

**LAPORAN PROYEK AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMROGRAMAN DASAR**



MANAJEMEN MENU DAN PEMESANAN KAFE

Oleh:

Kelompok 3

WAHYU ADITYA	2409106067
ANDI ANUGRAH ALIF S.	2409106058
INAYAH RAMADHANI	2409106068

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
SAMARINDA 2024**

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga kami dapat menyelesaikan proyek akhir praktikum dengan judul "Manajemen Menu dan Pemesanan Kafe" ini. Proyek akhir ini merupakan salah satu syarat untuk menyelesaikan praktikum mata kuliah Algoritma dan Pemrograman Dasar pada semester pertama Program Studi Informatika, Fakultas Teknik, Universitas Mulawarman.

Laporan ini berisi penjelasan tentang program yang kami buat, yaitu sebuah sistem yang bertujuan mempermudah proses manajemen kafe. Program ini dilengkapi dengan fitur untuk admin dalam mengelola menu dan pesanan, serta fitur untuk pengunjung dalam melakukan pemesanan dengan cara yang praktis dan efisien. Dengan menggunakan sistem ini, kami berharap dapat memberikan solusi sederhana untuk kebutuhan operasional kafe yang lebih baik.

Selama proses pembuatan program dan laporan, kami menghadapi beberapa kendala, seperti pemahaman logika pemrograman yang masih berkembang, serta kesulitan teknis dalam merancang sistem yang sesuai dengan kebutuhan. Namun, berkat bimbingan yang luar biasa dari Bang Ifnu Umar dan Bang Adi Muhammad Syifai selaku asisten laboratorium algoritma pemrograman dasar di kelas praktikum B1, kami berhasil menyelesaikan program ini. Kami juga mengucapkan terima kasih yang mendalam kepada Bang Adi Muhammad Syifai yang juga selaku asisten laboratorium pendamping kelompok kami atas bimbingannya selama proses pengerjaan proyek ini, serta kepada teman-teman yang telah membantu dan mendukung kami.

Kami menyadari bahwa laporan ini memiliki banyak kekurangan dan keterbatasan yang perlu diperbaiki. Maka dari itu kritik dan saran dari pembaca sangat diharapkan guna perbaikan di masa yang akan datang.

Akhir kata, semoga makalah ini dapat memberikan manfaat bagi pembaca serta menambah wawasan bagi kita semua.

Samarinda, 17 November 2024

Kelompok 3

TAKARIR

Daftar padanan kata bahasa asing dalam bahasa Indonesia yang digunakan adalah sebagai berikut:

<i>Admin</i>	Admin
<i>Barista</i>	Barista
<i>Data</i>	Data
<i>Detail</i>	Detail
<i>Digital</i>	Digital
<i>Flowchart</i>	Diagram alur
<i>Input</i>	Masukan
<i>Output</i>	Keluaran
<i>Login</i>	Masuk
<i>Logout</i>	Keluar
<i>Online</i>	Daring
<i>Password</i>	Kata sandi
<i>Register</i>	Daftar
<i>Registrasi</i>	Pendaftaran
<i>User</i>	Pengguna
<i>Username</i>	Nama pengguna
<i>Valid</i>	Sah
<i>Visual</i>	Visual

DAFTAR ISI

KATA PENGANTAR	ii
TAKARIR.....	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Kebutuhan Fungsional	2
1.3. Rumusan Masalah	2
1.4. Batasan Masalah.....	2
1.5. Tujuan.....	3
BAB II PERANCANGAN	4
2.1. Analisis Program.....	4
2.2. Flowchart	5
2.2.1. Flowchart Menu Utama.....	6
2.2.2. Flowchart Menu Admin	8
2.2.3. Flowchart Menu Pengunjung	9
2.3. Konsep/Materi Praktikum yang dipakai	10
BAB III HASIL DAN PEMBAHASAN	12
3.1. Tampilan Program.....	12
1.1.1. Tampilan Menu Utama.....	12
1.1.2. Tampilan Menu Admin	13
1.1.3. Tampilan Menu Pengunjung	17
3.2. Source Code.....	19

BAB IV PENUTUP	32
4.1 Kesimpulan.....	32
4.2 Saran.....	32
DAFTAR PUSTAKA	33
LAMPIRAN	34

DAFTAR GAMBAR

Gambar 2. 1	Flowchart Program.....	5
Gambar 2. 2	Flowchart menu login	5
Gambar 2. 3	Flowchart menu utama	6
Gambar 2. 4	Flowchart login	7
Gambar 2. 5	Flowchart menu admin	8
Gambar 2. 6	Flowchart menu pengunjung	9
Gambar 3. 1	Tampilan Awal Menu Utama	12
Gambar 3. 2	Pilihan role sebelum registrasi	12
Gambar 3. 3	Berhasil registrasi sebagai admin.....	12
Gambar 3. 4	Berhasil registrasi sebagai pengunjung	13
Gambar 3. 5	Berhasil login	13
Gambar 3. 6	Tampilan Menu Admin	13
Gambar 3. 7	Daftar menu kafe.....	14
Gambar 3. 8	Melihat pesanan pelanggan	14
Gambar 3. 9	Menyelesaikan pesanan pelanggan	15
Gambar 3. 10	Riwayat pesanan yang pernah dipesan oleh pelanggan.....	15
Gambar 3. 11	Menambahkan menu baru	15
Gambar 3. 12	Memperbarui menu	16
Gambar 3. 13	Menghapus menu	16
Gambar 3. 14	Tampilan Menu Pengunjung	17
Gambar 3. 15	Membuat pesanan	17
Gambar 3. 16	Melihat pesanan milik sendiri	18
Gambar 3. 17	Mengubah pesanan milik sendiri	18
Gambar 3. 18	Riwayat pesanan milik sendiri yang pernah dipesan	18
Gambar 3. 19	Setup Awal	19
Gambar 3. 20	Struktur data awal	19
Gambar 3. 21	Fungsi registrasi admin.....	20
Gambar 3. 22	Username unik	20
Gambar 3. 23	Fungsi registrasi user	21

Gambar 3. 24 Fungsi lihat daftar menu	21
Gambar 3. 25 Fungsi lihat pesanan	22
Gambar 3. 26 Fungsi pesanan selesai	23
Gambar 3. 27 Fungsi hapus menu	24
Gambar 3. 28 Fungsi tambah menu	24
Gambar 3. 29 Fungsi update menu	25
Gambar 3. 30 Fungsi menu admin.....	26
Gambar 3. 31 Fungsi pesan.....	27
Gambar 3. 32 Fungsi history admin	27
Gambar 3. 33 Fungsi history user	28
Gambar 3. 34 Fungsi login.....	29
Gambar 3. 35 Fungsi lihat pesanan	30
Gambar 3. 36 Fungsi menu user	30
Gambar 3. 37 Fungsi menu utama	31
Gambar 4. 1 Kartu konsul 1	34
Gambar 4. 2 Kartu konsul 2	34
Gambar 4. 3 Kartu konsul 3	35

BAB I

PENDAHULUAN

1.1. Latar Belakang

Samarinda mengalami peningkatan jumlah kafe dalam beberapa tahun terakhir, yang mencerminkan perubahan gaya hidup masyarakat yang gemar menghabiskan waktu di kafe. Namun, banyak kafe masih menggunakan sistem manual untuk pemesanan dan manajemen, yang seringkali menyebabkan kesalahan dalam pemesanan, waktu tunggu yang lama, dan ketidakpuasan pelanggan. Kesalahan pemesanan bisa terjadi karena komunikasi yang tidak jelas antara pelanggan dan barista, sementara manajemen stok yang tidak efektif dapat menyebabkan kekurangan bahan baku atau pemborosan. Selain itu, tanpa data yang cukup, sulit bagi pemilik kafe untuk menganalisis penjualan dan memahami kebutuhan pelanggan.

Untuk mengatasi masalah-masalah ini, terciptalah ide membuat program pemesanan menu dan manajemen kafe yang terintegrasi. Program ini diharapkan dapat mengurangi kesalahan pemesanan dengan sistem digital, mempercepat proses pemesanan, dan mempermudah manajemen stok bahan baku. Selain itu, program ini akan membantu mengumpulkan data transaksi untuk analisis lebih lanjut, sehingga pemilik kafe dapat membuat keputusan bisnis yang lebih baik dan strategi pemasaran yang efektif. Dengan demikian, program ini tidak hanya meningkatkan efisiensi operasional kafe tetapi juga meningkatkan kepuasan pelanggan serta memberikan pemahaman yang lebih baik bagi mereka yang ingin membuka kafe di masa depan.

1.2 Kebutuhan Fungsional

Agar program dapat memenuhi kebutuhan pengguna dan mencapai tujuan yang telah ditetapkan, maka program harus memiliki fitur-fitur sebagai berikut:

1. Program harus memiliki fitur registrasi untuk admin dan pengunjung.
2. Program menyediakan menu *login* berdasarkan peran pengguna.
3. Admin dapat menambah, mengubah, melihat, atau menghapus menu serta pesanan.
4. Pengunjung dapat melihat menu, membuat pesanan, melihat pesanan, dan mengubah pesanan.
5. Sistem mencatat total pembayaran dari setiap pesanan.
6. Sistem memberikan pesan validasi jika ada input yang salah.

1.3 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa permasalahan yang ingin diselesaikan melalui program ini, yaitu:

1. Bagaimana cara mempermudah proses pemesanan di kafe agar lebih cepat dan akurat?
2. Bagaimana membuat aplikasi yang sederhana namun efektif untuk digunakan oleh admin dan pengunjung?
3. Bagaimana program dapat mencatat data pesanan dan menghitung total dengan akurat?

1.4 Batasan Masalah

Untuk memastikan program tetap terarah, terdapat beberapa batasan dalam pengembangan program ini, yaitu:

1. Program hanya dirancang untuk penggunaan dasar kafe tanpa fitur integrasi pembayaran *online*.
2. Program tidak mencakup manajemen stok menu secara detail.

3. Pengguna hanya bisa melakukan login dan registrasi dengan *username* dan *password* sederhana tanpa fitur keamanan tambahan.
4. Analisis penjualan yang dihasilkan hanya berupa data pesanan tanpa grafik atau laporan visual.

1.5 Tujuan

Program ini dikembangkan dengan tujuan sebagai berikut:

1. Mempermudah admin untuk mengatur menu dan melihat pesanan dengan lebih efisien.
2. Memberikan pengalaman yang lebih praktis bagi pengunjung kafe.
3. Mengurangi kesalahan dalam proses pemesanan melalui sistem yang terorganisir.

BAB II

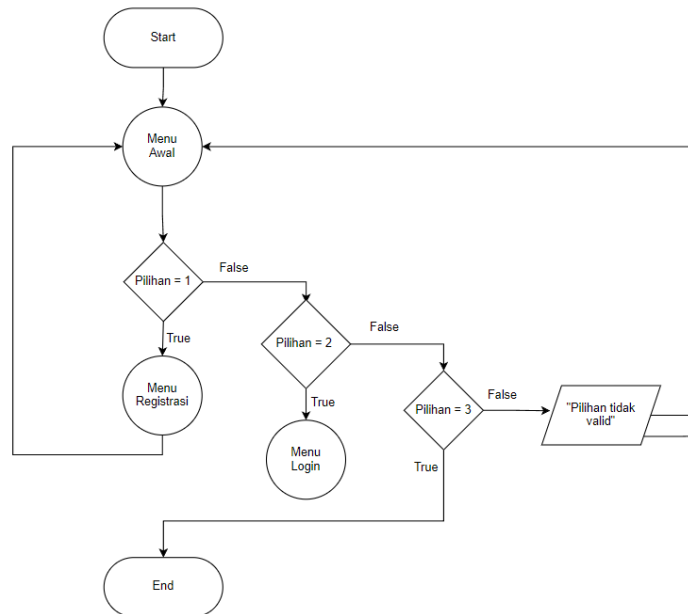
PERANCANGAN

2.1 Analisis Program

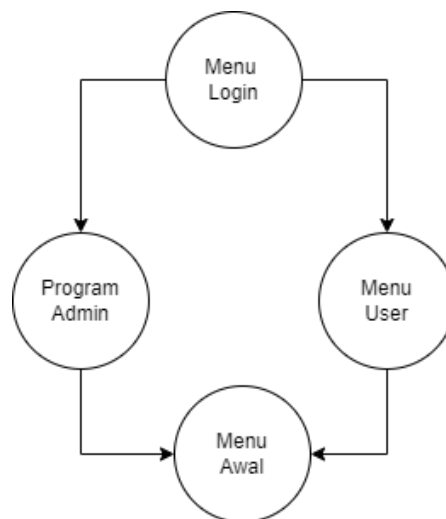
Program kami menjelaskan alur sistem manajemen kafe yang mencakup registrasi, *login*, dan akses menu berdasarkan peran pengguna (admin atau pengunjung). Di menu utama, pengguna dapat memilih *register*, *login*, dan *logout*. Registrasi sebagai Admin dan Pengunjung memerlukan *username*, *password*. Setelah *login*, sistem memverifikasi data dan mengarahkan pengguna ke Menu Admin atau Menu Pengunjung.

Dalam Menu Admin, terdapat opsi untuk melihat, menambah, memperbarui, atau menghapus menu dan pesanan. Admin yang memilih opsi Keluar akan kembali ke Menu Utama, sedangkan input tidak *valid* memunculkan pesan "Pilihan tidak valid". Di Menu Pengunjung, terdapat opsi untuk melihat menu, menambah, mengubah, atau melihat orderan, serta *logout*. Sistem memastikan setiap proses sederhana dan efisien, mendukung operasional kafe yang terorganisir dengan baik.

2.2 Flowchart

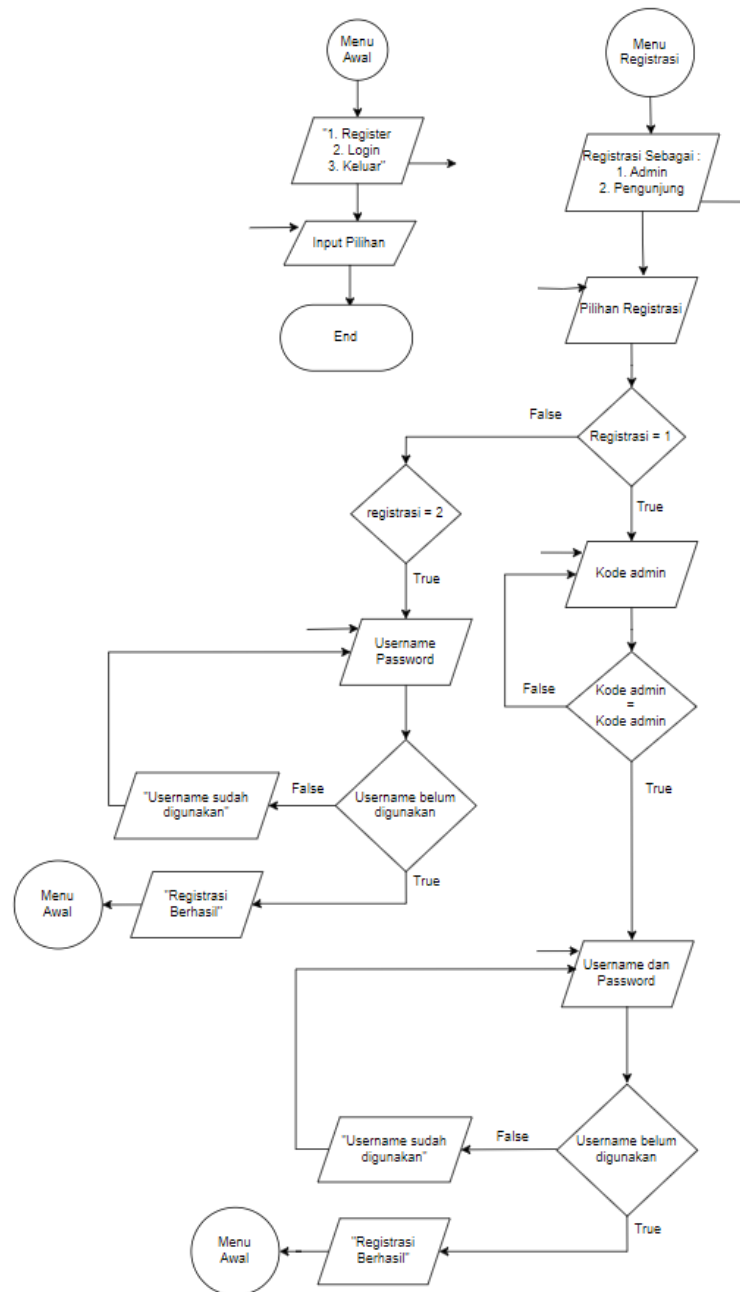


Gambar 2. 1 Flowchart Program

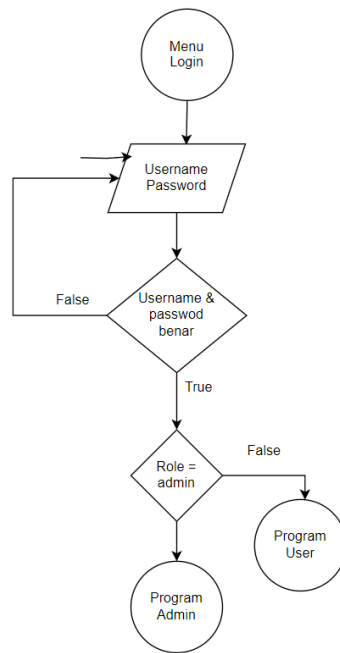


Gambar 2. 2 Flowchart menu login

2.2.1. Flowchart Menu Utama



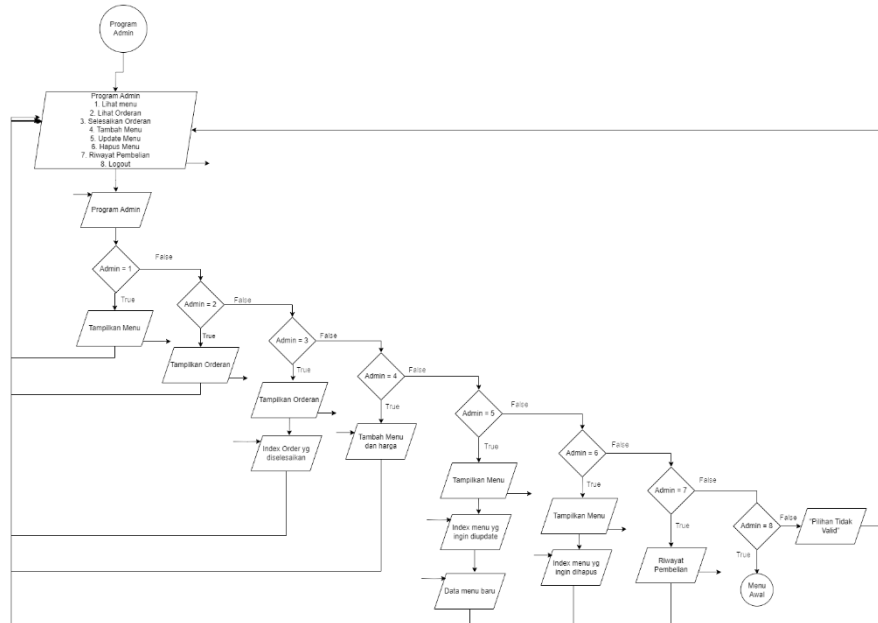
Gambar 2. 3 Flowchart menu utama



Gambar 2. 4 Flowchart login

Flowchart ini menunjukkan alur proses pemesanan dan manajemen kafe. Dari menu utama, pengguna bisa memilih *register*, *login*, atau keluar. Jika memilih *register*, pengguna dapat mendaftar sebagai Admin (dengan memasukkan *username*, *password*, dan kode admin) atau sebagai Pengunjung (cukup dengan *username* dan *password*). Setelah registrasi berhasil, pengguna kembali ke menu utama. Jika memilih *login*, pengguna memasukkan *username* dan *password*, lalu diarahkan ke menu admin untuk mengelola kafe (inventaris pemesanan menu) atau menu pengunjung untuk melihat menu, memesan, mengubah, atau membatalkan pesanan. Pilihan keluar atau input yang salah akan mengembalikan pengguna ke menu utama. Alur ini memastikan proses berjalan lancar dan akses dibedakan sesuai peran pengguna.

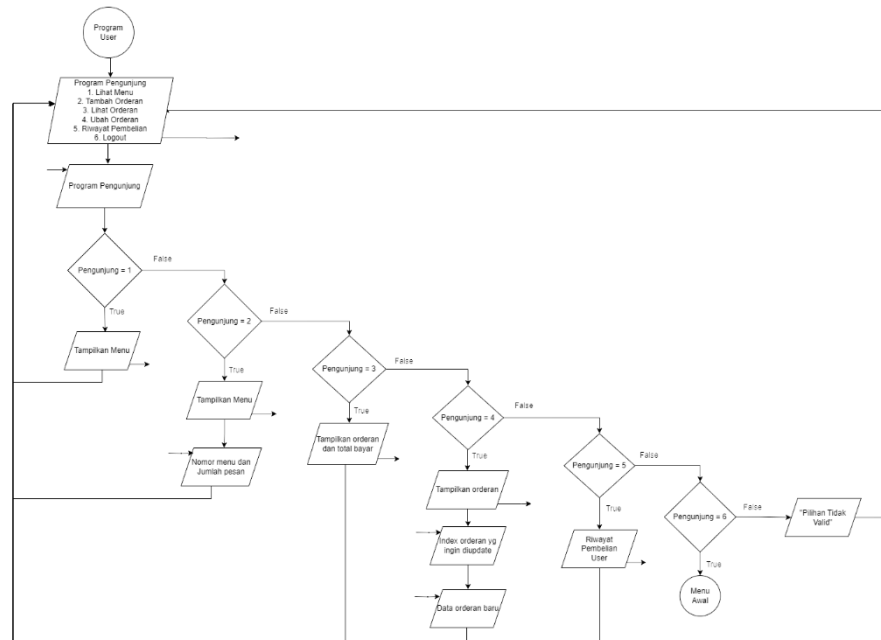
2.2.2. Flowchart Menu Admin



Gambar 2. 5 Flowchart menu admin

Flowchart ini menjelaskan alur sistem manajemen kafe untuk admin. Dari menu utama, admin bisa memilih melihat menu, melihat pesanan, menambah atau memperbarui menu, menghapus menu atau pesanan, atau keluar dari program. Setiap pilihan memiliki proses mudah, seperti memasukkan nama dan harga untuk menambah menu, memilih indeks untuk menghapus atau memperbarui data, dan melihat daftar sebelum melakukan perubahan. Jika admin memilih keluar, sistem kembali ke menu utama, sedangkan input tidak valid akan memunculkan pesan kesalahan.

2.2.3. Flowchart Menu Pengunjung



Gambar 2. 6 Flowchart menu pengunjung

Flowchart ini menjelaskan proses bagi pengunjung saat menggunakan aplikasi kafe. Dari Menu Pengunjung, tersedia lima pilihan: Lihat Menu, Tambah Orderan, Lihat Orderan, Ubah Orderan, dan *Logout*. Pilihan Lihat Menu menampilkan daftar makanan dan minuman yang tersedia. Jika memilih Tambah Orderan, pengunjung melihat menu, lalu memasukkan nomor menu dan jumlah pesanan. Pada Lihat Orderan, sistem menunjukkan daftar pesanan beserta total biaya. Untuk Ubah Orderan, pengunjung memilih orderan yang ingin diubah dan memasukkan data baru. Jika memilih *Logout* atau memberikan input yang salah, sistem akan kembali ke Menu Utama atau menampilkan pesan "Pilihan Tidak Valid".

2.3 Konsep/Materi Praktikum yang dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan persyaratan yang terdapat pada modul di antaranya:

1. Fungsi Dasar (fungsi print)

Fungsi print pada program merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga.

2. Fungsi Percabangan

Fungsi perulangan dalam program kami digunakan untuk mengelola input pengguna, menampilkan menu, dan memproses pesanan. Contohnya perulangan while.

3. Fungsi Perulangan

Fungsi perulangan dalam program kami digunakan untuk menentukan tindakan yang diambil berdasarkan input pengguna. Contohnya percabangan if-elif-else.

4. Fungsi List

Fungsi list dalam program ini, list digunakan untuk menyimpan daftar pengguna, menu, pesanan, dan riwayat pesanan. Contohnya list menu.

5. Error Handling

Error handling dalam program ini, error handling digunakan untuk memastikan program tetap berjalan meskipun ada kesalahan input dari pengguna. Contoh error handling menggunakan try-except.

6. Fungsi (def)

Fungsi dalam program kami digunakan untuk berbagai operasi seperti registrasi pengguna, menambah pesanan, dan mengelola menu.

7. Dictionary

Dictionary dalam program ini digunakan untuk menyimpan pesanan dan riwayat pesanan berdasarkan nama pengguna.

8. Fungsi Input

Fungsi input digunakan untuk menerima input dari pengguna. Program kami menggunakan input untuk mendapatkan pilihan menu dari pengguna, serta detail pesanan.

BAB III

HASIL DAN PEMBAHASAN

3.1. Tampilan Program

3.1.1. Tampilan Menu Utama

```
=====
          KAFE RUANG RASA
    Silahkan login terlebih dahulu
    Register jika belum punya akun
=====
[1]. Register
[2]. Login
[3]. Keluar
-----
Pilih menu:
```

Gambar 3. 1 Tampilan Awal Menu Utama

```
=====
          PILIH ROLE
=====
[1]. Registrasi sebagai admin
[2]. Registrasi sebagai pengunjung
-----
Masukkan Pilihan : █
```

Gambar 3. 2 Pilihan role sebelum registrasi

```
Buat Username: supra
Masukkan Password: supra123
Masukkan Nama Lengkap: Supra
Pengguna supra berhasil registrasi sebagai Admin.
Klik Enter untuk melanjutkan█
```

Gambar 3. 3 Berhasil registrasi sebagai admin

```
Buat Username: supri
Masukkan Password: supri123
Masukkan Nama Lengkap: Supri
Pengguna supri berhasil registrasi.
Klik Enter untuk melanjutkan
```

Gambar 3. 4 Berhasil registrasi sebagai pengunjung

```
=====
                SILAHKAN LOGIN
=====
Username: supra
Password: supra123
Selamat anda berhasil login
Klik enter untuk masuk ke menu
```

Gambar 3. 5 Berhasil login

3.1.2. Tampilan Menu Admin

```
=====
                MENU ADMIN
=====
[1]. Lihat Menu
[2]. Lihat Pesanan
[3]. Selesaikan Pesanan
[4]. Tambah Menu
[5]. Update Menu
[6]. Hapus Menu
[7]. Riwayat Pembelian
[8]. Logout
-----
Masukkan Pilihan : 
```

Gambar 3. 6 Tampilan Menu Admin

=====		
MENU KAFE RUANG RASA		
=====		
No	Menu	Harga
1	Americano	18000
2	Espresso	15000
3	Coffe Latte	22000
4	Latte Macchiato	23000
5	Caramel Macchiato	25000
6	Cafe Frappe	28000
7	Cappucino	24000
8	Frozen Mocha	27000
9	Smoothies	30000
10	Choco Mint	29000
11	Ice Cream	18000
12	Apple Pie	20000
13	Waffle	22000
14	Cinnamon Roll	25000
15	Cheese Cake	28000
Klik Enter Untuk Melanjutkan		

Gambar 3. 7 Daftar menu kafe

=====		
PESANAN PELANGGAN		
=====		
1. Pesanan oleh Supri:		
Americano - Jumlah :	3	
Total Harga :	Rp. 54000	

Klik Enter Untuk Melanjutkan		

Gambar 3. 8 Melihat pesanan pelanggan

```

=====
                        SELESAIKAN PESANAN
=====

1. Pesanan oleh Supri:
   Americano - Jumlah : 3
   Total Harga : Rp. 54000
-----
Pilih Pesanan untuk diselesaikan (0 untuk kembali): 1
Pesanan berhasil diselesaikan dan ditambahkan ke riwayat.
Klik Enter Untuk Melanjutkan

```

Gambar 3. 9 Menyelesaikan pesanan pelanggan

```

=====
                        RIWAYAT PESANAN PELANGGAN
=====

Pesanan oleh Supri (Username: supri):
1. Americano - Jumlah: 3 - Harga per pcs: 18000
   Total Harga: Rp. 54000
-----
Klik Enter Untuk Melanjutkan

```

Gambar 3. 10 Riwayat pesanan yang pernah dipesan oleh pelanggan

```

=====
                        TAMBAH MENU BARU
=====

Masukkan nama menu: Brownies
Masukkan harga: 26000
Menu Brownies berhasil ditambahkan.
Klik Enter Untuk Melanjutkan

```

Gambar 3. 11 Menambahkan menu baru

```

=====
PERBARUI MENU
=====

```

No	Menu	Harga
1	Americano	18000
2	Espresso	15000
3	Coffe Latte	22000
4	Latte Macchiato	23000
5	Caramel Macchiato	25000
6	Cafe Frappe	28000
7	Cappucino	24000
8	Frozen Mocha	27000
9	Smoothies	30000
10	Choco Mint	29000
11	Ice Cream	18000
12	Apple Pie	20000
13	Waffle	22000
14	Cinnamon Roll	25000
15	Cheese Cake	28000
16	Brownies	26000

```

Masukkan nomor menu yang ingin diupdate (0 Untuk Kembali): 16
Masukkan nama menu: Lemon Cake
Masukkan harga: 26000
Menu berhasil diupdate
Klik Enter Untuk Melanjutkan

```

Gambar 3. 12 Memperbarui menu

```

=====
HAPUS MENU
=====

```

No	Menu	Harga
1	Americano	18000
2	Espresso	15000
3	Coffe Latte	22000
4	Latte Macchiato	23000
5	Caramel Macchiato	25000
6	Cafe Frappe	28000
7	Cappucino	24000
8	Frozen Mocha	27000
9	Smoothies	30000
10	Choco Mint	29000
11	Ice Cream	18000
12	Apple Pie	20000
13	Waffle	22000
14	Cinnamon Roll	25000
15	Cheese Cake	28000
16	Lemon Cake	26000

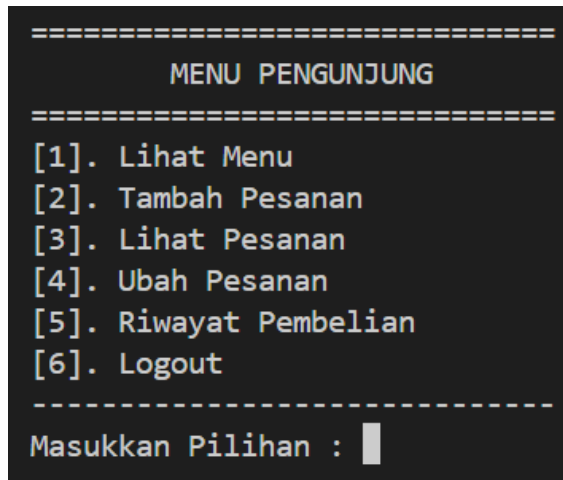
```

Masukkan nomor data yang ingin dihapus (0 Untuk Kembali): 16
Menu Lemon Cake berhasil dihapus
Klik Enter Untuk Melanjutkan

```

Gambar 3. 13 Menghapus menu

3.1.3. Tampilan Menu Pengunjung



Gambar 3. 14 Tampilan Menu Pengunjung



Gambar 3. 15 Membuat pesanan

```

=====
                        LIHAT PESANAN SAYA
=====
Pesanan Anda :
1. Americano - Jumlah: 3
Total Harga Pesanan Anda : Rp. 54000
-----
Klik Enter Untuk Melanjutkan

```

Gambar 3. 16 Melihat pesanan milik sendiri

```

=====
                        UBAH PESANAN
=====
Pesanan Anda :
1. Americano - Jumlah: 3
Total Harga Pesanan Anda : Rp. 54000
-----
Masukkan Nomor Menu Yang Ingin Diganti (0 Untuk Kembali): 1
[1]. Ubah Jumlah Pesanan
[2]. Hapus Pesanan
Masukkan Pilihan: 

```

Gambar 3. 17 Mengubah pesanan milik sendiri

```

=====
                        RIWAYAT PESANAN SAYA
=====
History Anda :
1. Americano - Jumlah: 3
Total Harga Pesanan Anda : Rp. 54000
-----
Klik Enter Untuk Melanjutkan

```

Gambar 3. 18 Riwayat pesanan milik sendiri yang pernah dipesan

3.2.Source Code

- `import os` : Mengimpor modul `os` yang menyediakan fungsi untuk berinteraksi dengan sistem operasi
- `os.system("cls")` : Menjalankan perintah `cls` untuk membersihkan layar konsol (berfungsi di Windows).

```
Setup Awal

import os
from tabulate import tabulate

def clear():
    os.system("cls")
```

Gambar 3. 19 Setup Awal

```
Struktur Data Awal

users = [["admin", "admin", "admin",
"admin"]]

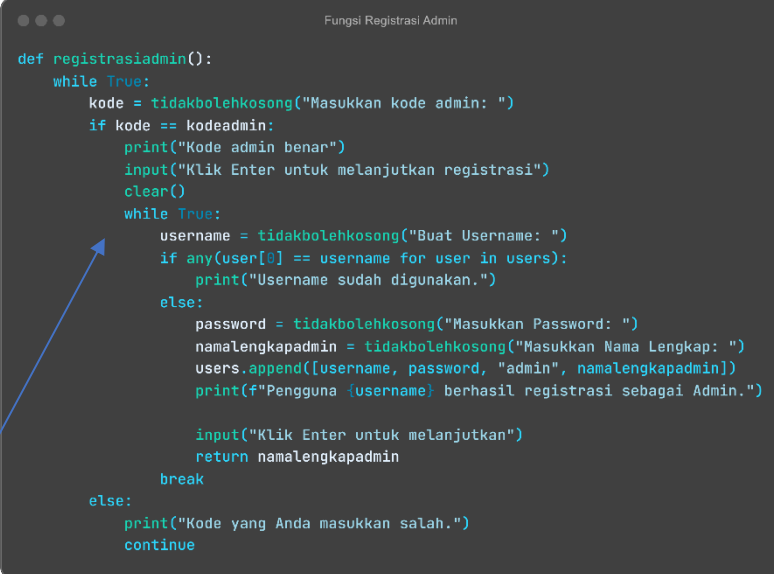
menu = [
["Americano", 18000],
["Espresso", 15000],
["Coffe Latte", 22000],
["Latte Macchiato", 23000],
["Caramel Macchiato", 25000],
["Cafe Frappe", 28000],
["Cappucino", 24000],
["Frozen Mocha", 27000],
["Smoothies", 30000],
["Choco Mint", 29000],
["Ice Cream", 18000],
["Apple Pie", 20000],
["Waffle", 22000],
["Cinnamon Roll", 25000],
["Cheese Cake", 28000]]

pesanan = {}
history = {}
```

Menginisialisasi daftar users dengan satu pengguna yang merupakan admin.

Daftar menu yang berisi item menu dan harganya.

Gambar 3. 20 Struktur data awal



```

def registrasiadmin():
    while True:
        kode = tidakbolehkosong("Masukkan kode admin: ")
        if kode == kodeadmin:
            print("Kode admin benar")
            input("Klik Enter untuk melanjutkan registrasi")
            clear()
            while True:
                username = tidakbolehkosong("Buat Username: ")
                if any(user[0] == username for user in users):
                    print("Username sudah digunakan.")
                else:
                    password = tidakbolehkosong("Masukkan Password: ")
                    namalengkapadmin = tidakbolehkosong("Masukkan Nama Lengkap: ")
                    users.append([username, password, "admin", namalengkapadmin])
                    print(f"Pengguna {username} berhasil registrasi sebagai Admin.")

                    input("Klik Enter untuk melanjutkan")
                    return namalengkapadmin
                break
            else:
                print("Kode yang Anda masukkan salah.")
                continue

```

Gambar 3. 21 Fungsi registrasi admin

- Mendefinisikan `registrasiadmin()`, sebuah fungsi untuk registrasi admin.
- Meminta kode admin dan memeriksa kebenarannya.



```

if any(user[0] == username for user in users):
    print("Username sudah digunakan")

```

Gambar 3. 22 Username unik

- **users** adalah sebuah daftar (list) yang diasumsikan sudah ada sebelumnya, berisi informasi pengguna dalam format: `[username, password, role, namalengkap]`.
- **any(...)**: Mengecek apakah ada elemen dalam `users` di mana `user[0]` (username) cocok dengan input `username`.

```

Fungsi Registrasi User

def registrasiuser():
    while True:
        username = tidakbolehkong("Buat Username: ")
        if any(user[0] == username for user in users):
            print("Username sudah digunakan, silakan ganti.")
        else:
            password = tidakbolehkong("Masukkan Password: ")
            namaengkapuser = tidakbolehkong("Masukkan Nama Lengkap: ")
            users.append([username, password, "user", namaengkapuser])
            print(f"Pengguna {username} berhasil registrasi.")
            input("Klik Enter untuk melanjutkan")
            return namaengkapuser
    break

```

Gambar 3. 23 Fungsi registrasi user

- Mendefinisikan `registrasiuser()`, sebuah fungsi untuk registrasi pengguna.
- Meminta username, password, dan nama lengkap.

```

Fungsi Lihat Menu

head = ["No", "Menu", "Harga"]

def lihatmenu():
    tabelmenu = [[i + 1, item[0], item[1]] for i, item in enumerate(menu)]
    print(tabulate(tabelmenu, headers=head, tablefmt="fancy_outline"))

```

- Menggunakan `tabulate` untuk membuat tabel terformat dari item menu.
- Mendefinisikan `lihatmenu()`, sebuah fungsi untuk menampilkan

Gambar 3. 24 Fungsi lihat daftar menu

```
Fungsi Lihat Pesanan

def lihatpesanan():
    print("=" * 40)
    print("          PESANAN PELANGGAN")
    print("=" * 40)
    if len(pesanan) == 0:
        print("Tidak Ada Pesanan")
        input("Klik Enter Untuk Melanjutkan")
    else:
        for index, (user_name, user_orders) in enumerate(pesanan.items(), start=1):
            total = sum(order['harga'] * order['jumlah'] for order in user_orders)
            print(f"\n {index}. Pesanan oleh {user_orders[0]['nama']}:")
            for order in user_orders:
                print(f"    {order['pesanan']} - Jumlah : {order['jumlah']}")
                print(f"    Total Harga : Rp. {total}")
            print("-" * 40)
            input("Klik Enter Untuk Melanjutkan")
```

Gambar 3. 25 Fungsi lihat pesanan

- Mendefinisikan `lihatpesanan()`, sebuah fungsi untuk menampilkan pesanan.
- Memeriksa apakah ada pesanan dan mencetaknya bersama dengan total harga.

```

Fungsi Pesanan Selesai

def pesananselesai():
    print("=" * 40)
    print("          SELESAIKAN PESANAN")
    print("=" * 40)
    if len(pesanan) == 0:
        print("Tidak Ada Pesanan")
        input("Klik Enter Untuk Melanjutkan")
    else:
        for index, (user_name, user_orders) in enumerate(pesanan.items(), start=1):
            total = sum(order['harga'] * order['jumlah'] for order in user_orders)
            print(f"\n {index}. Pesanan oleh {user_orders[0]['nama']}:")
            for order in user_orders:
                print(f"    {order['pesanan']} - Jumlah : {order['jumlah']}")
            print(f"    Total Harga : Rp. {total}")
            print("-" * 40)

    while True:
        try:
            selesai = tidakbolehkosong("Pilih Pesanan untuk diselesaikan (0 untuk
kembali): ")
            if selesai.isdigit() and 0 < int(selesai) <= len(pesanan):
                selesai = int(selesai) - 1
                user_name = list(pesanan.keys())[selesai]
                user_orders = pesanan[user_name]

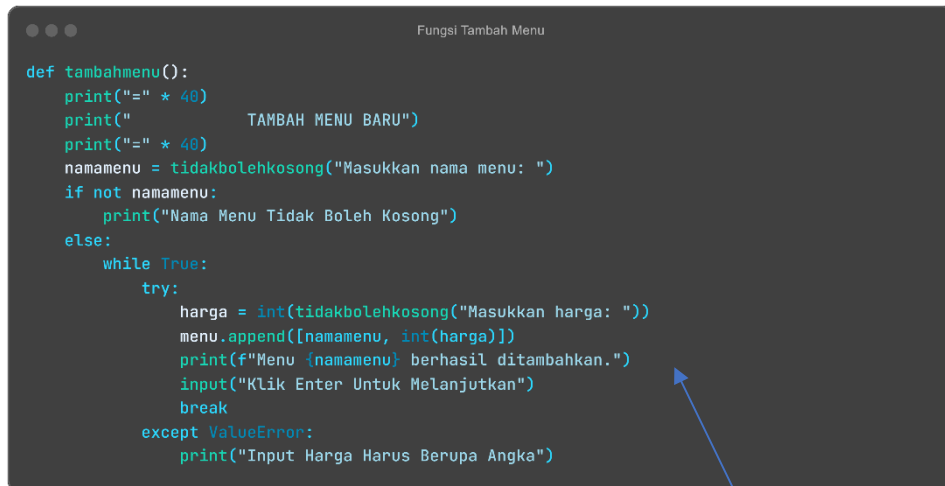
                if user_name not in history:
                    history[user_name] = []
                history[user_name].extend(user_orders)

                del pesanan[user_name]
                print("Pesanan berhasil diselesaikan dan ditambahkan ke riwayat.")
                input("Klik Enter Untuk Melanjutkan")
                break
            elif selesai == "0":
                input("Klik Enter Untuk Melanjutkan")
                break
            else:
                print("Nomor pesanan tidak valid")
        except ValueError:
            print("Input Salah")

```

Gambar 3. 26 Fungsi pesanan selesai

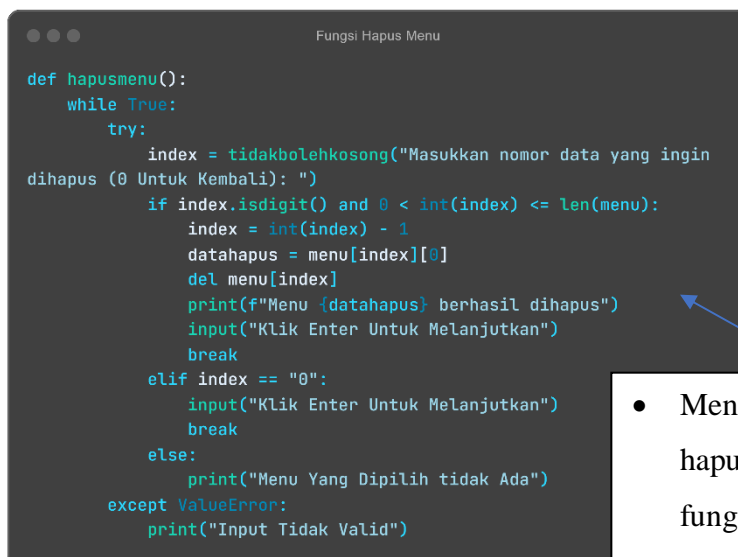
- Mendefinisikan pesananselesai(), sebuah fungsi untuk menyelesaikan pesanan.
- Memungkinkan pemilihan pesanan untuk ditandai sebagai selesai dan memindahkannya ke riwayat.



```
def tambahmenu():
    print("=" * 40)
    print("          TAMBAH MENU BARU")
    print("=" * 40)
    namamenu = tidakbolehkosong("Masukkan nama menu: ")
    if not namamenu:
        print("Nama Menu Tidak Boleh Kosong")
    else:
        while True:
            try:
                harga = int(tidakbolehkosong("Masukkan harga: "))
                menu.append([namamenu, int(harga)])
                print(f"Menu {namamenu} berhasil ditambahkan.")
                input("Klik Enter Untuk Melanjutkan")
                break
            except ValueError:
                print("Input Harga Harus Berupa Angka")
```

Gambar 3. 28 Fungsi tambah menu

- Mendefinisikan tambahmenu(), sebuah fungsi untuk menambahkan item menu baru.
- Meminta nama dan harga menu, kemudian menambahkannya ke dalam daftar menu.



```
def hapusmenu():
    while True:
        try:
            index = tidakbolehkosong("Masukkan nomor data yang ingin dihapus (0 Untuk Kembali): ")
            if index.isdigit() and 0 < int(index) <= len(menu):
                index = int(index) - 1
                datahapus = menu[index][0]
                del menu[index]
                print(f"Menu {datahapus} berhasil dihapus")
                input("Klik Enter Untuk Melanjutkan")
                break
            elif index == "0":
                input("Klik Enter Untuk Melanjutkan")
                break
            else:
                print("Menu Yang Dipilih tidak Ada")
        except ValueError:
            print("Input Tidak Valid")
```

Gambar 3. 27 Fungsi hapus menu

- Mendefinisikan hapusmenu(), sebuah fungsi untuk menghapus item menu.
- Meminta item menu yang ingin dihapus.


```

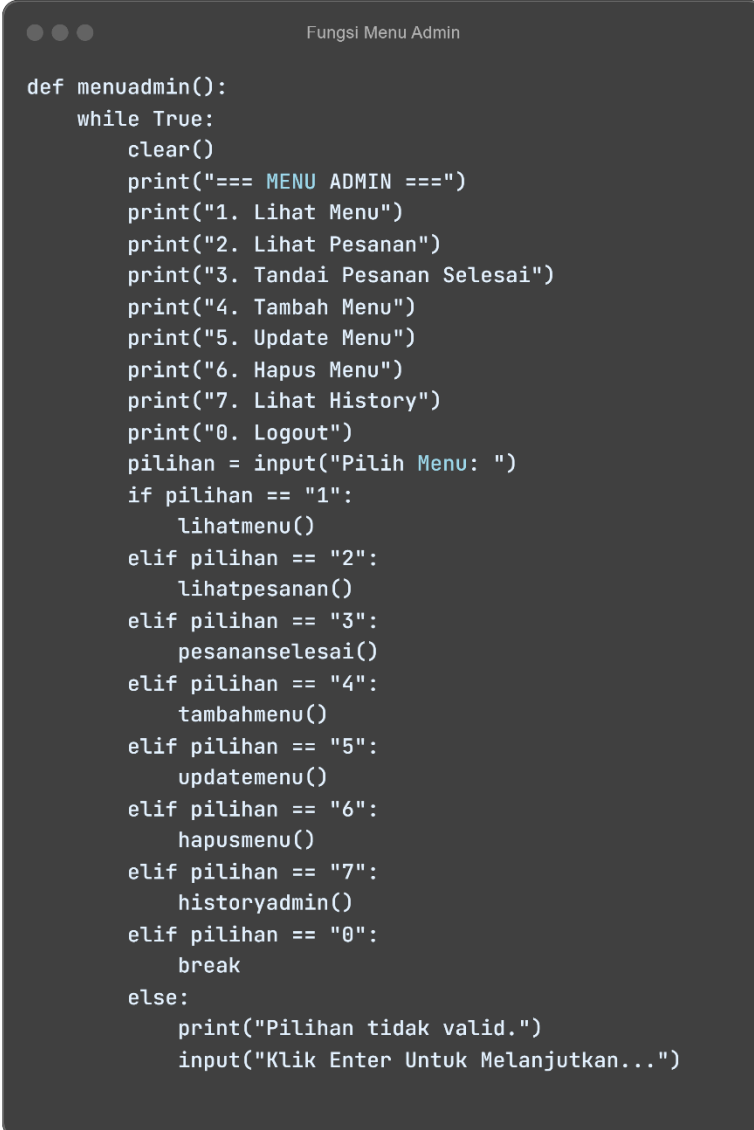
Fungsi Update Menu

def updatemenu():
    while True:
        index = tidakbolehkosong("Masukkan nomor menu yang ingin diupdate (0 Untuk Kembali): ")
        if index == "0":
            break
        elif index.isdigit() and 0 < int(index) <= len(menu):
            index = int(index) - 1
            nama_baru = input("Masukkan nama menu: ")
            while True:
                try:
                    harga_baru = int(input("Masukkan harga: "))
                    if nama_baru:
                        menu[index][0] = nama_baru
                    if (harga_baru):
                        menu[index][1] = int(harga_baru)
                    print("Menu berhasil diupdate")
                    input("Klik Enter Untuk Melanjutkan")
                    clear()
                    return
                except ValueError:
                    print("Input Harga Harus Berupa Angka")
            else:
                print("Nomor menu tidak valid")

```

Gambar 3. 29 Fungsi update menu

- Mendefinisikan `updatemenu()`, sebuah fungsi untuk memperbaiki item menu yang ada.
- Meminta item menu yang ingin diperbarui dan detail baru.



```
def menuadmin():
    while True:
        clear()
        print("=== MENU ADMIN ===")
        print("1. Lihat Menu")
        print("2. Lihat Pesanan")
        print("3. Tandai Pesanan Selesai")
        print("4. Tambah Menu")
        print("5. Update Menu")
        print("6. Hapus Menu")
        print("7. Lihat History")
        print("0. Logout")
        pilihan = input("Pilih Menu: ")
        if pilihan == "1":
            lihatmenu()
        elif pilihan == "2":
            lihatpesanan()
        elif pilihan == "3":
            pesananselesai()
        elif pilihan == "4":
            tambahmenu()
        elif pilihan == "5":
            updatemenu()
        elif pilihan == "6":
            hapusmenu()
        elif pilihan == "7":
            historyadmin()
        elif pilihan == "0":
            break
        else:
            print("Pilihan tidak valid.")
            input("Klik Enter Untuk Melanjutkan...")
```

Gambar 3. 30 Fungsi menu admin

```

Fungsi History Admin

def historyadmin():
    if len(history) == 0:
        print("Belum ada pesanan dari pengguna.")
        input("Klik Enter Untuk Melanjutkan...")
    else:
        print("Riwayat Pembelian User")
        for user_name, user_orders in history.items():
            if user_orders:
                print(f"\nPesanan oleh {user_orders[0]['nama']}: (Username: {user_name}):")
                total = 0
                for order in user_orders:
                    print(f"    {order['pesanan']} - Jumlah: {order['jumlah']} - Harga per unit: {order['harga']}")
                    total += order['harga'] * order['jumlah']
                print(f"    Total Harga: Rp. {total}")
            input("Klik Enter Untuk Melanjutkan...")

```

Gambar 3. 32 Fungsi history admin

- Mendefinisikan historyadmin(), sebuah fungsi untuk melihat riwayat pesanan dari semua pengguna.
- Menampilkan pesanan untuk setiap pengguna beserta total harga.

```

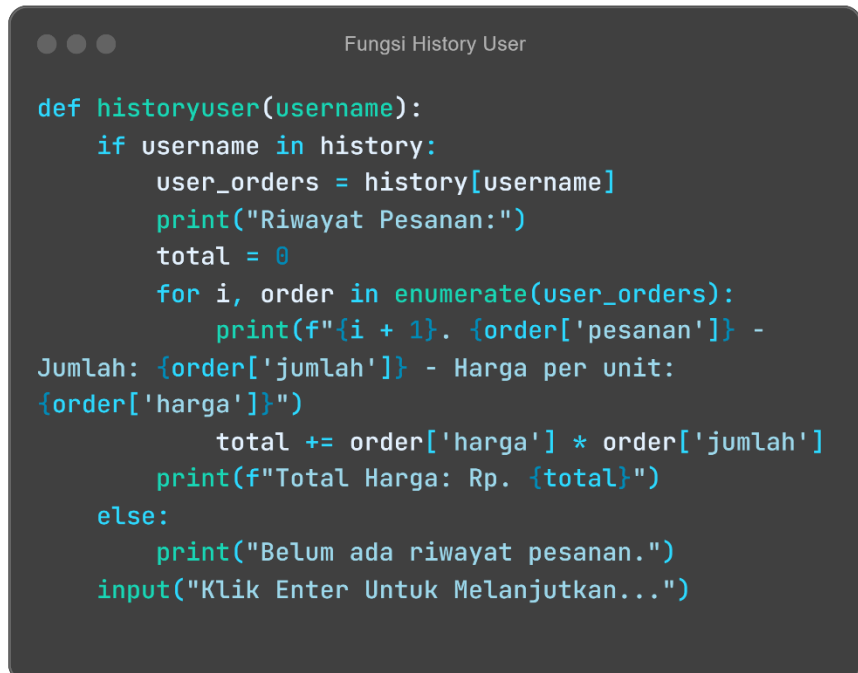
Fungsi Pesan

def pesan(username, namalengkap):
    while True:
        clear()
        print("=== MENU ===")
        lihatmenu()
        pesan = input("Masukkan nomor menu yang ingin dipesan (0 untuk selesai): ")
        if pesan.isdigit() and 0 < int(pesan) <= len(menu):
            pesan = int(pesan) - 1
            menu_pesanan = menu[pesan]
            while True:
                try:
                    jumlah = int(input(f"Masukkan jumlah untuk {menu_pesanan[0]}: "))
                    if username not in pesanan:
                        pesanan[username] = []
                    pesanan[username].append({"nama": namalengkap, "pesanan": menu_pesanan[0], "jumlah": jumlah, "harga": menu_pesanan[1]})
                    print(f"{menu_pesanan[0]} berhasil dipesan.")
                    input("Klik Enter Untuk Melanjutkan...")
                    break
                except ValueError:
                    print("Input jumlah harus berupa angka.")
            elif pesan == "0":
                break
            else:
                print("Nomor menu tidak valid.")

```

Gambar 3. 31 Fungsi pesan

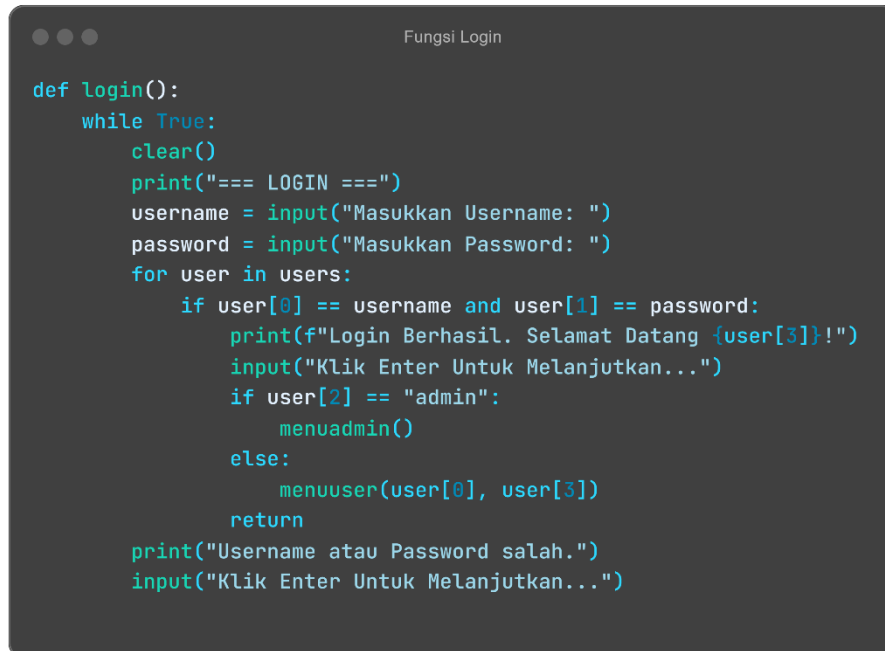
- Mendefinisikan pesan(), sebuah fungsi untuk pengguna memesan item.
- Meminta item menu dan jumlah, kemudian menambahkannya ke dalam kamus pesanan.



```
def historyuser(username):  
    if username in history:  
        user_orders = history[username]  
        print("Riwayat Pesanan:")  
        total = 0  
        for i, order in enumerate(user_orders):  
            print(f"{i + 1}. {order['pesanan']} -  
Jumlah: {order['jumlah']} - Harga per unit:  
{order['harga']}")  
            total += order['harga'] * order['jumlah']  
        print(f"Total Harga: Rp. {total}")  
    else:  
        print("Belum ada riwayat pesanan.")  
        input("Klik Enter Untuk Melanjutkan...")
```

Gambar 3. 33 Fungsi history user

- Mendefinisikan historyuser(), sebuah fungsi untuk melihat riwayat pesanan pengguna tertentu.
- Menampilkan pesanan untuk pengguna beserta total harga.



```
def login():
    while True:
        clear()
        print("=== LOGIN ===")
        username = input("Masukkan Username: ")
        password = input("Masukkan Password: ")
        for user in users:
            if user[0] == username and user[1] == password:
                print(f"Login Berhasil. Selamat Datang {user[3]}!")
                input("Klik Enter Untuk Melanjutkan...")
                if user[2] == "admin":
                    menuadmin()
                else:
                    menuuser(user[0], user[3])
                return
        print("Username atau Password salah.")
        input("Klik Enter Untuk Melanjutkan...")
```

Gambar 3. 34 Fungsi login

- Mendefinisikan login(), sebuah fungsi untuk login pengguna.
- Meminta username dan password, kemudian memeriksa kecocokannya dengan daftar users.
- Memanggil menuadmin() jika pengguna adalah admin, atau menuuser() jika pengguna adalah pengguna biasa.

```
Fungsi Lihat Pesanan

def lihatpesanan():
    if len(pesanan) == 0:
        print("Tidak Ada Pesanan")
        input("Klik Enter Untuk Melanjutkan...")
    else:
        for index, (user_name, user_orders) in enumerate(pesanan.items(), start=1):
            total = sum(order['harga'] * order['jumlah'] for order in user_orders)
            print(f"\n {index}. Pesanan oleh {user_orders[0]['nama']}:")
            for order in user_orders:
                print(f"    {order['pesanan']} - Jumlah : {order['jumlah']}")
            print(f"    Total Harga : Rp. {total}")
            input("Klik Enter Untuk Melanjutkan...")
```

Gambar 3. 35 Fungsi lihat pesanan

- Mendefinisikan `lihatpesanan()`, sebuah fungsi untuk menampilkan pesanan.
- Memeriksa apakah ada pesanan dan mencetaknya bersama dengan total harga.

```
Fungsi Menu User

def menuuser(username, namalengkap):
    while True:
        clear()
        print("=== MENU USER ===")
        print("1. Lihat Menu")
        print("2. Pesan")
        print("3. Lihat History")
        print("0. Logout")
        pilihan = input("Pilih Menu: ")
        if pilihan == "1":
            lihatmenu()
        elif pilihan == "2":
            pesan(username, namalengkap)
        elif pilihan == "3":
            historyuser(username)
        elif pilihan == "0":
            break
        else:
            print("Pilihan tidak valid.")
            input("Klik Enter Untuk Melanjutkan...")
```

Gambar 3. 36 Fungsi menu user

```

    Fungsi Menu Utama

while True:
    clear()
    print("=" * 40)
    print("          KAFE RUANG RASA  ")
    print("    Silahkan login terlebih dahulu")
    print("    Register jika belum punya akun")
    print("=" * 40)
    print(" [1]. Register")
    print(" [2]. Login")
    print(" [3]. Keluar")
    print("-" * 40)
    pilihan = tidakbolehdkosong("Pilih menu: ")
    input("Klik enter untuk melanjutkan")
    clear()

    if pilihan == "1":
        print("=" * 40)
        print("          PILIH ROLE")
        print("=" * 40)
        print("[1]. Registrasi sebagai admin \n[2]. Registrasi sebagai pengunjung")
        print("-"*40)
        registrasi = tidakbolehdkosong("Masukkan Pilihan : ")
        input("Klik enter untuk melanjutkan")
        clear()

```

Gambar 3. 37 Fungsi menu utama

BAB IV

PENUTUP

4.1 Kesimpulan

Program ini adalah sebuah aplikasi kasir sederhana untuk sebuah kafe yang dibuat menggunakan Python. Program ini memungkinkan pengguna untuk melakukan autentikasi (sebagai admin atau pengunjung), melihat menu, menambah pesanan, melihat pesanan, mengubah pesanan, dan melihat riwayat Pembelian. Admin memiliki opsi tambahan untuk menambah, mengupdate, dan menghapus item menu.


4.2 Saran

1. Menambahkan pengecekan validasi input, terutama untuk angka, agar aplikasi tidak menerima data yang tidak valid.
2. Hash kata sandi pengguna sebelum menyimpannya untuk meningkatkan keamanan, gunakan modul hashlib.
3. Menyimpan data pengguna, menu, dan pesanan ke file (JSON atau CSV) agar data tetap ada antara sesi penggunaan aplikasi.



DAFTAR PUSTAKA

- Kelas Terbuka. *“Belajar Python Bahasa Indonesia”*. 2020. Youtube,
<https://www.youtube.com/watch?v=iA8lLwmtKQM&list=PLZS-MHyEIRo59lUBwU-XHH7Ymmb04ffOY&index=1&t=6s>
- w3school. *“Python Tutorial”*. w3school.com,
<https://www.w3schools.com/python/>

LAMPIRAN


<p>Aturan Konsultasi :</p> <p>A. Kartu Konsul wajib dibawa saat dilakukan konsultasi</p> <p>B. Ketua Kelompok dan Anggota diwajibkan untuk hadir tiap konsultasi</p> <p>C. Konsul dilaksanakan minimal 2 kali dengan ketentuan sebagai berikut :</p> <ul style="list-style-type: none"> - Konsul 1 : Konsep Program - Konsul 2 : Penyelesaian Program <p>N.B : Batas waktu konsultasi ialah H+7 untuk konsul 1 dan H+14 untuk konsul 2 (Dimulai sejak pengumuman dan pembentukan kelompok PA)</p>	<p>Manajemen Menu dan Pemesanan Kafe</p> <p>Algoritma dan Pemrograman Dasar</p> <p>KELAS IF 24 B1:</p> <p>KELOMPOK 3:</p> <p>Wahyu Aditya/240910606867 (Ketua)</p> <p>Andi Anugrah Alif Saputra/2409106058</p> <p>Inayah Ramadhani/2409106068</p>  <p>LABORATORIUM FAKULTAS TEKNIK UNIVERSITAS MULAWARMAN 2024</p>
--	--

Gambar 4. 1 Kartu konsul 1

<p>Tanggal Konsultasi : Juniol , 1 November 2024</p> <p>Urutan / Pembahasan :</p> <ul style="list-style-type: none"> - Autentikasi login username tidak boleh sama - Tambahkan riwayat pembelian - Admin bisa melihat semua riwayat - pelanggan hanya bisa melihat riwayat pembelian masing-masing 	
<p>Asisten Lab</p>  <p>Nama: Adi Muhammad S.</p>	<p>Ketua Kelompok</p>  <p>Nama: Wahyu Aditya</p>

<p>Tanggal Konsultasi : 14 November 2024</p> <p>Urutan / Pembahasan :</p> <ul style="list-style-type: none"> - Input menu tidak boleh ada spasi atau karakter lain, contoh "C1 1" - Memberikan tampilan program - Memberi jeda dulu sebelum masuk ke menu yang dipilih 	
<p>Asisten Lab</p> <p>Nama: Adi Muhammad S.</p>	<p>Ketua Kelompok</p> <p>Nama: Wahyu Aditya</p>

Gambar 4. 2 Kartu konsul 2

Tanggal Konsultasi: 22 November 2024	
Uraian / Pembahasan : - Penambahan fungsi untuk menginput library pada program - Penambahan terminator end pada reference - Penambahan tampilan terminal tiap fungsi di laporan	
Asisten Lab	Ketua Kelompok
Nama: Adi Muhammad S.	 Nama: Wahyu Aditya

CS

Gambar 4. 3 Kartu konsul 3