

JUDUL LAPORAN



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 pengerjaan 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

Penyusun

Daftar Isi

Kata Pengantar	iii
Daftar Isi	iv
Daftar Gambar	v
Daftar Tabel.....	v
BAB I.....	7
1.1 Deskripsi Masalah	7
1.2 Rumusan Masalah	7
1.3 Batasan Masalah	8
1.4 Tujuan	8
1.5 Manfaat	8
BAB II.....	9
2.1 Analisis Program	9
2.2 Flowchart.....	11
BAB III.....	23
3.1 Implementasi Program	23
3.2 Alur Program.....	23
3.3 Source Code	31
BAB IV	81
4.1 Kesimpulan	81
4.2 Saran.....	81
Daftar Pustaka.....	82
Lampiran.....	83

Daftar Gambar

Gambar 2. 1 Flowchart Login	11
Gambar 2. 2 Flowchart Menu User.....	12
Gambar 2. 3 Flowchart Menu Makanan (User)	13
Gambar 2. 4 Flowchart Opsi Order Makanan (User)	14
Gambar 2. 5 Flowchart Menu Barang (User)	15
Gambar 2. 6 Flowchart Opsi Order Barang (User).....	16
Gambar 2. 7 Flowchart Cek Keranjang (User)	17
Gambar 2. 8 Flowchart Menu Admin	18
Gambar 2. 9 Flowchart Menu Admin	19
Gambar 2. 10 Flowchart Menu Makanan (Admin)	20
Gambar 2. 11 Flowchart Menu Barang (Admin).....	21
Gambar 2. 12 Flowchart Menu Voucher (Admin).....	22

Daftar Tabel

Tabel 3.1 Source Code	53
Tabel 1.1 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 python 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 python 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 masih merupakan program sederhana
3. Program ini dibuat menggunakan bahasa pemrograman Python

Commented [PA1]: Ubah kalimatnya ttg batasan masalah kita

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 mengantri
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 layanan dalam bentuk tabel agar memiliki tampilan yang rapi dan 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 menginput atau tidak sesuai dengan ketentuan, maka akan terjadi perulangan. “While True” biasanya diakhiri dengan “break” untuk mengakhiri perulangan apabila diinginkan.

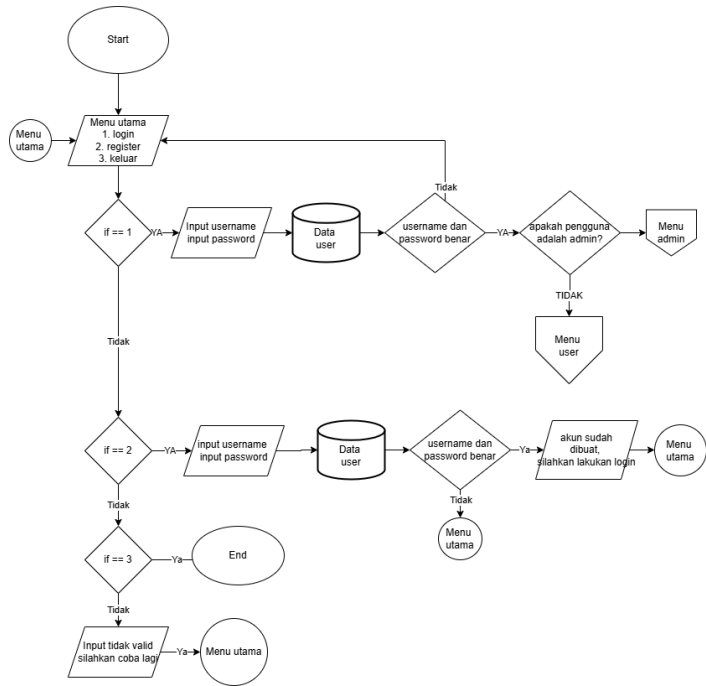
7. Try Except

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

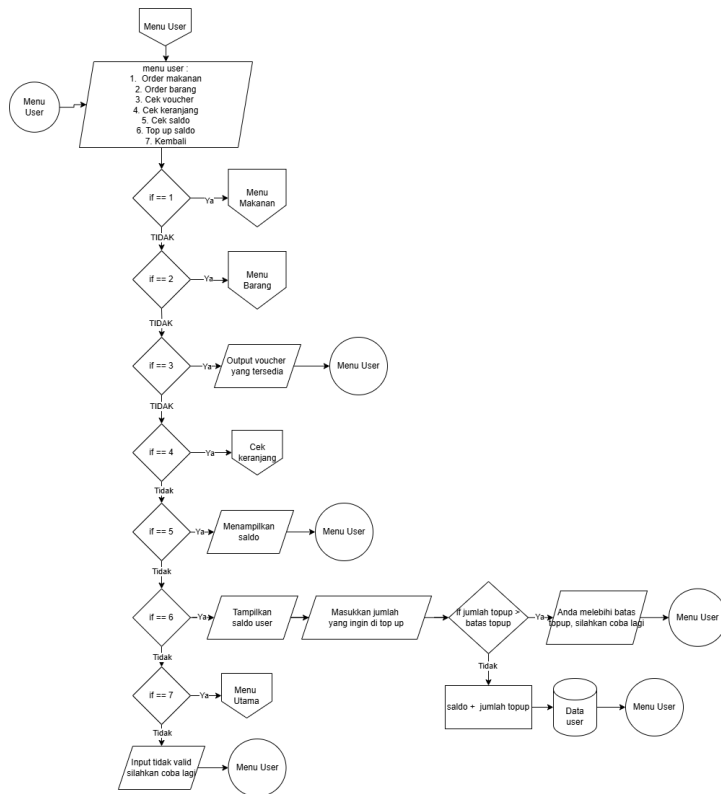
8. Date time

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

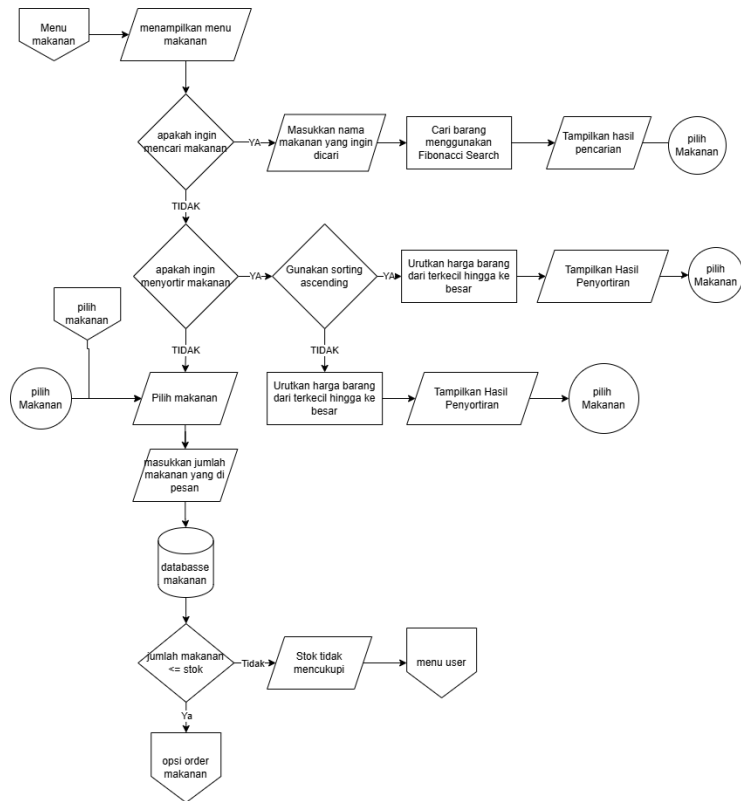
2.2 Flowchart



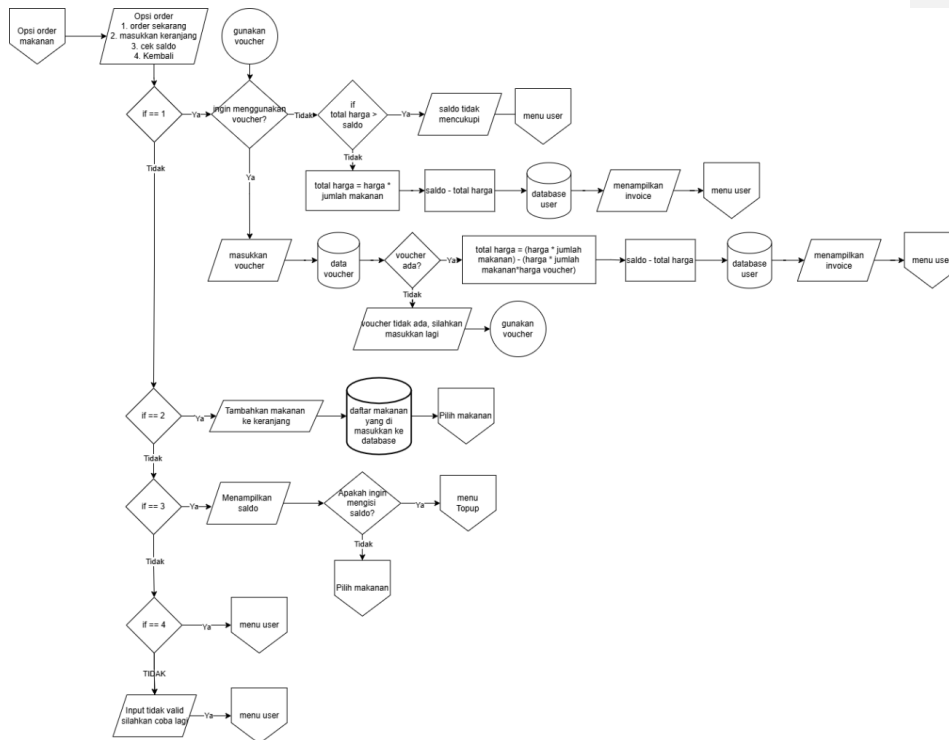
Gambar 2.1 Flowchart Login



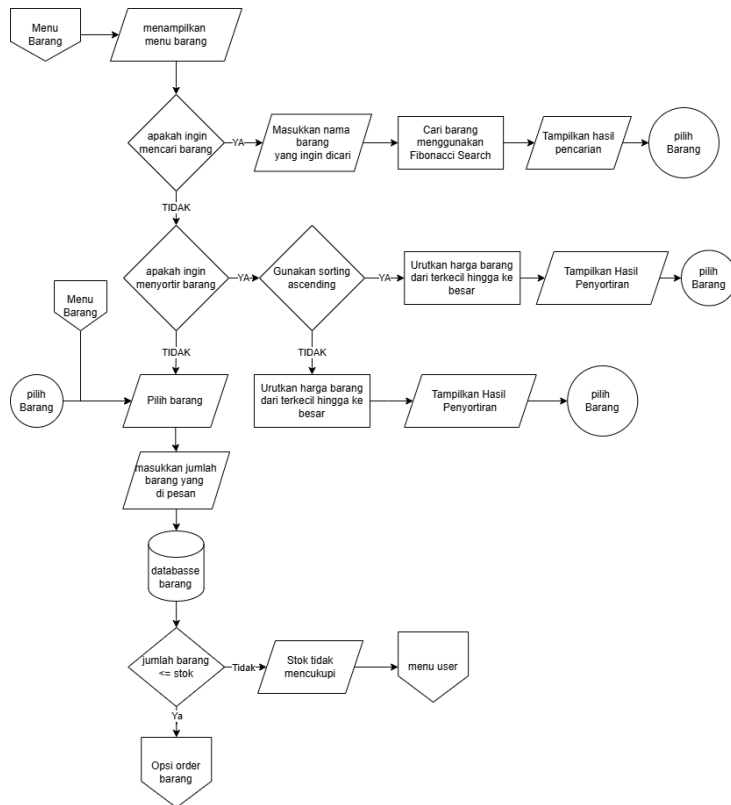
Gambar 2. 2 Flowchart Menu User



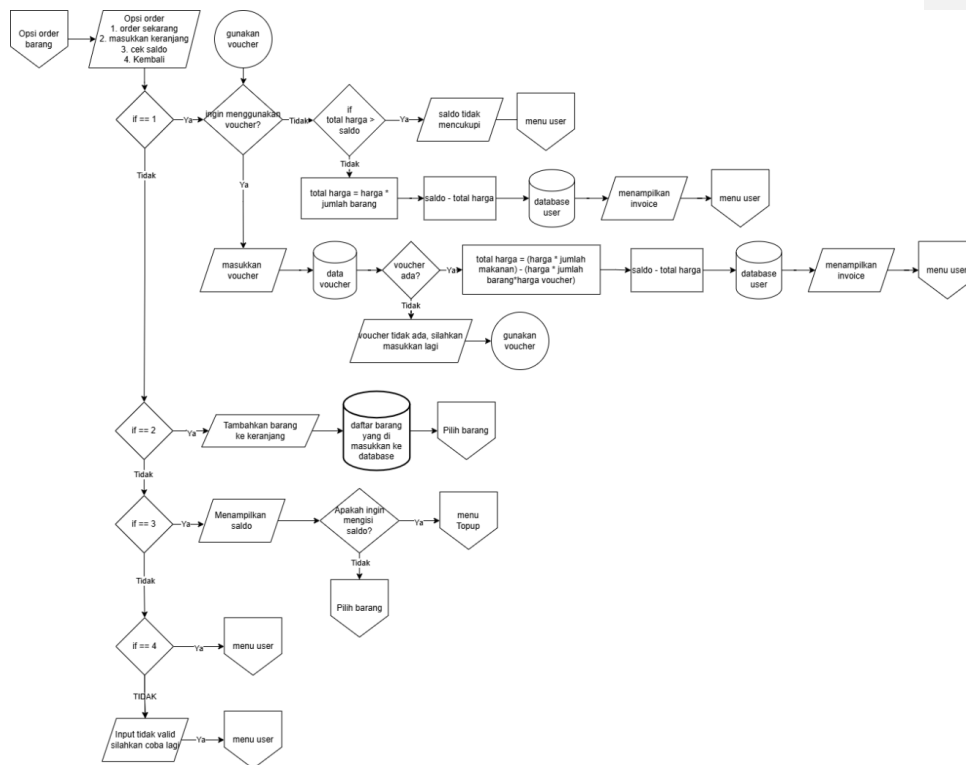
Gambar 2. 3 Flowchart Menu Makanan (User)



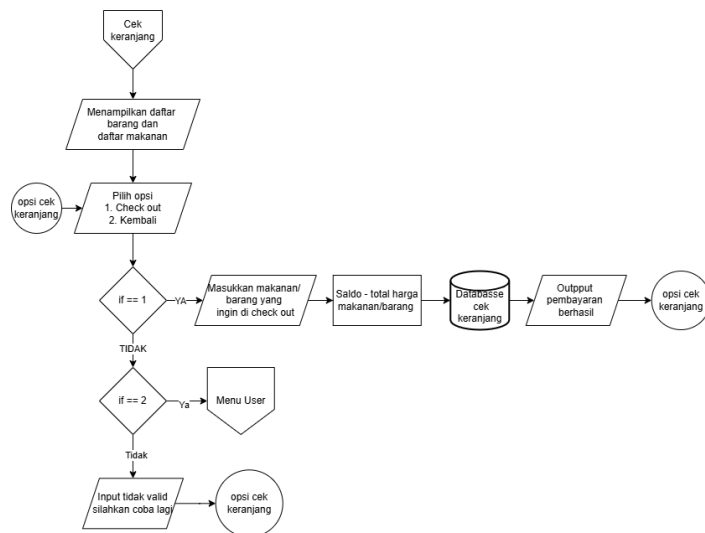
Gambar 2. 4 Flowchart Opsi Order Makanan (User)



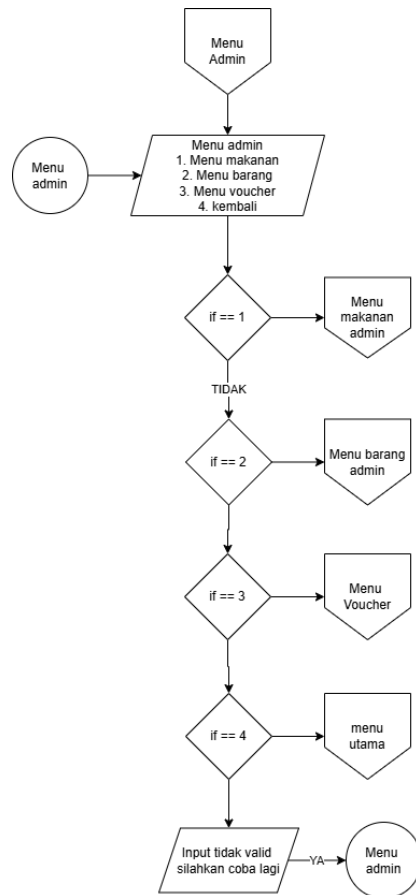
Gambar 2. 5 Flowchart Menu Barang (User)



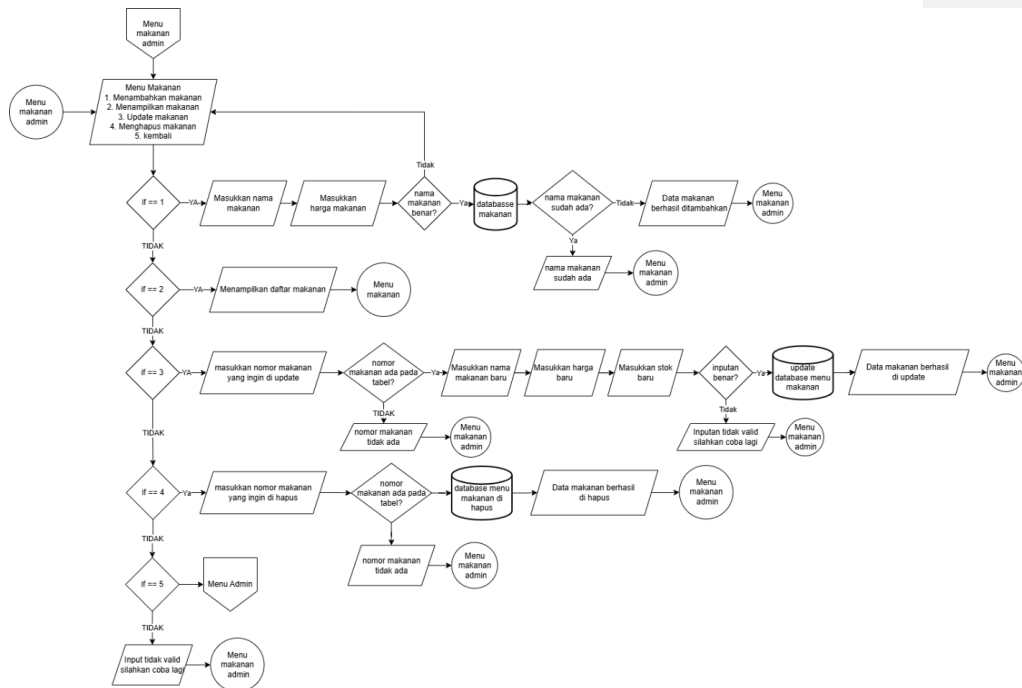
Gambar 2. 6 Flowchart Opsi Order Barang (User)



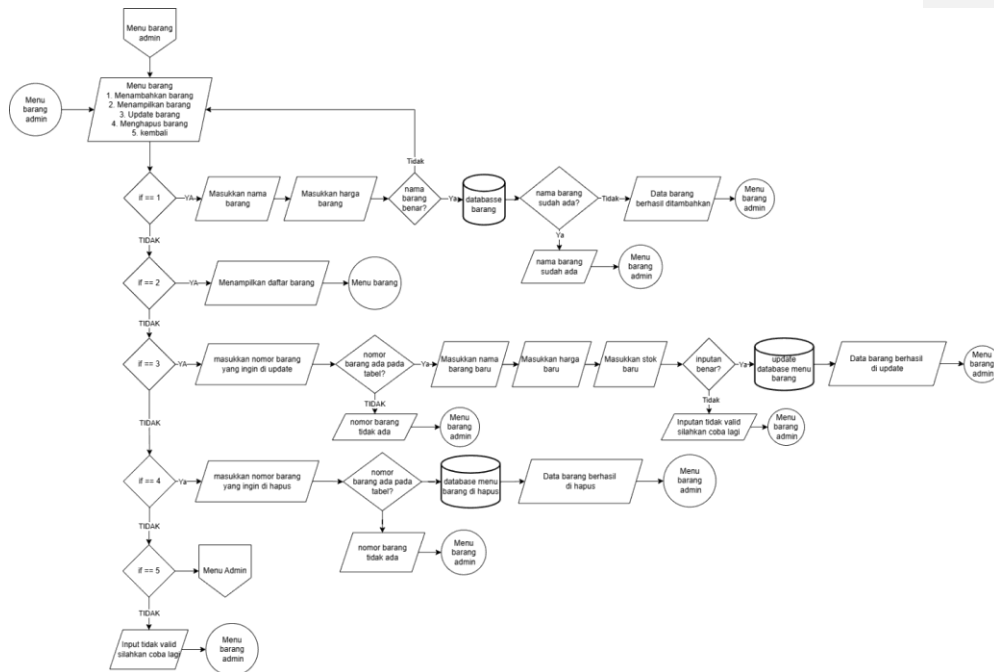
Gambar 2. 7 Flowchart Cek Keranjang (User)



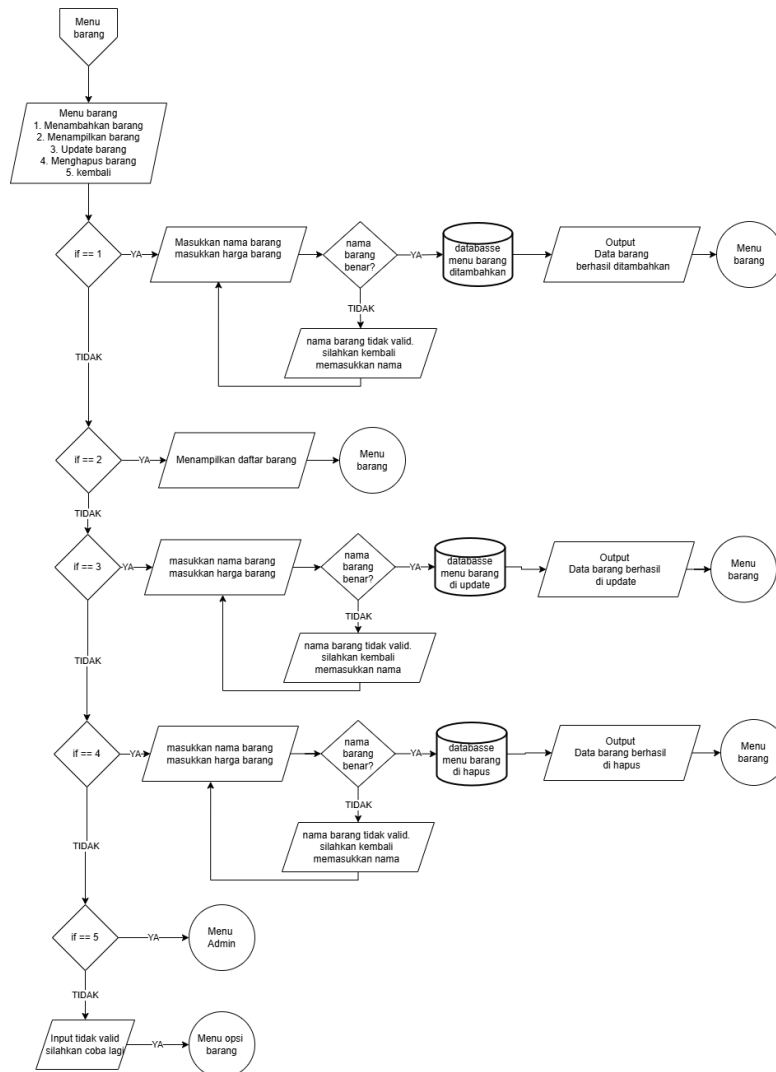
Gambar 2. 8 Flowchart Menu Admin



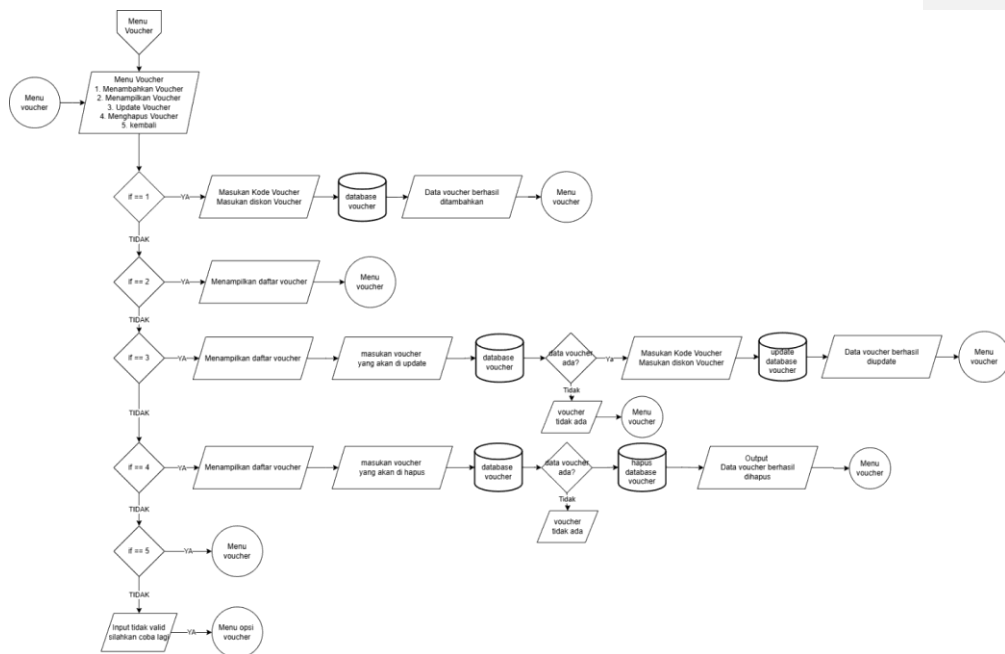
Gambar 2. 9 Flowchart Menu Admin



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

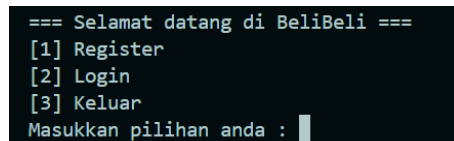
3.1 Implementasi Program

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

1. Csv
2. Isdigit
3. Pwinput
4. Pretty table
5. If, Elif Else
6. Def
7. Loop while True
8. Try except
9. Date time
10. Sorting, Searching, dan CRUD

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

2. Order barang

3. Cek voucher

4. Cek keranjang

5. Cek saldo

6. Top up saldo

7. Kembali

- 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 3.7 Login sebagai Admin

```

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

```

Gambar 3.8 Menu Layanan Admin

1. Menu makanan

```

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

```

Gambar 3.9 Menu Makanan

- Ketika admin memilih opsi 1 : Menu makanan

Admin akan diarahkan menuju layanan yang mengatur menu makanan pada tampilan user. Admin akan diminta untuk memilih opsi.

2. Menu barang

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

- 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.
```

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 |
+-----+-----+-----+
```

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.

```

Untuk mengupdate barang, admin akan diminta untuk menginput nama, harga, stok barang yang ingin di update, sehingga akan memperbaharui 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.

```

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 : 

```

- 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.
```

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 |
+-----+
```

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 | VOUCHER5 | 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.
```

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	VOUCHER5	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.
```

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
```

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 = 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

        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("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)

# 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

```

```

#=====
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 makanan
        lihat_makanan()

        saldo = user["saldo"]

        data_makanan = load_data_makanan() # Memuat data makanan 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_makanan = merge_sort(data_makanan, "harga", reverse)

```

Fitur

```

        # Menampilkan hasil pengurutan
        print("Hasil pengurutan 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)

        # 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_makanan, keyword)

            # Menampilkan hasil pencarian
            if data_makanan:
                print("Hasil pencarian 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)
            else:

```



```

print("Tidak ada makanan 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 makanan 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("Makanan 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

        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}")

    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
            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}")

        elif action == '2':
            # Menambahkan barang 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 barang lain
        continue_order = input("Apakah Anda ingin memesan barang lain?
(ya/tidak): ").strip().lower()

        if continue_order != 'ya':
            print("Terima kasih telah berbelanja!")
            return # Keluar dari fungsi jika pengguna tidak ingin memesan lagi

    except ValueError:
        print("Input tidak valid.")

# 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):
    while True:
        cek_saldo = data_user()
        for user in cek_saldo:
            try:
                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"] = str(int(user["saldo"]) + int(dana))
                            simpan_data_user_ke_csv(cek_saldo)
                            print(f"Saldo anda telah ditambahkan. Saldo sekarang = Rp
{user['saldo']}")
                        else:
                            print("Masukkan nominal yang valid.")
            except ValueError:
                print("Input harus berupa angka. Silakan coba lagi.")
                top_up(username)
            except EOFError:
                print("\nInput tidak valid. Harus berupa angka. Silakan coba lagi.")

```

```

        top_up(username)
    except KeyboardInterrupt:
        print("Input tidak valid. Silakan coba lagi.")
        return top_up(username)
    break

#===== 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 harga_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 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()

# 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("Ops! 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_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.")
    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()

#=====
Menu                                Voucher                                Admin
=====

    # 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

# Fugsi 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

4.2 Saran

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