suatu data. Kami menggunakan searching untuk mencari data pada def order\_makanan dan def order\_barang dan menggunakan sorting untuk mengurutkan harga dari barang atau makanan

### 3.2 Alur Program

- Tampilan awal
  - Ketika user memilih opsi 1 : register

```

    === Selamat datang di BeliBeli ===
    [1] Register
    [2] Login
    [3] Keluar
    Masukkan pilihan anda : █

```

**Gambar 3. 1** Registrasi User

User akan diminta untuk input username dan password ketika berhasil akan tersimpan ke users.csv dan diarahkan untuk login ke dalam program.

Tampilan Ketika username sudah ada

```
Masukkan pilihan anda : 1
Masukkan username anda : dinda
Masukkan Password anda : dinda01
Username sudah ada, masukkan username lain
```

Gambar 3. 2 Kesalahan Registrasi User

- Ketika user memilih opsi 2 : login

```
Masukkan pilihan anda : 2
Masukkan username anda : dinda
Masukkan Password anda : *****
==> Login berhasil, Selamat datang dinda, selamat berbelanja ==>
```

Gambar 3. 3 Login User

User akan diminta untuk menginput username dan password yang sudah dibuat, ketika berhasil akan diarahkan untuk memilih layanan yang tersedia dalam program.

Tampilan Ketika input tidak sesuai atau belum memiliki akun

```
==> Selamat datang di BeliBeli ==
[1] Register
[2] Login
[3] Keluar
Masukkan pilihan anda : 2
Masukkan username anda : dinda
Masukkan Password anda : *****
==> Akun tidak ada atau password salah ==>
```

Gambar 3. 4 Kesalahan Login User

- Ketika user memilih opsi 3 : keluar

```
Masukkan pilihan anda : 3
==> Terima kasih telah menggunakan BeliBeli! ==>
```

Gambar 3. 5 Keluar Program

Akan mengarahkan user untuk keluar dari program dan program akan berhenti untuk dijalankan.

- Tampilan menu layanan pada user

```
== Login berhasil, Selamat datang dinda, selamat berbelanja ==
+-----+
| Selamat datang di BELIBELI |
+-----+
| [1] Order Makanan      |
| [2] Order Barang       |
| [3] Cek Voucher        |
| [4] Cek Keranjang      |
| [5] Cek Saldo          |
| [6] Top Up Saldo       |
| [7] Kembali             |
+-----+
Masukan Pilihan Anda : █
```

Gambar 3. 6 Menu Layanan User

### 1. Order makanan

Ketika user memilih opsi 1 : Order makanan

```
Masukan Pilihan Anda : 1
+-----+-----+-----+
| No | Nama Makanan | Harga | Stok |
+-----+-----+-----+
| 1 | mie ayam     | 15000 | 4   |
| 2 | Ayam penyet   | 10000 | 5   |
| 3 | Pecel Lele    | 13000 | 5   |
+-----+-----+-----+
Apakah Anda ingin mengurutkan makanan berdasarkan harga? (ya/tidak): ya
Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah: asc
Hasil pengurutan makanan:
+-----+-----+-----+
| No | Nama Makanan | Harga | Stok |
+-----+-----+-----+
| 2 | Ayam penyet   | 10000 | 5   |
| 3 | Pecel Lele    | 13000 | 5   |
| 1 | mie ayam     | 15000 | 4   |
+-----+-----+-----+
```

Gambar 3. 7 Order Makanan Fitur Sorting (asc)

```
Apakah Anda ingin mengurutkan makanan berdasarkan harga? (ya/tidak): ya
Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah: desc
Hasil pengurutan makanan:
+-----+-----+-----+
| No | Nama Makanan | Harga | Stok |
+-----+-----+-----+
| 1 | mie ayam     | 15000 | 3   |
| 2 | Ayam penyet   | 10000 | 4   |
+-----+-----+-----+
```

Gambar 3. 8 Order Makanan Fitur Sorting (desc)

User akan diminta untuk memilih apakah ingin mengurutkan makanan berdasarkan harga. Jika user memilih ya, maka program akan kembali memberikan pilihan apakah ingin mengurutkan berdasarkan harga termurah (asc) atau termahal (desc). Lalu program akan mengeksekusi sesuai inputan yang diberikan user.

```
Apakah Anda ingin mencari makanan? (ya/tidak): ya
Masukkan kata kunci pencarian: ayam
Hasil pencarian makanan:
+-----+-----+-----+
| No | Nama Makanan | Harga | Stok |
+-----+-----+-----+
| 2  | Ayam penyet   | 10000 | 5   |
| 1  | mie ayam      | 15000 | 4   |
+-----+-----+-----+
```

**Gambar 3. 9** Order Makanan Fitur Searching

Namun, apabila user memilih untuk tidak mengurutkan harga maka program akan langsung memberikan pilihan untuk ‘searching menu makanan’ yang tersedia. Jika user memilih ya, fitur searching akan memudahkan user mencari menu makanan yang diinginkan dengan memasukkan kata kunci. Jika memilih tidak, akan diarahkan untuk melakukan transaksi

```
Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): 1
Apakah Anda ingin menggunakan voucher? (ya/tidak): ya
+-----+
| No | Kode Voucher | Diskon (%) |
+-----+
| 1  | bca          | 9    |
| 2  | VOUCHER20    | 20   |
| 3  | VOUCHER15    | 15   |
| 4  | VOUCHER25    | 25   |
| 5  | VOUCHERS5    | 5    |
| 6  | VOUCHER30    | 30   |
| 7  | VOUCHER50    | 50   |
| 8  | VOUCHER35    | 35   |
| 9  | VOUCHER40    | 40   |
| 10 | VOUCH          | 5    |
+-----+
Masukkan kode voucher: bca
Voucher digunakan! Diskon: 9%, Total harga setelah diskon: 13650.00
Pembelian berhasil! Saldo Anda sekarang: Rp 199999999996739715072.00
```

**Gambar 3. 10** Order Makanan Transaksi Order Sekarang

Setelah itu user akan diberikan pilihan untuk bayar sekarang atau masukkan pesanan ke dalam keranjang. Apabila user memilih opsi 1 : Beli sekarang. user akan diberikan pilihan lagi apakah ingin menggunakan kode voucher atau tidak. Setelah memilih, program akan menampilkan invoice dari menu barang yang dibeli.

Setelah itu, program akan memberikan pilihan apakah ingin memesan makanan yang lainnya. Jika user memilih ya, program akan kembali menerima input untuk order makanan. Jika, tidak maka akan menampilkan output.

- Tampilan apabila voucher sudah tidak berlaku apabila sudah digunakan

```
Apakah Anda ingin menggunakan voucher? (ya/tidak): ya
+-----+
| No | Kode Voucher | Diskon (%) |
+-----+
| 1  | bca          | 9      |
| 2  | VOUCHER20    | 20     |
| 5  | VOUCHER5     | 5      |
| 6  | VOUCHER30    | 30     |
| 7  | VOUCHER50    | 50     |
| 8  | VOUCHER35    | 35     |
| 9  | VOUCHER40    | 40     |
| 10 | VOUCH          | 5      |
+-----+
Masukkan kode voucher: bca
Voucher tidak valid atau sudah digunakan.
Pembelian berhasil! Saldo Anda sekarang: Rp 485000.00
```

Gambar 3. 11 Order Makanan Fitur Voucher



Gambar 3. 12 Invoice order makanan

```

Masukkan nomor makanan yang ingin dibeli: 1
Masukkan jumlah yang ingin dibeli: 1
Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): 2
1 mie ayam telah ditambahkan ke keranjang.

```

**Gambar 3. 13 Order Makanan Masukkan Ke Keranjang**

Tetapi, jika kita memilih opsi 2 : masukkan ke keranjang. Maka menu makanan akan di masukkan ke dalam keranjang yang terhubung dengan csv.

## 2. Order barang

```

Masukan Pilihan Anda : 2
+---+-----+---+----+
| No | Nama Barang | Harga | Stok |
+---+-----+---+----+
| 4 | flatshoes | 55000 | 8 |
| 2 | gelang | 25000 | 7 |
| 3 | gelas | 35000 | 5 |
| 1 | kalung | 15000 | 7 |
+---+-----+---+----+
Apakah Anda ingin mengurutkan barang berdasarkan harga? (ya/tidak): ya
Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah: asc
Hasil pengurutan barang:
+---+-----+---+----+
| No | Nama Barang | Harga | Stok |
+---+-----+---+----+
| 1 | kalung | 15000 | 7 |
| 2 | gelang | 25000 | 7 |
| 3 | gelas | 35000 | 5 |
| 4 | flatshoes | 55000 | 8 |
+---+-----+---+----+

```

**Gambar 3. 14 Order barang sorting (asc)**

```

Apakah Anda ingin mengurutkan barang berdasarkan harga? (ya/tidak): ya
Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah: desc
Hasil pengurutan barang:
+---+-----+---+----+
| No | Nama Barang | Harga | Stok |
+---+-----+---+----+
| 4 | flatshoes | 55000 | 8 |
| 3 | gelas | 35000 | 5 |
| 2 | gelang | 25000 | 7 |
| 1 | kalung | 15000 | 6 |
+---+-----+---+----+

```

**Gambar 3. 15 Order Barang Sorting (desc)**

User akan diminta untuk memilih apakah ingin mengurutkan barang berdasarkan harga. Jika user memilih ya, maka program akan kembali memberikan pilihan apakah ingin mengurutkan berdasarkan harga termurah (asc) atau termahal (desc). Lalu program akan mengeksekusi sesuai inputan yang diberikan user.

```

Apakah Anda ingin mencari barang? (ya/tidak): ya
Masukkan kata kunci pencarian: gelang
Hasil pencarian barang:
+---+-----+-----+
| No | Nama Barang | Harga | Stok |
+---+-----+-----+
| 2 | gelang | 25000 | 7 |
+---+-----+-----+

```

**Gambar 3. 16 Order Barang Fitur Searching**

Namun, apabila user memilih untuk tidak mengurutkan harga maka program akan langsung memberikan pilihan untuk ‘searching’ menu barang yang tersedia. Jika user memilih ya, fitur searching akan memudahkan user mencari menu barang yang diinginkan dengan memasukkan kata kunci. Jika memilih tidak, akan diarahkan untuk melakukan transaksi

```

Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): 1
Apakah Anda ingin menggunakan voucher? (ya/tidak): ya
+---+-----+-----+
| No | Kode Voucher | Diskon (%) |
+---+-----+-----+
| 1 | bca | 9 |
| 2 | VOUCHER20 | 20 |
| 3 | VOUCHER15 | 15 |
| 4 | VOUCHER25 | 25 |
| 5 | VOUCHERS5 | 5 |
| 6 | VOUCHER30 | 30 |
| 7 | VOUCHER50 | 50 |
| 8 | VOUCHER35 | 35 |
| 9 | VOUCHER40 | 40 |
| 10 | VOUCH | 5 |
+---+-----+-----+
Masukkan kode voucher: VOUCHER20
Voucher digunakan! Diskon: 20%, Total harga setelah diskon: 8000000.00
Pembelian berhasil! Saldo Anda sekarang: Rp 199999999996731588608.00

```

**Gambar 3. 17 Order Barang Beli Sekarang**

Setelah itu user akan diberikan pilihan untuk bayar sekarang atau masukkan pesanan ke dalam keranjang. Apabila user memilih opsi 1 : Beli sekarang, user akan diberikan pilihan lagi apakah ingin menggunakan kode voucher atau tidak. Setelah memilih, program akan menampilkan invoice dari menu barang yang dibeli.

Setelah itu, program akan memberikan pilihan apakah ingin memesan barang yang lainnya. Jika user memilih ya, program akan kembali

menerima input untuk order barang. Jika, tidak maka akan menampilkan output.

- Tampilan apabila voucher sudah tidak berlaku atau sudah digunakan

```
Apakah Anda ingin menggunakan voucher? (ya/tidak): ya
+-----+
| No | Kode Voucher | Diskon (%) |
+-----+
| 1 | bca           | 9   |
| 2 | VOUCHER20    | 20  |
| 3 | VOUCHER15    | 15  |
| 4 | VOUCHER25    | 25  |
| 5 | VOUCHERS5    | 5   |
| 6 | VOUCHER30    | 30  |
| 7 | VOUCHER50    | 50  |
| 8 | VOUCHER35    | 35  |
| 9 | VOUCHER40    | 40  |
| 10| VOUCH          | 5   |
+-----+
Masukkan kode voucher: 10
Voucher tidak valid atau sudah digunakan.
Pembelian berhasil! Saldo Anda sekarang: Rp 135000.00
```

Gambar 3. 18 Order Barang Fitur Voucher

```
+-----+
|           Invoice           |
+-----+
| Nama Customer : asd        |
| Tanggal Transaksi : 10-11-24 06:47:38 |
| Nama Produk : Laptop       |
| Jumlah Item : 1            |
| Total Bayar : 8000000.0     |
| Terima kasih sudah bertransaksi |
+-----+
```

Gambar 3. 19 Invoice Order Barang

```
Masukkan nomor barang yang ingin dibeli: 2
Masukkan jumlah yang ingin dibeli: 1
Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): 2
1 gelang telah ditambahkan ke keranjang.
Apakah Anda ingin memesan barang lain? (ya/tidak): tidak
Terima kasih telah berbelanja!
```

Gambar 3. 20 Order Barang Masukkan Ke Keranjang

Tetapi, jika kita memilih opsi 2 : masukkan ke keranjang. Maka menu makanan akan di masukkan ke dalam keranjang yang terhubung dengan CSV.

### 3. Cek voucher

Masukan Pilihan Anda : 3		
No	Kode Voucher	Diskon (%)
1	bca	9
2	VOUCHER20	20
3	VOUCHER15	15
4	VOUCHER25	25
5	VOUCHERS5	5
6	VOUCHER30	30
7	VOUCHER50	50
8	VOUCHER35	35
9	VOUCHER40	40
10	VOUCH	5

Gambar 3. 21 Cek Voucher

Ketika user memilih opsi 3 : Cek voucher, akan menampilkan seluruh voucher yang tersedia.

### 4. Cek keranjang

Masukan Pilihan Anda : 4			
No	Nama Item	Total Harga	Status
1	mie ayam	15000	belum dibayar
2	Ayam penyet	10000	belum dibayar
3	gelang	25000	belum dibayar

Apakah Anda ingin checkout semua item di keranjang (ketik 'semua') atau memilih item tertentu (ketik 'pilih')? semua  
Checkout berhasil! Saldo Anda sekarang: Rp 10000

Gambar 3. 22 Cek Keranjang Check Out (semua)

Ketika user memilih opsi 4 : Cek keranjang dan memilih untuk check out ‘semua’, maka semua item pada keranjang akan segera terbeli dan saldo akan terpotong sesuai jumlah barang yang ada.

Masukan Pilihan Anda : 4			
No	Nama Item	Total Harga	Status
1	Ayam penyet	20000	belum dibayar
2	Ayam penyet	10000	belum dibayar

Apakah Anda ingin checkout semua item di keranjang (ketik 'semua') atau memilih item tertentu (ketik 'pilih')? pilih  
Masukkan nomor item yang ingin Anda checkout (0 untuk selesai): 1  
Masukkan nomor item yang ingin Anda checkout (0 untuk selesai): 0  
Checkout berhasil! Saldo Anda sekarang: Rp 115000

Gambar 3. 23 Cek Keranjang Check Out (pilih)

Ketika user menginput untuk ‘pilih’, user akan diminta untuk memilih nomor item yang ingin di check out. Kemudian, jika user selesai memilih item, inputkan 0 untuk membeli barang/makanan tersebut.

## 5. Cek saldo

```
Masukan Pilihan Anda : 5  
Saldo anda adalah = Rp 10000
```

Gambar 3. 24 Cek Saldo

Ketika user memilih opsi 5 : Cek saldo, akan menampilkan saldo yang dimiliki user saat ini.

## 6. Top up saldo

```
Masukan Pilihan Anda : 6  
Saldo anda saat ini adalah = Rp 10000  
Berapakah yang ingin anda topup? : Rp200000  
Saldo anda telah ditambahkan. Saldo sekarang = Rp 210000
```

Gambar 3. 25 Top Up Saldo

Ketika user memilih opsi 6 : Top up saldo, user akan diminta untuk mengisi nominal saldo yang ingin di tambahkan agar bisa membeli barang atau makanan dari layanan yang tersedia

## 7. Kembali

```
Masukan Pilihan Anda : 7  
== Selamat datang di BeliBeli ==  
[1] Register  
[2] Login  
[3] Keluar  
Masukkan pilihan anda : [
```

Gambar 3. 26 Kembali

Ketika user memilih opsi 7 : Kembali, user akan kembali ke menu login.

- Tampilan login sebagai admin

```
== Selamat datang di BeliBeli ===  
[1] Register  
[2] Login  
[3] Keluar  
Masukkan pilihan anda : 2  
Masukkan username anda : admin  
Masukkan Password anda : *****  
== Login berhasil, Selamat datang Admin ==
```

Gambar 3.27 27 Login sebagai Admin

Jika login sebagai admin, maka masukkan username dan password yang sudah fix sesuai gambar diatas.

```
== Login berhasil, Selamat datang Admin ==  
+-----+  
| Selamat datang admin |  
+-----+  
| [1] Menu Makanan |  
| [2] Menu Barang |  
| [3] Menu Voucher |  
| [4] Keluar Program |  
+-----+  
Masukan Pilihan Anda: |
```

Gambar 3. 28 Menu Layanan Admin

### 1. Menu makanan

```
Masukan Pilihan Anda: 1  
== Menu Makanan ==  
[1] Tambah Makanan  
[2] Lihat Makanan  
[3] Update Makanan  
[4] Hapus Makanan  
[5] Kembali  
Masukkan pilihan anda : |
```

Gambar 3.2929 Menu Makanan

- Ketika admin memilih opsi 1 : Tambah makanan

```
Masukkan pilihan anda : 1
Masukkan nama makanan: Nasi Goreng
Masukkan harga makanan: 20000
Masukkan stok makanan: 9
Makanan baru berhasil ditambahkan.
```

Gambar 3. 30 Create Makanan

Admin akan diarahkan menuju layanan yang mengatur menu makanan pada tampilan user. Untuk menambah makanan, admin akan diminta untuk menginput nama, harga, dan stok makanan, sehingga akan menambah pilihan makanan yang tersedia pada menu user.

- Ketika admin memilih opsi 2 : Lihat makanan

Masukkan pilihan anda : 2			
No	Nama Makanan	Harga	Stok
2	Ayam penyet	10000	3
1	mie ayam	15000	1
3	Nasi Goreng	20000	9

Gambar 3. 31 Read Makanan

Perintah untuk menampilkan pilihan menu makanan yang tersedia

- Ketika admin memilih opsi 3 : Update makanan

```
Masukkan pilihan anda : 3
+---+-----+-----+-----+
| No | Nama Makanan | Harga | Stok |
+---+-----+-----+-----+
| 2 | Ayam penyet | 10000 | 3 |
| 1 | mie ayam    | 15000 | 1 |
| 3 | Nasi Goreng | 20000 | 9 |
+---+-----+-----+-----+
Masukkan nomor makanan yang ingin diupdate: 1
Masukkan nama baru untuk mie ayam (tekan Enter untuk melewati): Steak daging sapi
Masukkan harga baru untuk mie ayam (tekan Enter untuk melewati): 30000
Masukkan stok baru untuk mie ayam (tekan Enter untuk melewati): 8
Barang berhasil diperbarui.
```

**Gambar 3. 32** Update makanan

Untuk mengupdate makanan, admin akan diminta untuk menginput nama, harga, stok makanan yang ingin di update, sehingga akan memperbaharui pilihan menu makanan yang tersedia pada menu user.

- Ketika admin memilih opsi 4 : Hapus makanan

```
Masukkan pilihan anda : 4
+---+-----+-----+-----+
| No | Nama Makanan | Harga | Stok |
+---+-----+-----+-----+
| 2 | Ayam penyet | 10000 | 3 |
| 1 | Steak daging sapi | 30000 | 8 |
| 3 | Nasi Goreng | 20000 | 9 |
+---+-----+-----+-----+
Masukkan nomor makanan yang ingin dihapus: 1
makanan berhasil dihapus.
```

**Gambar 3. 33** Delete Makanan

Untuk menghapus makanan, admin akan diminta untuk menginput nomor makanan yang ingin dihapus

## 2. Menu barang

```
==== Menu Barang ====
[1] Tambah Barang
[2] Lihat Barang
[3] Update Barang
[4] Hapus Barang
[5] Kembali
Masukkan pilihan anda : [REDACTED]
```

Gambar 3. 34 Menu barang

- Ketika admin memilih opsi 1 : Tambah barang

```
Masukkan pilihan anda : 1
Masukkan nama barang: sweater
Masukkan harga barang: 60000
Masukkan stok barang: 6
Barang baru berhasil ditambahkan.
```

Gambar 3. 35 Create barang

Untuk menambah barang, admin akan diminta untuk menginput nama, harga, dan stok barang yang ingin di update, sehingga akan menambah barang yang tersedia pada menu user

- Ketika admin memilih opsi 2 : Lihat barang

```
Masukkan pilihan anda : 2
+---+-----+-----+-----+
| No | Nama Barang | Harga | Stok |
+---+-----+-----+-----+
| 1 | sepatu      | 300000 | 2   |
| 2 | kaos        | 50000  | 4   |
| 3 | celana      | 80000  | 4   |
| 4 | jaket       | 100000 | 4   |
| 5 | Hoodie      | 55000  | 8   |
| 6 | sweater      | 60000  | 6   |
+---+-----+-----+-----+
```

Gambar 3. 36 Read barang

Perintah untuk menampilkan pilihan barang yang tersedia

- Ketika admin memilih opsi 3 : Update barang

```
Masukkan pilihan anda : 3
+---+-----+-----+-----+
| No | Nama Barang | Harga | Stok |
+---+-----+-----+-----+
| 1 | sepatu | 300000 | 2 |
| 2 | kaos | 50000 | 4 |
| 3 | celana | 80000 | 4 |
| 4 | jaket | 100000 | 4 |
| 5 | Hoodie | 55000 | 8 |
| 6 | sweater | 60000 | 6 |
+---+-----+-----+-----+
Masukkan nomor barang yang ingin diupdate: 6
Masukkan nama baru untuk sweater (tekan Enter untuk melewati): Cardingan
Masukkan harga baru untuk sweater (tekan Enter untuk melewati): 130000
Masukkan stok baru untuk sweater (tekan Enter untuk melewati): 7
Barang berhasil diperbarui.
```

**Gambar 3. 37** Update barang

Untuk mengupdate barang, admin akan diminta untuk menginput nama, harga, stok barang yang ingin di update, sehingga akan memperbarui pilihan barang yang tersedia pada menu user.

- Ketika admin memilih opsi 4 : Hapus barang

```
Masukkan pilihan anda : 4
+---+-----+-----+-----+
| No | Nama Barang | Harga | Stok |
+---+-----+-----+-----+
| 1 | sepatu | 300000 | 2 |
| 2 | kaos | 50000 | 4 |
| 3 | celana | 80000 | 4 |
| 4 | jaket | 100000 | 4 |
| 5 | Hoodie | 55000 | 8 |
| 6 | Cardingan | 130000 | 7 |
+---+-----+-----+-----+
Masukkan nomor barang yang ingin dihapus: 6
Barang berhasil dihapus.
```

**Gambar 3. 38** Delete barang

Untuk menghapus barang, admin akan diminta untuk menginput nomor barang yang ingin dihapus

### 3. Menu voucher

```
==== Menu Voucher ====
[1] Tambah Voucher
[2] Lihat Voucher
[3] Update Voucher
[4] Hapus Voucher
[5] Kembali
Masukkan pilihan anda :
```

Gambar 3. 39 Menu voucher

- Ketika admin memilih opsi 1 : Tambah Voucher

```
Masukkan pilihan anda : 1
Masukkan kode voucher: SPESIAL RAMADHAN
Masukkan besaran diskon (dalam persen): 25
Voucher baru berhasil ditambahkan.
```

Gambar 3. 40 Create voucher

Untuk menambah voucher, admin akan diminta untuk menginput kode dan besaran diskon yang nantinya akan diberikan kepada user

- Ketika admin memilih opsi 2 : Lihat Voucher

Masukkan pilihan anda : 2		
No	Kode Voucher	Diskon (%)
1	abc	5
2	VOUCHER20	20
3	VOUCHER15	15
4	VOUCHER25	25
5	VOUCHER5	5
6	VOUCHER30	30
7	VOUCHER50	50
8	VOUCHER35	35
9	VOUCHER40	40
10	VOUCH	5
11	abc	21

Gambar 3. 41 Read voucher

Perintah untuk menampilkan pilihan voucher yang tersedia

- Ketika admin memilih opsi 3 : Update Voucher

```

Masukkan pilihan anda : 3
+---+-----+-----+
| No | Kode Voucher | Diskon (%) |
+---+-----+-----+
| 1 | abc           | 5      |
| 2 | VOUCHER20     | 20     |
| 3 | VOUCHER15     | 15     |
| 4 | VOUCHER25     | 25     |
| 5 | VOUCHERS5     | 5      |
| 6 | VOUCHER30     | 30     |
| 7 | VOUCHER50     | 50     |
| 8 | VOUCHER35     | 35     |
| 9 | VOUCHER40     | 40     |
| 10| VOUCH           | 5      |
| 11| abc           | 21     |
+---+-----+-----+
Masukkan nomor voucher yang ingin diupdate: 1
Input yang diterima: '1'
Nomor voucher yang dicari: 1
Masukkan kode baru untuk abc (tekan Enter untuk melewati): bca
Masukkan diskon baru untuk 5 (tekan Enter untuk melewati): 9
Voucher 'bca' berhasil diperbarui.

```

**Gambar 3. 42** Update voucher

Untuk mengupdate voucher, admin akan diminta untuk menginput nomor voucher yang ingin di update, selanjutnya menginput kode dan diskon baru untuk memperbarui voucher yang tersedia.

- Ketika admin memilih opsi 4 : Hapus Voucher

```

Masukkan pilihan anda : 4
+---+-----+-----+
| No | Kode Voucher | Diskon (%) |
+---+-----+-----+
| 1 | bca           | 9      |
| 2 | VOUCHER20     | 20     |
| 3 | VOUCHER15     | 15     |
| 4 | VOUCHER25     | 25     |
| 5 | VOUCHERS5     | 5      |
| 6 | VOUCHER30     | 30     |
| 7 | VOUCHER50     | 50     |
| 8 | VOUCHER35     | 35     |
| 9 | VOUCHER40     | 40     |
| 10| VOUCH           | 5      |
| 11| abc           | 21     |
+---+-----+-----+
Masukkan nomor voucher yang ingin dihapus: 11
Voucher dengan nomor 11 berhasil dihapus.

```

**Gambar 3. 43** Delete voucher

Untuk menghapus voucher, admin akan diminta untuk menginput nomor voucher yang ingin di hapus, maka akan hilang dari data voucher yang tersedia.

#### 4. Keluar program

```
Masukan Pilihan Anda: 4  
Program berakhir
```

Gambar 3. 44 Keluar program

Program akan berhenti dijalankan.

### 3.3 Source Code

```
from prettytable import PrettyTable  
  
import csv  
  
import pwinput  
  
from datetime import datetime  
  
  
try:  
    try:  
        #===== Data dan menu  
        user/                                admin  
        =====  
  
        # Fungsi untuk memuat data user di file csv  
  
def data_user():  
    users = []  
  
    with open("user.csv", mode="r") as file:  
        reader = csv.DictReader(file)  
  
        for row in reader:  
            row["saldo"] = int(float(row["saldo"])) # Convert saldo to integer  
            users.append(row)  
  
    return users
```

```

# Fungsi untuk menyimpan data user ke file csv

def simpan_data_user_ke_csv(data_users):

    with open("user.csv", mode="w", newline= "") as file:

        writer = csv.DictWriter(file, fieldnames=["username", "password", "saldo"])

        writer.writeheader()

        for user in data_users:

            user["saldo"] = int(user["saldo"])

            writer.writerow(user)

# Fungsi untuk register dan login

def login():

    while True:

        print("==== Selamat datang di BeliBeli ====")

        print("[1] Register")

        print("[2] Login")

        print("[3] Keluar")

        try:

            opsi = int(input("Masukkan pilihan anda :"))

            if opsi == 1:

                username1 = input("Masukkan username anda :")

                if not username1:

                    print("Username tidak boleh kosong")

                    continue

                password1 = input("Masukkan Password anda :")

```

```

if not password1:
    print("Password tidak boleh kosong")
    continue

    user_sudah_ada = data_user()
    if any(user["username"] == username1 for user in user_sudah_ada):
        print("Username sudah ada, masukkan username lain")
        continue

    # Menambahkan pengguna baru
    tambah_user_baru = {"username": username1, "password": password1,
"saldo": "0"}
    user_sudah_ada.append(tambah_user_baru)
    simpan_data_user_ke_csv(user_sudah_ada)
    print("===== Akun anda telah dibuat =====")

elif opsi == 2:
    username2 = input("Masukkan username anda : ")
    password2 = pwininput.pwininput("Masukkan Password anda : ")
    user_sudah_ada = data_user()
    login_success = False

    if username2 == "admin" and password2 == "admin123":
        print("===== Login berhasil, Selamat datang Admin =====")
        menu_admin()
        login_success = True

```

```

else:
    for user in user_sudah_ada:
        if user["username"] == username2 and user["password"] ==
password2:
            print(f"==== Login berhasil, Selamat datang {username2}, selamat
berbelanja ===")
            login_success = True
            menu_user(username2)
            break

        if not login_success:
            print("==== Akun tidak ada atau password salah ===")

    elif opsi == 3:
        print("==== Terima kasih telah menggunakan BeliBeli! ===")
        break # Keluar dari fungsi login

else:
    print("Opsi tidak ada")

except ValueError:
    print("Nomor harus berupa angka. Silakan coba lagi.")
    return login()

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    return login()

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")

```

```

return login()

# Fungsi untuk menampilkan menu user

def menu_user(username):

    while True:

        tabel = PrettyTable()

        tabel.field_names = ["Selamat datang di BELIBELI"]

        tabel.add_row(["[1] Order Makanan\n[2] Order Barang\n[3] Cek Voucher\n[4] Cek Keranjang\n[5] Cek Saldo\n[6] Top Up Saldo\n[7] Kembali"])

        print(tabel)

    try:

        opsi = int(input("Masukan Pilihan Anda : "))

        if opsi == 1:

            order_makanan(username)

        elif opsi == 2:

            order_barang(username)

        elif opsi == 3:

            lihat_voucher()

        elif opsi == 4:

            cek_keranjang(username)

        elif opsi == 5:

            cek_saldo(username)

        elif opsi == 6:

            top_up(username)

        elif opsi == 7:

            break

    except ValueError:

```

```

        print(" i  nput harus berupa angka. Silakan coba lagi.")

    return menu_user(username)

except EOFError:

    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")

    return menu_user(username)

except KeyboardInterrupt:

    print("Input tidak valid. Silakan coba lagi.")

    return menu_user(username)

# Fungsi untuk menampilkan menu Admin

def menu_admin():

    while True:

        tabel = PrettyTable()

        tabel.field_names = ["Selamat datang admin"]

        tabel.add_row(["[1] Menu Makanan\n[2] Menu Barang\n[3] Menu Voucher\n[4] Keluar Program\n"])

        print(tabel)

        try:

            opsi = int(input("Masukan Pilihan Anda: "))

            if opsi == 1:

                menu_makanan_admin()

            elif opsi == 2:

                menu_barang_admin()

            elif opsi == 3:

                menu_voucher()

            elif opsi == 4:

                print("Program berakhir")

```

```

login()
break

except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")

    return menu_admin()

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")

    return menu_admin()

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")

    return menu_admin()

#=====
=====

#===== Fitur Sorting
dan
searching
=====

=


# Fungsi untuk sorting
def merge_sort(data, key, reverse=False):
    if len(data) <= 1:
        return data

    mid = len(data) // 2

```

```

left_half = merge_sort(data[:mid], key, reverse)
right_half = merge_sort(data[mid:], key, reverse)

return merge(left_half, right_half, key, reverse)

def merge(left, right, key, reverse):
    sorted_list = []
    while left and right:
        if (left[0][key] <= right[0][key] and not reverse) or (left[0][key] >
right[0][key] and reverse):
            sorted_list.append(left.pop(0))
        else:
            sorted_list.append(right.pop(0))

    sorted_list.extend(left)
    sorted_list.extend(right)

return sorted_list

def search_item(data, keyword):
    results = [
        item for item in data
        if (item.get("nama_makanan") and keyword.lower() in
item["nama_makanan"].lower()) or
            (item.get("nama_barang") and keyword.lower() in
item["nama_barang"].lower())
    ]

```

```

return results

#===== Fitur
User/Pengguna=====
=====

def order_makanan(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username),
        None)
        if not user:
            print("Pengguna tidak ditemukan.")
            return

        # Menampilkan daftar barang
        lihat_makanan()
        saldo = user["saldo"]
        data_makanan = load_data_makanan() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan makanan berdasarkan
harga? (ya/tidak): ").strip().lower()
        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc'
untuk termahal ke termurah: ").strip().lower()
            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

```

```

# Menampilkan hasil pengurutan
print("Hasil pengurutan makanan:")
tabel = PrettyTable()
tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
for makanan in data_makanan:
    tabel.add_row([makanan["no"], makanan["nama_makanan"],
makanan["harga"], makanan["stok"]])
print(tabel)

# Opsi pencarian
search_option = input("Apakah Anda ingin mencari makanan? (ya/tidak):")
").strip().lower()
if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_makanan = search_item(data_barang, keyword)

# Menampilkan hasil pencarian
if data_makanan:
    print("Hasil pencarian makanan:")
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama makanan", "Harga", "Stok"]
    for barang in data_barang:
        tabel.add_row([makanan["no"], makanan["nama_makanan"],
makanan["harga"], makanan["stok"]])
    print(tabel)
else:

```

```

print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")

continue # Kembali ke awal loop untuk mencoba lagi

# Memilih item untuk dipesan

while True:

    try:

        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))

        item_to_order = next((makanan for makanan in data_makanan if
makanan["no"] == item_no), None)

        if not item_to_order:

            print("Barang dengan nomor tersebut tidak ditemukan.")

            continue # Kembali ke awal loop untuk mencoba lagi

        if item_to_order["stok"] <= 0:

            print("Maaf, stok tidak tersedia untuk item ini.")

            continue # Kembali ke awal loop untuk mencoba lagi

        quantity = int(input("Masukkan jumlah yang ingin dibeli: "))

        if quantity > item_to_order["stok"]:

            print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")

            continue # Kembali ke awal loop untuk mencoba lagi

# Pilihan untuk langsung beli atau masukkan ke keranjang

action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")

if action == '1':

```

```

# Proses pembelian langsung
total_harga = item_to_order["harga"] * quantity

# Menggunakan voucher
use_voucher = input("Apakah Anda ingin menggunakan voucher?  

(ya/tidak): ").strip().lower()

if use_voucher == 'ya':
    lihat_voucher()
    kode_voucher = input("Masukkan kode voucher: ")
    diskon, voucher = gunakan_voucher(kode_voucher)
    if diskon > 0:
        a = total_harga
        b = diskon
        total_harga_setelah_diskon = a - (a * b / 100)
        total_harga = total_harga_setelah_diskon
        print(f"Voucher digunakan! Diskon: {diskon}%, Total harga  

setelah diskon: {total_harga:.2f}")

        voucher['used_by'] = username # Update voucher status
        update_voucher_status(voucher) # Update voucher status di
CSV

else:
    print("Voucher tidak valid atau sudah digunakan.")

if saldo < total_harga:
    print("Saldo Anda tidak cukup untuk membeli jumlah yang  

diminta.")

continue # Kembali ke awal loop untuk mencoba lagi

```

```

# Memperbarui saldo dan stok

user["saldo"] -= total_harga
item_to_order["stok"] -= quantity
simpan_data_user_ke_csv(pengguna)
simpan_makanan_ke_csv(data_makanan)

print(f"Pembelian berhasil! Saldo Anda sekarang: Rp
{user['saldo']:.2f}")

invoice_makanan(username, item_to_order["nama_makanan"],
quantity, total_harga)

elif action == '2':
# Menambahkan makanan ke keranjang
tambah_ke_keranjang(username, item_to_order["nama_makanan"],
item_to_order["harga"], quantity)
print(f"{quantity} {item_to_order['nama_makanan']} telah
ditambahkan ke keranjang.")

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

# Menanyakan apakah pengguna ingin memesan makanan lain
continue_order = input("Apakah Anda ingin memesan makanan lain?
(ya/tidak): ").strip().lower()
if continue_order != 'ya':
print("Terima kasih telah berbelanja!")

```

```

return

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")

    continue # Kembali ke awal loop untuk mencoba lagi

def order_barang(username):
    while True:
        pengguna = data_user()
        user = next((user for user in pengguna if user["username"] == username),
        None)

        if not user:
            print("Pengguna tidak ditemukan.")

            return

        # Menampilkan daftar barang
        lihat_barang()
        saldo = user["saldo"]
        data_barang = load_data_barang() # Memuat data barang dengan benar

        # Opsi pengurutan data
        sort_option = input("Apakah Anda ingin mengurutkan barang berdasarkan harga? (ya/tidak): ").strip().lower()

        if sort_option == 'ya':
            order_option = input("Ketik 'asc' untuk termurah ke termahal atau 'desc' untuk termahal ke termurah: ").strip().lower()

            reverse = True if order_option == 'desc' else False
            data_barang = merge_sort(data_barang, "harga", reverse)

```

```

# Menampilkan hasil pengurutan
print("Hasil pengurutan barang:")
tabel = PrettyTable()
tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
for barang in data_barang:
    tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"],
    barang["stok"]])
print(tabel)

# Opsi pencarian
search_option = input("Apakah Anda ingin mencari barang? (ya/tidak):")
").strip().lower()

if search_option == 'ya':
    keyword = input("Masukkan kata kunci pencarian: ")
    data_barang = search_item(data_barang, keyword)

# Menampilkan hasil pencarian
if data_barang:
    print("Hasil pencarian barang:")
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]
    for barang in data_barang:
        tabel.add_row([barang["no"], barang["nama_barang"],
        barang["harga"], barang["stok"]])
    print(tabel)
else:

```

```

print("Tidak ada barang yang ditemukan dengan kata kunci tersebut.")

continue # Kembali ke awal loop untuk mencoba lagi

# Memilih item untuk dipesan

while True:

    try:

        item_no = int(input("Masukkan nomor barang yang ingin dibeli: "))

        item_to_order = next((barang for barang in data_barang if barang["no"]
== item_no), None)

        if not item_to_order:

            print("Barang dengan nomor tersebut tidak ditemukan.")

            continue # Kembali ke awal loop untuk mencoba lagi

        if item_to_order["stok"] <= 0:

            print("Maaf, stok tidak tersedia untuk item ini.")

            continue # Kembali ke awal loop untuk mencoba lagi

        quantity = int(input("Masukkan jumlah yang ingin dibeli: "))

        if quantity > item_to_order["stok"]:

            print("Maaf, stok tidak mencukupi untuk jumlah yang diminta.")

            continue # Kembali ke awal loop untuk mencoba lagi

# Pilihan untuk langsung beli atau masukkan ke keranjang

action = input("Apakah Anda ingin (1) Beli Sekarang atau (2) Masukkan ke Keranjang? (masukkan 1 atau 2): ")

if action == '1':

```

```

# Proses pembelian langsung
total_harga = item_to_order["harga"] * quantity

# Menggunakan voucher
use_voucher = input("Apakah Anda ingin menggunakan voucher?  

(ya/tidak): ").strip().lower()

if use_voucher == 'ya':
    lihat_voucher()
    kode_voucher = input("Masukkan kode voucher: ")
    diskon, voucher = gunakan_voucher(kode_voucher)
    if diskon > 0:
        a = total_harga
        b = diskon
        total_harga_setelah_diskon = a - (a * b / 100)
        total_harga = total_harga_setelah_diskon
        print(f"Voucher digunakan! Diskon: {diskon}%, Total harga  

setelah diskon: {total_harga:.2f}")

        voucher['used_by'] = username # Update voucher status
        update_voucher_status(voucher) # Update voucher status di
CSV

else:
    print("Voucher tidak valid atau sudah digunakan.")

if saldo < total_harga:
    print("Saldo Anda tidak cukup untuk membeli jumlah yang  

diminta.")

continue # Kembali ke awal loop untuk mencoba lagi

```

```

# Memperbarui saldo dan stok

user["saldo"] -= total_harga

item_to_order["stok"] -= quantity

simpan_data_user_ke_csv(pengguna)

simpan_barang_ke_csv(data_barang)

print(f"Pembelian berhasil! Saldo Anda sekarang: Rp
{user['saldo']:.2f}")

invoice_barang(username, item_to_order["nama_barang"], quantity,
total_harga)

elif action == '2':

# Menambahkan makanan ke keranjang

tambah_ke_keranjang(username, item_to_order["nama_barang"],
item_to_order["harga"], quantity)

print(f"{quantity} {item_to_order['nama_barang']} telah ditambahkan
ke keranjang.")

else:

print("Opsi tidak valid. Silakan coba lagi.")

# Menanyakan apakah pengguna ingin memesan makanan lain

continue_order = input("Apakah Anda ingin memesan barang lain?
(ya/tidak): ").strip().lower()

if continue_order != 'ya':

print("Terima kasih telah berbelanja!")

```

```

return

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")

    continue # Kembali ke awal loop untuk mencoba lagi

# Fungsi untuk menampilkan Invoice makanan

def invoice_makanan(username, nama_makanan, jumlah_makanan,
total_harga):
    struk = PrettyTable()
    struk.field_names = ["Invoice"]
    struk.add_row([f"Nama Customer : {username}"])
    struk.add_row([f"Tanggal Transaksi : {datetime.now().strftime("%d-%m-%Y %H:%M:%S")}]])
    struk.add_row([f"Nama Produk : {nama_makanan}"])
    struk.add_row([f"Jumlah Item : {jumlah_makanan}"])
    struk.add_row([f"Total Bayar : {total_harga}"])
    struk.add_row(["Terima kasih sudah bertransaksi"])
    print(struk)

# Fungsi untuk menampilkan Invoice makanan

def invoice_barang(username, nama_barang, jumlah_barang, total_harga):
    struk = PrettyTable()
    struk.field_names = ["Invoice"]
    struk.add_row([f"Nama Customer : {username}"])
    struk.add_row([f"Tanggal Transaksi : {datetime.now().strftime("%d-%m-%Y %H:%M:%S")}]])
    struk.add_row([f"Nama Produk : {nama_barang}"])

```

```

struk.add_row([f"Jumlah Item : {jumlah_barang}"])
struk.add_row([f"Total Bayar : {total_harga}"])
struk.add_row(["Terima kasih sudah bertransaksi"])
print(struk)

# Fungsi untuk menampilkan data voucher

def lihat_voucher():

    data_vouchers = load_data_voucher()

    tabel = PrettyTable()

    tabel.field_names = ["No", "Kode Voucher", "Diskon (%)"]

    for voucher in data_vouchers:

        tabel.add_row([voucher["no"], voucher["kode"], voucher["diskon"]])

    print(tabel)

def cek_keranjang(username):

    keranjang = load_keranjang(username)

    if not keranjang:

        print("Keranjang Anda kosong.")

        return

    tabel = PrettyTable()

    tabel.field_names = ["No", "Nama Item", "Total Harga", "Status"]

    for index, item in enumerate(keranjang, start=1):

```

```

        tabel.add_row([index, item["nama_item"], item["total_harga"],
item["status"]])

    print(tabel)

    checkout_option = input("Apakah Anda ingin checkout semua item di keranjang
(ketik 'semua') atau memilih item tertentu (ketik 'pilih')? ").strip().lower()

    if checkout_option == 'semua':
        checkout_keranjang(username, keranjang)
    elif checkout_option == 'pilih':
        checkout_selected_items(username, keranjang)
    else:
        print("Opsi tidak valid. Silakan coba lagi.")

# Fungsi untuk menampilkan saldo

def cek_saldo(username):
    cek_saldo = data_user()
    for user in cek_saldo:
        if user["username"] == username:
            print(f"Saldo anda adalah = Rp {user['saldo']}")

# Fungsi untuk top-up saldo

def top_up(username):
    cek_saldo = data_user()
    for user in cek_saldo:
        if user["username"] == username:

```

```

print(f"Saldo anda saat ini adalah = Rp {user['saldo']}")

dana = input("Berapakah yang ingin anda topup? : Rp")

if dana.isdigit():

    dana = int(dana)

    if dana > 2000000:

        print("Top up melampaui batas maksimal, batas maksimal : Rp
2.000.000")

    else:

        user["saldo"] = int(user["saldo"]) + dana # Pastikan saldo tetap
sebagai integer

        simpan_data_user_ke_csv(cek_saldo)

        print(f"Saldo anda telah ditambahkan. Saldo sekarang = Rp
{user['saldo']}")

    else:

        print("Masukkan nominal yang valid.")

    break # Keluar dari loop setelah memproses pengguna

#===== Menu =====
Keranjang
=====
==

# Fungsi untuk memuat data makanan

def load_data_makanan():

    makanan = []

    with open("stok_makanan.csv", mode="r") as file:

        reader = csv.DictReader(file)

```

```

for row in reader:
    row["no"] = int(row["no"])
    row["harga"] = int(row["harga"])
    row["stok"] = int(row["stok"])
    makanan.append(row)

return makanan

# Fungsi untuk menyimpan data makanan ke CSV

def simpan_makanan_ke_csv(data_makanan):
    with open("stok_makanan.csv", mode="w", newline= "") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "nama_makanan", "harga",
        "stok"])
        writer.writeheader()
        for makanan in data_makanan:
            writer.writerow(makanan)

# Fungsi untuk memuat data barang

def load_data_barang():
    barang = []
    with open("stok_barang.csv", mode="r") as file:
        reader = csv.DictReader(file)
        for row in reader:
            row["no"] = int(row["no"])
            row["harga"] = int(row["harga"])
            row["stok"] = int(row["stok"])
            barang.append(row)

return barang

```

```

# Fungsi untuk menyimpan data barang ke CSV

def simpan_barang_ke_csv(data_barang):
    with open("stok_barang.csv", mode="w", newline= "") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "nama_barang", "harga",
        "stok"])
        writer.writeheader()
        for barang in data_barang:
            writer.writerow(barang)

# Fungsi untuk menyimpan data keranjang ke CSV

def simpan_keranjang_ke_csv(keranjang):
    with open("keranjang.csv", mode="w", newline= "") as file:
        writer = csv.DictWriter(file, fieldnames["username", "nama_item",
        "total_harga", "status"])
        writer.writeheader()
        for item in keranjang:
            writer.writerow(item)

def save_keranjang(username, keranjang):
    existing_data = []
    try:
        with open('keranjang.csv', mode='r', newline= '') as file:
            reader = csv.DictReader(file)
            existing_data = list(reader)
    except FileNotFoundError:
        pass

```

```

for item in keranjang:
    existing_data.append({
        "username": username,
        "nama_item": item["nama_item"],
        "total_harga": item["total_harga"],
        "status": item["status"]
    })

with open('keranjang.csv', mode='w', newline=") as file:
    fieldnames = ['username', 'nama_item', 'total_harga', 'status']
    writer = csv.DictWriter(file, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerows(existing_data)

def checkout_keranjang(username, keranjang):
    pengguna = data_user()
    user = next((user for user in pengguna if user["username"] == username), None)

    if not user:
        print("Pengguna tidak ditemukan.")
        return

    saldo = user["saldo"]
    total_belanja = sum(int(item["total_harga"]) for item in keranjang)

```

```

if saldo < total_belanja:
    print("Saldo Anda tidak cukup untuk checkout.")
    return

user["saldo"] -= total_belanja
for item in keranjang:
    item["status"] = "sudah dibayar"

simpan_data_user_ke_csv(pengguna)
simpan_keranjang_ke_csv(keranjang)
print(f"Checkout berhasil! Saldo Anda sekarang: Rp {user['saldo']}")

def tambah_ke_keranjang(username, nama_item, harga, quantity):
    keranjang = load_keranjang(username)
    total_harga = harga * quantity
    keranjang.append({
        "nama_item": nama_item,
        "total_harga": total_harga,
        "status": "belum dibayar"
    })
    save_keranjang(username, keranjang)

def checkout_selected_items(username, keranjang):
    pengguna = data_user()
    user = next((user for user in pengguna if user["username"] == username), None)

```

```

if not user:
    print("Pengguna tidak ditemukan.")

return

selected_items = []

while True:
    try:
        item_no = int(input("Masukkan nomor item yang ingin Anda checkout (0
untuk selesai): "))

        if item_no == 0:
            break

        if 1 <= item_no <= len(keranjang):
            selected_items.append(keranjang[item_no - 1])

        else:
            print("Nomor item tidak valid. Silakan coba lagi.")

    except ValueError:
        print("Input tidak valid. Silakan coba lagi.")

if not selected_items:
    print("Tidak ada item yang dipilih untuk checkout.")

return

saldo = user["saldo"]

total_belanja = sum(int(item["total_harga"]) for item in selected_items)

if saldo < total_belanja:
    print("Saldo Anda tidak cukup untuk checkout.")

return

user["saldo"] -= total_belanja

```

```

for item in selected_items:
    item["status"] = "sudah dibayar"

    simpan_data_user_ke_csv(pengguna)
    simpan_keranjang_ke_csv(keranjang)
    print(f"Checkout berhasil! Saldo Anda sekarang: Rp {user['saldo']}")

def load_keranjang(username):
    keranjang = []
    try:
        with open('keranjang.csv', mode='r', newline=") as file:
            reader = csv.DictReader(file)
            for row in reader:
                if row['username'] == username:
                    keranjang.append({
                        "nama_item": row['nama_item'],
                        "total_harga": row['total_harga'],
                        "status": row['status']
                    })
    except FileNotFoundError:
        print("File keranjang tidak ditemukan.")

    return keranjang

def simpan_keranjang_ke_csv(existing_data):
    with open("keranjang.csv", mode="w", newline= "") as file:
        writer = csv.DictWriter(file, fieldnames=["username", "nama_item",
        "total_harga", "status"])

```

```

writer.writeheader()

for item in existing_data:
    writer.writerow(item)

def save_keranjang(username, keranjang):
    existing_data = load_all_keranjang()

    existing_data = [item for item in existing_data if item['username'] != username]

    for item in keranjang:
        existing_data.append({
            "username": username,
            "nama_item": item["nama_item"],
            "total_harga": item["total_harga"],
            "status": item["status"]
        })

    simpan_keranjang_ke_csv(existing_data)

def load_all_keranjang():
    keranjang_data = []
    try:
        with open('keranjang.csv', mode='r', newline='') as file:
            reader = csv.DictReader(file)
            for row in reader:
                keranjang_data.append(row)
    except FileNotFoundError:

```

```

print("File keranjang tidak ditemukan.")

return keranjang_data


#=====
=====

=====

#=====

Menu Barang
Admin=====

=====

def load_data_barang():

    barang = []

    with open("stok_barang.csv", mode="r") as file:
        reader = csv.DictReader(file)

        for row in reader:
            row["no"] = int(row["no"])
            row["harga"] = int(row["harga"])
            row["stok"] = int(row["stok"])
            barang.append(row)

    return barang


# Fungsi untuk menyimpan data barang ke csv

def simpan_barang_ke_csv(data_barang):

    with open("stok_barang.csv", mode="w", newline= "") as file:

```

```

writer = csv.DictWriter(file, fieldnames=["no", "nama_barang", "harga",
"stok"])

writer.writeheader()

for barang in data_barang:
    writer.writerow(barang)

# Fungsi untuk menambah data barang

def tambah_barang():
    while True:
        data_barang = load_data_barang()
        no = len(data_barang) + 1
        try:
            nama_barang = input("Masukkan nama barang: ")
            harga = int(input("Masukkan harga barang: "))
            stok = int(input("Masukkan stok barang: "))

            barang_baru = {"no": no, "nama_barang": nama_barang, "harga": harga,
                           "stok": stok}
            data_barang.append(barang_baru)

            simpan_barang_ke_csv(data_barang)
            print("Barang baru berhasil ditambahkan.")

        except ValueError:
            print("Input harus berupa angka. Silakan coba lagi.")
        return tambah_barang()

    except EOFError:
        print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")

```

```

return tambah_barang()

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return tambah_barang()

break

# Fungsi untuk menampilkan data barang

def lihat_barang():
    data_barang = load_data_barang()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Barang", "Harga", "Stok"]

    for barang in data_barang:
        tabel.add_row([barang["no"], barang["nama_barang"], barang["harga"],
                     barang["stok"]])
    print(tabel)

# Fungsi untuk mengupdate data barang

def update_barang():
    lihat_barang()
    data_barang = load_data_barang()

    try:
        item_no = int(input("Masukkan nomor barang yang ingin diupdate: "))
        item_to_update = next((barang for barang in data_barang if barang["no"] ==
                             item_no), None)
        if not item_to_update:

```

```

print("Barang dengan nomor tersebut tidak ditemukan.")

return

    nama_baru_barang = input(f"Masukkan nama baru untuk
{item_to_update['nama_barang']} (tekan Enter untuk melewati): ")

    harga_baru_barang = input(f"Masukkan harga baru untuk
{item_to_update['nama_barang']} (tekan Enter untuk melewati): ")

    stok_baru_barang = input(f"Masukkan stok baru untuk
{item_to_update['nama_barang']} (tekan Enter untuk melewati): ")

if nama_baru_barang:
    item_to_update["nama_barang"] = nama_baru_barang
if harga_baru_barang.isdigit():
    item_to_update["harga"] = int(harga_baru_barang)
if stok_baru_barang.isdigit():
    item_to_update["stok"] = int(stok_baru_barang)

simpan_barang_ke_csv(data_barang)
print("Barang berhasil diperbarui.")

except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")
return update_barang()

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
return update_barang()

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
return update_barang()

```

```

# Fungsi untuk menghapus data barang

def hapus_barang():

    data_barang = load_data_barang()

    lihat_barang()

    try:

        item_no = int(input("Masukkan nomor barang yang ingin dihapus: "))

        item_to_delete = next((barang for barang in data_barang if barang["no"] ==
item_no), None)

        if not item_to_delete:

            print("Barang dengan nomor tersebut tidak ditemukan.")

            return

        data_barang = [barang for barang in data_barang if barang["no"] != item_no]

        for index, barang in enumerate(data_barang, start=1):

            barang["no"] = index

            simpan_barang_ke_csv(data_barang)

            print("Barang berhasil dihapus.")

        except ValueError:

            print("Input harus berupa angka. Silakan coba lagi.")

            return hpus_barang()

        except EOFError:

            print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")

            return hpus_barang()

        except KeyboardInterrupt:

            print("Input tidak valid. Silakan coba lagi.")

            return hpus_barang()

```

```

# Fungsi untuk menampilkan menu barang khusus admin

def menu_barang_admin():

    while True:

        print("==== Menu Barang ====")

        print("[1] Tambah Barang")

        print("[2] Lihat Barang")

        print("[3] Update Barang")

        print("[4] Hapus Barang")

        print("[5] Kembali")

        try:

            opsi = int(input("Masukkan pilihan anda :"))

            if opsi == 1:

                tambah_barang()

            elif opsi == 2:

                lihat_barang()

            elif opsi == 3:

                update_barang()

            elif opsi == 4:

                hapus_barang()

            elif opsi == 5:

                print("keluar dari menu barang")

                break

            else:

                print("Opsi tidak ada.")

        except ValueError:

```

```

print("Input harus berupa angka. Silakan coba lagi.")
menu_barang_admin()
return

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    menu_barang_admin()
    return

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    menu_barang_admin()
    return

#=====
==           Menu           Makanan           Admin
=====

# Fungsi untuk menampilkan menu makanan untuk admin

def menu_makanan():
    while True:
        print("==== Menu Makanan ====")
        print("[1] Tambah Makanan")
        print("[2] Lihat Makanan")
        print("[3] Update Makanan")
        print("[4] Hapus Makanan")
        print("[5] Kembali")

        try:
            opsi = int(input("Masukkan pilihan anda : "))

```

```

if opsi == 1:
    tambah_makanan()

elif opsi == 2:
    lihat_makanan()

elif opsi == 3:
    update_makanan()

elif opsi == 4:
    hapus_makanan()

elif opsi == 5:
    break

else:
    print("Opsi tidak ada.")

except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")
    menu_makanan_admin()
    return

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    menu_makanan_admin()
    return

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    menu_makanan_admin()
    return

# Fungsi untuk menampilkan menu makanan

def load_data_makanan():

```

```

makanan = []

with open("stok_makanan.csv", mode="r") as file:
    reader = csv.DictReader(file)
    for row in reader:
        row["no"] = int(row["no"])
        row["harga"] = int(row["harga"])
        row["stok"] = int(row["stok"])
        makanan.append(row)
    return makanan

# Fungsi untuk menyimpan menu makanan ke dalam CSV

def simpan_makanan_ke_csv(data_makanan):
    with open("stok_makanan.csv", mode="w", newline= "") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "nama_makanan", "harga",
        "stok"])
        writer.writeheader()
        for makanan in data_makanan:
            writer.writerow(makanan)

#Fungsi untuk menambah menu makanan

def tambah_makanan():
    data_makanan = load_data_makanan()
    try:
        no = len(data_makanan) + 1

        nama_makanan = input("Masukkan nama makanan: ")
        harga = int(input("Masukkan harga makanan: "))

```

```

stok = int(input("Masukkan stok makanan: "))

makanan_baru = {"no": no, "nama_makanan": nama_makanan, "harga": harga, "stok": stok}

data_makanan.append(makanan_baru)

simpan_makanan_ke_csv(data_makanan)
print("Makanan baru berhasil ditambahkan.")

except:
    print("\nPerhatikan input")
    return tambah_makanan()

# Fungsi untuk melihat menu makanan

def lihat_makanan():
    data_makanan = load_data_makanan()
    tabel = PrettyTable()
    tabel.field_names = ["No", "Nama Makanan", "Harga", "Stok"]

    for makanan in data_makanan:
        tabel.add_row([makanan["no"], makanan["nama_makanan"],
                      makanan["harga"], makanan["stok"]])
    print(tabel)

# Fungsi untuk mengupdate menu makanan

def update_makanan():
    while True:
        lihat_makanan()

```

```

data_makanan = load_data_makanan()

try:
    item_no = int(input("Masukkan nomor makanan yang ingin diupdate: "))

    item_to_update = next((makanan for makanan in data_makanan if
makanan["no"] == item_no), None)

    if not item_to_update:
        print("Barang dengan nomor tersebut tidak ditemukan.")

    return

    nama_baru_makanan = input(f"Masukkan nama baru untuk
{item_to_update['nama_makanan']} (tekan Enter untuk melewati): ")

    harga_baru_makanan= input(f"Masukkan harga baru untuk
{item_to_update['nama_makanan']} (tekan Enter untuk melewati): ")

    stok_baru_makanan = input(f"Masukkan stok baru untuk
{item_to_update['nama_makanan']} (tekan Enter untuk melewati): ")

    if nama_baru_makanan:
        item_to_update["nama_makanan"] = nama_baru_makanan

    if harga_baru_makanan.isdigit():
        item_to_update["harga"] = int(harga_baru_makanan)

    if stok_baru_makanan.isdigit():
        item_to_update["stok"] = int(stok_baru_makanan)

    simpan_makanan_ke_csv(data_makanan)
    print("Barang berhasil diperbarui.")

except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")

return update_makanan()

```

```

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    return update_makanan()

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return update_makanan()
    break

# Fungsi untuk mengupdate stok makanan

def update_stok(menu_makanan):
    with open("stok_makanan.csv", "w", newline= "") as file:
        fieldnames = ["no", "nama_makanan", "harga", "stok"]
        writer = csv.DictWriter(file, fieldnames=fieldnames)
        writer.writeheader()
        for menu in menu_makanan:
            writer.writerow({"no": menu["no"], "nama_makanan": menu["nama_makanan"], "harga": menu["harga"], "stok": menu["stok"]})

# Fungsi untuk menghapus menu makanan

def hapus_makanan():
    data_makanan = load_data_makanan()
    lihat_makanan()
    try:
        item_no = int(input("Masukkan nomor makanan yang ingin dihapus:"))

        item_to_delete = next((makanan for makanan in data_makanan if makanan["no"] == item_no), None)

```

```

if not item_to_delete:
    print("makanan dengan nomor tersebut tidak ditemukan.")

return

data_makanan = [makanan for makanan in data_makanan if makanan["no"]
!= item_no]

for index, makanan in enumerate(data_makanan, start=1):
    makanan["no"] = index
    simpan_makanan_ke_csv(data_makanan)
    print("makanan berhasil dihapus.")

except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")
    return hapus_makanan()

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    return hapus_makanan()

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return hapus_makanan()

#=====
# Fungsi untuk memuat data voucher
#=====

def load_data_voucher():

    data_vouchers = []

    with open('voucher.csv', mode='r') as file:

```

```

reader = csv.DictReader(file)

for row in reader:
    # Pastikan 'no' adalah integer
    row['no'] = int(row['no']) # Konversi ke integer
    row['diskon'] = int(row['diskon']) # Pastikan diskon juga integer
    data_vouchers.append(row)

return data_vouchers

# Fungsi untuk menggunakan voucher

def gunakan_voucher(kode_voucher):
    vouchers = load_data_voucher()
    for voucher in vouchers:
        if voucher['kode'] == kode_voucher and not voucher['used_by']:
            return int(voucher['diskon']), voucher
    return 0, None

def update_voucher_status(voucher_used):
    vouchers = load_data_voucher()
    for voucher in vouchers:
        if voucher['kode'] == voucher_used['kode']:
            voucher['used_by'] = voucher_used['used_by']

# Menyimpan kembali ke file CSV

with open("voucher.csv", mode="w", newline= "") as file:
    writer = csv.DictWriter(file, fieldnames=["no", "kode", "diskon",
    "used_by"])
    writer.writeheader()

```

```

writer.writerows(vouchers)

def simpan_voucher_ke_csv(data_vouchers):
    with open("voucher.csv", mode="w", newline= "") as file:
        writer = csv.DictWriter(file, fieldnames=["no", "kode", "diskon",
"used_by"])
        writer.writeheader()
        for voucher in data_vouchers:
            writer.writerow(voucher)

# Fungsi untuk menambah data voucher

def tambah_voucher():
    while True:
        data_vouchers = load_data_voucher()
        try:
            kode = input("Masukkan kode voucher: ")
            diskon = input("Masukkan besaran diskon (dalam persen): ")

            if not diskon.isdigit():
                print("Diskon harus berupa angka.")
                return

            diskon = int(diskon)
            no = max([voucher["no"] for voucher in data_vouchers], default=0) + 1
            voucher_baru = {"no": no, "kode": kode, "diskon": diskon}
            data_vouchers.append(voucher_baru)

```

```

simpan_voucher_ke_csv(data_vouchers)
print("Voucher baru berhasil ditambahkan.")

break # Keluar dari loop setelah berhasil menambah voucher

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")

    continue # Kembali ke awal loop untuk mencoba lagi

except EOFError:
    print("\nInput tidak valid. Harap masukkan angka yang benar.")

    continue # Kembali ke awal loop untuk mencoba lagi

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")

    return # Keluar dari fungsi jika ada interupsi

# Fungsi untuk menampilkan data voucher

def lihat_voucher():

    data_vouchers = load_data_voucher()

    tabel = PrettyTable()

    tabel.field_names = ["No", "Kode Voucher", "Diskon (%)"]

    for voucher in data_vouchers:
        tabel.add_row([voucher["no"], voucher["kode"], voucher["diskon"]])

    print(tabel)

# Fungsi untuk mengupdate data voucher

def update_voucher():

    while True:
        lihat_voucher() # Menampilkan daftar voucher

```

```

data_vouchers = load_data_voucher() # Memuat data voucher

try:
    item_no = input("Masukkan nomor voucher yang ingin diupdate:
").strip() # Menghapus spasi di awal dan akhir

    print(f"Input yang diterima: '{item_no}'") # Debugging: Menampilkan
input yang diterima

    if not item_no.isdigit():

        print("Input harus berupa angka. Silakan coba lagi.")

        continue # Kembali ke awal loop untuk mencoba lagi

    item_no = int(item_no) # Konversi ke integer

    print(f"Nomor voucher yang dicari: {item_no}") # Debugging:
Menampilkan nomor yang dicari

    item_to_update = next((voucher for voucher in data_vouchers if
int(voucher["no"]) == item_no), None)

    if not item_to_update:

        print("Voucher dengan nomor tersebut tidak ditemukan.")

        continue # Kembali ke awal loop untuk mencoba lagi

    # Meminta input baru untuk kode dan diskon

    kode_baru_voucher = input(f"Masukkan kode baru untuk
{item_to_update['kode']} (tekan Enter untuk melewati): ")

    diskon_baru_voucher = input(f"Masukkan diskon baru untuk
{item_to_update['diskon']} (tekan Enter untuk melewati): ")

```

```

# Memperbarui kode jika ada input baru
if kode_baru_voucher:
    item_to_update["kode"] = kode_baru_voucher

# Memperbarui diskon jika ada input baru
if diskon_baru_voucher.isdigit():
    item_to_update["diskon"] = int(diskon_baru_voucher)

# Menyimpan kembali data voucher ke CSV
simpan_voucher_ke_csv(data_vouchers)
print(f"Voucher '{item_to_update['kode']}' berhasil diperbarui.")

break # Keluar dari loop setelah berhasil memperbarui

except ValueError:
    print("Input tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi

except EOFError:
    print("\nInput tidak valid. Harap masukkan angka yang benar.")
    continue # Kembali ke awal loop untuk mencoba lagi

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    return # Keluar dari fungsi jika ada interupsi

# Fungsi untuk menghapus data voucher
def hapus_voucher():
    while True:
        try:

```

```

lihat_voucher()

data_vouchers = load_data_voucher()

try:

    no = int(input("Masukkan nomor voucher yang ingin dihapus: "))

except ValueError:

    print("Nomor harus berupa angka. Silakan coba lagi.")

return

    voucher_to_delete = next((voucher for voucher in data_vouchers if
voucher["no"] == no), None)

if not voucher_to_delete:

    print("Voucher dengan nomor tersebut tidak ditemukan.")

return

data_vouchers = [voucher for voucher in data_vouchers if voucher["no"]
!= no]

for index, voucher in enumerate(data_vouchers, start=1):

    voucher["no"] = index

    simpan_voucher_ke_csv(data_vouchers)

    print(f"Voucher dengan nomor {no} berhasil dihapus.")

except ValueError:

    print("Input harus berupa angka. Silakan coba lagi.")

    hapus_voucher()

return

except EOFError:

    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")

    hapus_voucher()

return

```

```

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    hapus_voucher()
    return
break

# Fungsi untuk menampilkan menu voucher untuk admin

def menu_voucher():
    while True:
        print("==== Menu Voucher ====")
        print("[1] Tambah Voucher")
        print("[2] Lihat Voucher")
        print("[3] Update Voucher")
        print("[4] Hapus Voucher")
        print("[5] Kembali")

        try:
            opsi = int(input("Masukkan pilihan anda : "))

            if opsi == 1:
                tambah_voucher()
            elif opsi == 2:
                lihat_voucher()
            elif opsi == 3:
                update_voucher()
            elif opsi == 4:
                hapus_voucher()
            elif opsi == 5:

```

```
break
else:
    print("Opsi tidak ada.")

except ValueError:
    print("Input harus berupa angka. Silakan coba lagi.")
    menu_voucher()
    return

except EOFError:
    print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")
    menu_voucher()
    return

except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
    menu_voucher()
    return

login()

except ValueError:
    print("Input tidak valid. Silakan coba lagi")
except KeyboardInterrupt:
    print("Input tidak valid. Silakan coba lagi.")
```



## **BAB IV**

## **PENUTUP**

### **4.1 Kesimpulan**

Proyek e-commerce ini menunjukkan bahwa platform digital dapat meningkatkan efisiensi transaksi jual beli dengan mengintegrasikan fitur seperti pengelolaan produk, manajemen stok, penggunaan voucher, dan mekanisme pembayaran yang aman. Melalui implementasi yang tepat, sistem ini mampu menawarkan pengalaman berbelanja yang nyaman bagi pengguna, sekaligus membantu penjual dalam mengoptimalkan proses bisnis mereka. Keunggulan lain dari platform ini adalah kemampuannya untuk menyesuaikan fitur-fitur dengan kebutuhan pasar, seperti personalisasi penawaran dan pengaturan promosi. Dengan demikian, proyek ini membuktikan bahwa solusi e-commerce yang baik mampu mendorong pertumbuhan bisnis secara signifikan.

### **4.2 Saran**

1. Analisis Kinerja Platform: Jelaskan bagaimana platform e-commerce diuji dan hasil performa yang diperoleh, seperti kecepatan akses, responsivitas, dan stabilitas sistem.
2. Evaluasi Pengalaman Pengguna (User Experience): Diskusikan umpan balik dari pengguna yang mencoba platform, termasuk aspek antarmuka, navigasi, dan kemudahan penggunaan.
3. Keamanan Data: Soroti langkah-langkah keamanan yang diambil untuk melindungi data pengguna, seperti enkripsi dan pengelolaan akses.
4. Perbandingan dengan Platform Serupa: Berikan perbandingan singkat dengan platform e-commerce lain di pasar, menunjukkan keunggulan atau kelemahan dari proyek ini.
5. Potensi Pengembangan di Masa Depan: Usulkan fitur tambahan atau peningkatan yang dapat meningkatkan fungsionalitas platform, seperti

sistem rekomendasi berbasis AI atau integrasi pembayaran dengan mata uang digital.

## **DAFTAR PUSTAKA**

## LAMPIRAN

Tabel Kontribusi

Nama	Kontribusi	Bagian
Muhammad Rizky Febrianto (2409116045)	Konsep, Coding, Flowchart, Pengecekan	1. Logika program 2. Coding 3. Flowchart Program 4. Mencari bug dan <i>error</i>
Dinda Aulia Rizky (2409116076)	Coding, Flowchart, Laporan	1. Referensi konsep program 2. Flowchart program 3. Coding 4. Konsep dan Penyusunan Laporan
Abdurrahman Al Farissy (2409116055)	Konsep, Flowchart, Pengembangan	1. Konsep program 2. Penyusunan Flowchart 3. Pengembangan program 4. Mengecek alur program

**Tabel 1.1** Tabel Kontribusi