

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



APLIKASI SISTEM ANGKRINGAN

Oleh:

Kelompok 1

AHMAD AIDIL	2409106080
AULIA NATASYA	2409106084
MUHAMMAD	2409106090
FEBRIANSYAH	

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

KATA PENGANTAR

Segala puji dan syukur kami panjatkan kepada Tuhan yang Maha Esa atas limpahan rahmat dan hidayah-Nya, sehingga laporan proyek akhir praktikum yang berjudul “Aplikasi Sistem Angkringan” dapat terselesaikan dengan baik. Laporan proyek akhir praktikum ini disusun sebagai bentuk pertanggungjawaban pelaksanaan kegiatan praktikum, sekaligus sebagai dokumentasi hasil kerja, mengevaluasi pemahaman, serta melatih kemampuan analisis dan penulisan ilmiah. Kami berharap laporan ini dapat memberikan manfaat sebagai bahan referensi dan evaluasi untuk pembelajaran di masa mendatang.

Pada program yang berjudul ”Aplikasi Sistem Angkringan” ini dirancang untuk mendukung pengelolaan operasional sebuah angkringan, termasuk login dan registrasi pengguna, pengelolaan menu oleh admin, hingga transaksi pembelian oleh admin, hingga transaksi pembelian oleh pelanggan. Program ini juga dilengkapi dengan fitur Riwayat transaksi dan pemberian diskon berdasarkan hari tertentu atau kondisi khusus, seperti pembelian bersama pasangan.

Selama proses pembuatan program, kami menghadapi beberapa kendala, seperti kurangnya pemahaman mengenai penggunaan file CSV dan data external, pengelolaan waktu yang kurang optimal, sikap yang cenderung menunda pekerjaan. Selain itu, minimnya komunikasi antar anggota kelompok sempat menjadi tantangan, yang disebabkan oleh kesibukan masing-masing dalam kegiatan kepanitiaan sebelumnya. Namun, masalah komunikasi ini berhasil diselesaikan melalui diskusi dan kerja sama yang lebih intensif selama proses pengerjaan. Proyek ini dapat diselesaikan berkat arahan dan dukungan dari Bang Afrizal selaku pembimbing proyek akhir praktikum, serta teman-teman yang selalu bersedia membantu, menjawab berbagai pertanyaan, dan memberikan semangat. Tak lupa, kami juga berterima kasih kepada rekan satu kelompok yang telah bekerja sama dengan baik dan saling mendukung satu sama lain hingga proyek akhir ini dapat terselesaikan dengan baik.

Laporan ini kami susun dengan harapan dapat memberikan Gambaran yang jelas mengenai program yang telah dibuat. Kami menyadari bahwa laporan ini mungkin masih memiliki kekurangan. Oleh karena itu, saran dan kritik yang membangun sangat

kami harapkan agar dapat menjadi bahan evaluasi di masa mendatang. Semoga laporan ini bermanfaat dan dapat menjadi inspirasi bagi pengembangan program serupa.

Samarinda, 10 November 2024

Kelompok 1

TAKARIR

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

<i>Database</i>	Basis Data
<i>Managemen</i>	Mengatur
<i>Input</i>	Memasukkan
<i>Admin</i>	Pengelola Sistem
<i>Menu</i>	Daftar Pilihan
<i>Transaction</i>	Transaksi
<i>Discount</i>	Diskon
<i>Log</i>	Catatan

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	1
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
2.3 Konsep/Materi Praktikum yang dipakai	6
BAB III HASIL DAN PEMBAHASAN	8
3.1 Tampilan Program	8
3.1.1 Tampilan Awal Program	8
3.1.2 Tampilan Login	8
3.1.3 Tampilan Menu Pembeli	10
3.1.4 Tampilan Daftar Menu	10
3.1.5 Tampilan Pembelian Menu	10
3.1.6 Tampilan Riwayat Pembelian	11
3.1.7 Tampilan Admin Menu	12
3.1.8 Tampilan Tambah Menu (Admin)	12

3.1.9	Tampilan Update Menu (Admin)	13
3.1.10	Tampilan Hapus Menu (Admin).....	13
3.2	Source Code.....	15
BAB IV PENUTUP		31
4.1	Kesimpulan	31
4.2	Saran.....	31
DAFTAR PUSTAKA		32
LAMPIRAN		33

DAFTAR GAMBAR

Gambar 1 Flowchart.....	5
Gambar 2 Tampilan Program	8
Gambar 3 Tampilan Program	8
Gambar 4 Tampilan Program	9
Gambar 5 Tampilan Program	9
Gambar 6 Tampilan Program	10
Gambar 7 Tampilan Program	10
Gambar 8 Tampilan Program	11
Gambar 9 Tampilan Program	11
Gambar 10 Tampilan Program	12
Gambar 11 Tampilan Program	12
Gambar 12 Tampilan Program	13
Gambar 13 Tampilan Program	13
Gambar 14 Tampilan Program	14
Gambar 15 Tampilan Program	14
Gambar 16 Tampilan Program	15
Gambar 17 Tampilan Program	15
Gambar 18 Fungsi untuk memvalidasi bahwa input hanya mengandung huruf dan angka	16
Gambar 19 Fungsi untuk membaca data pengguna dari file CSV.	16
Gambar 20 Fungsi menampilkan menu angkringan.	17
Gambar 21 Fungsi menampilkan menu anfkringan.	17
Gambar 22 Fungsi menampilkan menu angkringan yang hanya untuk beli menu.	18
Gambar 23 Fungsi menghitung total stok semua barang.....	18
Gambar 24 Fungsi login dengan validasi karakter khusus.	19
Gambar 25 Fungsi logout.	19
Gambar 26 Fungsi menghitung diskon.	20
Gambar 27 Fungsi menyimpan Riwayat pembelian.	20
Gambar 28 Fungsi menampilkan riwayat pembelian.	21
Gambar 29 Fungsi registrasi pengguna baru dengan validasi karakter khusus.	22
Gambar 30 Fungsi menampilkan daftar pembeli khusus admin.	23
Gambar 31 Fungsi membeli menu.	24
Gambar 32 Fungsi utama program.....	25
Gambar 33 Fungsi menghapus menu.....	26
Gambar 34 Fungsi menu admin.	27
Gambar 35 Fungsi untuk mengupdate menu.....	28
Gambar 36 Fungsi menu pembeli.	29

Gambar 37 Fungsi menambah menu baru.	30
Gambar 38 Lembar Konsultasi.....	33
Gambar 39 Lembar Konsultasi.....	34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Angkringan adalah warung kecil yang biasanya berlokasi di pinggir jalan atau trotoar. Usaha ini umumnya terdapat di Indonesia, khususnya di Daerah Istimewa Yogyakarta yang dikenal kuat dalam ekonomi kreatif yaitu sektor kuliner. Angkringan tidak hanya menawarkan makanan sederhana, tetapi juga menjadi tempat berkumpul yang erat dengan kehidupan masyarakat. Namun, di era modern saat ini, banyak angkringan yang belum memanfaatkan teknologi dalam pengelolaan menu, transaksi, dan administrasi pengguna secara lebih efisien dan terstruktur. Oleh karena itu, dibutuhkan sebuah program manajemen angkringan yang dapat mempermudah pengelolaan menu, memonitor transaksi, dan mempercepat proses administrasi, sehingga dapat meningkatkan kualitas layanan dan produktivitas usaha, sehingga angkringan dapat terus berkembang dan tetap relevan dalam menghadapi tantangan zaman.

1.2 Kebutuhan Fungsional

Program sistem manajemen angkringan ini memiliki sejumlah kebutuhan fungsional untuk dapat mendukung operasionalnya. Dalam hal manajemen pengguna, sistem menyediakan sebuah fitur login untuk memverifikasi username dan password, dan memastikan hanya pengguna terdaftar yang dapat mengakses sistem. Selain itu, terdapat fitur registrasi untuk menambahkan pengguna baru, baik dengan peran sebagai admin maupun pembeli. Sistem juga memungkinkan pengguna untuk keluar dari akun melalui fitur logout. Data pengguna disimpan dalam file CSV untuk keperluan akses di masa mendatang. Fitur utama lainnya adalah manajemen menu angkringan yang mencakup kemampuan untuk menampilkan daftar menu lengkap dengan informasi harga dan stok. Admin

memiliki akses untuk menambah menu baru, memperbarui harga atau stok menu yang sudah ada, dan menghapus menu dari daftar. Admin juga dapat menghitung total stok semua barang dalam menu. Untuk mendukung pembelian, sistem memungkinkan pembeli untuk memilih barang, menentukan jumlah pembelian, dan otomatis memperbarui stok setelah transaksi. Sistem juga menghitung total belanja, termasuk penerapan diskon berdasarkan kondisi tertentu, seperti diskon 3% pada hari Minggu, diskon 5% untuk pembelian bersama pasangan, dan diskon 7% untuk pembelian di atas Rp100.000. Pembeli juga dapat melihat riwayat pembelian yang mencakup detail barang yang dibeli, total belanja, diskon, serta total setelah diskon.

1.3 Rumusan Masalah

Bagaimana cara merancang dan mengimplementasikan program sistem aplikasi berbasis Python yang dapat membantu mengelola menu, mencatat transaksi, serta memberikan fitur diskon otomatis untuk mendukung efisiensi dan kenyamanan operasional angkringan?

1.4 Batasan Masalah

Program ini dirancang untuk mencakup beberapa fitur utama, yaitu manajemen menu, stok, login, dan pencatatan transaksi. Fitur manajemen menu memungkinkan pengelolaan daftar makanan atau item yang tersedia di angkringan, sementara manajemen stok digunakan untuk memantau jumlah persediaan makanan yang ada. Sistem login mengharuskan pengguna untuk memasukkan username dan password, yang divalidasi agar tidak mengandung karakter khusus, untuk mencegah masalah keamanan. Selain itu, sistem diskon hanya berlaku dalam kondisi tertentu, seperti pembelian dengan pasangan atau dalam jumlah besar, serta pada hari-hari tertentu

yang telah ditentukan. Data pengguna dan riwayat transaksi disimpan dalam format file CSV, bukan dalam sistem basis data yang lebih kompleks, sehingga memudahkan pengelolaan data secara sederhana namun efisien. Batasan-batasan ini diterapkan untuk memastikan bahwa pembahasan aplikasi tetap fokus dan tidak meluas ke fitur-fitur yang tidak diperlukan.

1.5 Tujuan

Tujuan dari program ini adalah untuk menyediakan solusi digital yang membantu pemilik angkringan mengelola menu, mencatat transaksi, dan memberikan pelayanan yang lebih terstruktur kepada pelanggan. Dengan demikian, proses operasional menjadi lebih efisien, data penjualan lebih terorganisir, dan pelanggan mendapatkan pengalaman berbelanja yang lebih baik.

BAB II

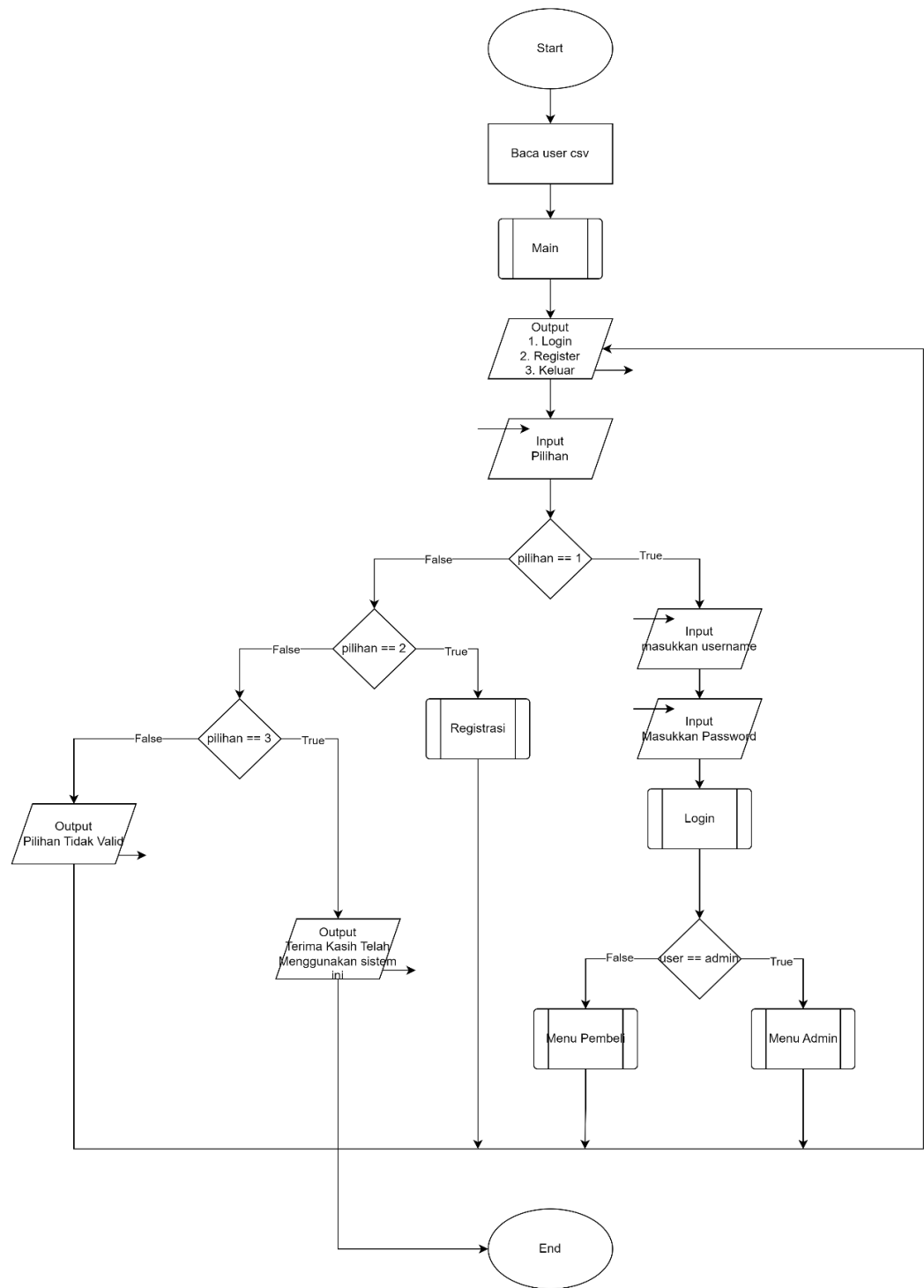
PERANCANGAN

2.1 Analisis Program

Program ini adalah sebuah aplikasi sistem angkringan yang dibuat untuk mengelola pengguna, menu, pembelian, serta riwayat transaksi. Program menggunakan Python dengan fitur berbasis fungsi. Sistem memiliki dua jenis pengguna, yaitu admin dan pembeli, dengan fitur login untuk membedakan akses. Data pengguna dikelola dalam format CSV untuk memudahkan penyimpanan dan pembacaan. Menu angkringan diatur dalam bentuk dictionary yang mencatat harga dan stok. Fitur-fitur utama meliputi menampilkan menu, menambahkan, mengupdate, dan menghapus menu, serta menghitung total stok yang tersedia.

Untuk pembeli, sistem memungkinkan mereka melihat daftar menu, membeli item, menghitung diskon berdasarkan kondisi tertentu seperti hari pembelian atau belanja bersama pasangan, dan menyimpan riwayat pembelian. Admin memiliki akses untuk mengelola data pembeli, menambahkan menu baru, dan melihat daftar pengguna. Semua interaksi dalam program ini dilakukan melalui tampilan terminal, dengan pemeriksaan untuk memastikan data yang dimasukkan sesuai dengan format yang benar. Program ini juga menggunakan file CSV untuk menyimpan data pengguna, sehingga data tetap tersimpan dan bisa diakses lagi setiap kali program dijalankan.

2.2 Flowchart



Gambar 1 Flowchart

Pada flowchart ini, alur program dimulai dengan membaca data pengguna yang tersimpan dalam file CSV. Data tersebut berisi informasi username, password, dan peran pengguna (admin atau pembeli). Setelah data berhasil dibaca, program masuk ke menu utama yang menawarkan tiga opsi, yaitu login, registrasi, dan keluar. Pengguna diminta untuk memilih salah satu opsi tersebut. Jika pengguna memilih opsi login, mereka harus memasukkan username dan password. Program kemudian memvalidasi data tersebut dengan informasi yang tersimpan di dalam file CSV. Jika pengguna adalah admin, mereka akan diarahkan ke menu pembeli untuk melakukan pembelian.

Jika pengguna memilih opsi registrasi, program akan mengarahkan mereka ke proses registrasi untuk menambahkan data pengguna baru. Sementara itu, jika pengguna memilih opsi keluar, program akan menampilkan pesan “Terimakasih telah menggunakan sistem ini” dan mengakhiri proses. Apabila pengguna memasukkan pilihan yang tidak valid, program akan menampilkan pesan “Pilihan tidak valid”. Selanjutnya, pengguna akan diarahkan kembali ke menu utama.

2.3 Konsep/Materi Praktikum yang dipakai

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

1. Fungsi dasar: fungsi yang kami gunakan adalah fungsi print. Fungsi print pada program merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga.
2. Fungsi percabangan: kami menggunakan struktur percabangan if, if/else, dan if/elif/else yang digunakan untuk mengontrol alur program. Struktur percabangan dibuat dengan tujuan menentukan tindakan dan perintah sesuai logika/kondisi

3. Fungsi perulangan: pada modul ini, kami memanfaatkan perulangan for dan while untuk memerintahkan komputer melakukan sesuatu secara berulang-ulang dengan jumlah yang ditentukan atau selama masih memenuhi suatu kondisi yang telah ditentukan.
4. Fungsi list dan tuple: fungsi untuk bekerja dengan list dan tuple digunakan dalam penyimpanan data menu dan riwayat pembelian. List digunakan untuk menyimpan item yang dapat diperbaharui, sedangkan tuple digunakan untuk data yang bersifat tetap dan tidak berubah, seperti data pengguna atau informasi terkait menu.
5. Fungsi dictionary: pada modul ini, kami menggunakan dictionary untuk menyimpan data pengguna dan menu.
6. Fungsi dan prosedur: fungsi digunakan untuk memisahkan logika program ke dalam bagian-bagian yang lebih kecil dan terstruktur, seperti fungsi untuk validasi input, login, registrasi, serta riwayat pembelian.
7. Error handling: kami menggunakan error handling untuk menangani potensi kesalahan input dari pengguna dan untuk memastikan program berjalan dengan baik.
8. File external: kami juga menggunakan file external, seperti file CSV untuk menyimpan data pengguna, yang memungkinkan program untuk memuat dan menyimpan data secara berkelanjutan.

BAB III

HASIL DAN PEMBAHASAN

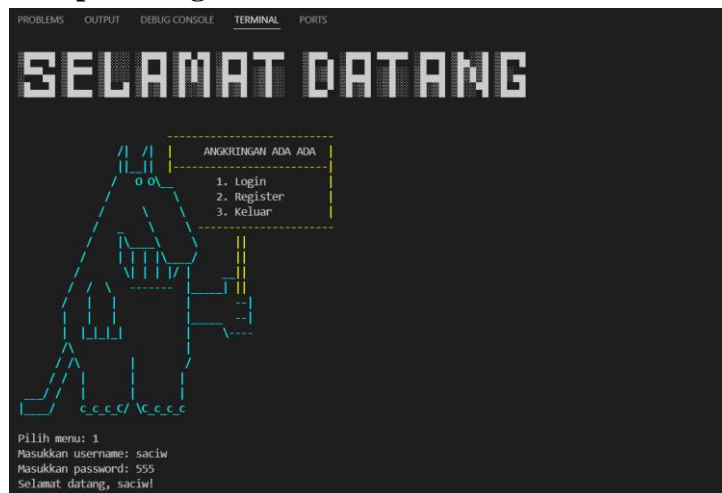
3.1 Tampilan Program

3.1.1 Tampilan Awal Program

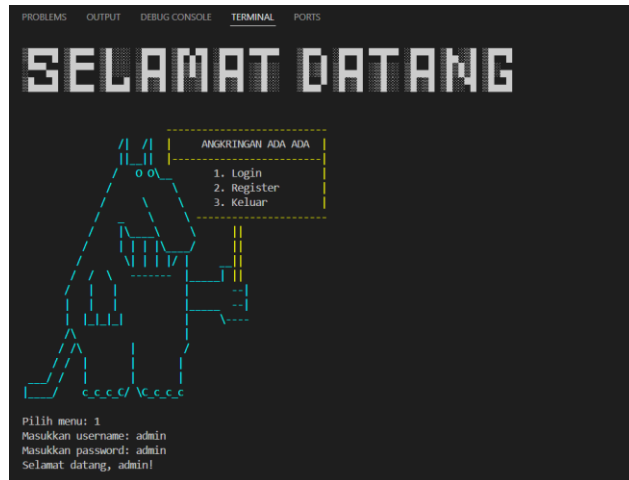


Gambar 2 Pada tampilan ini, pengguna diminta untuk memilih apakah ingin login atau mendaftar sebagai pengguna baru.

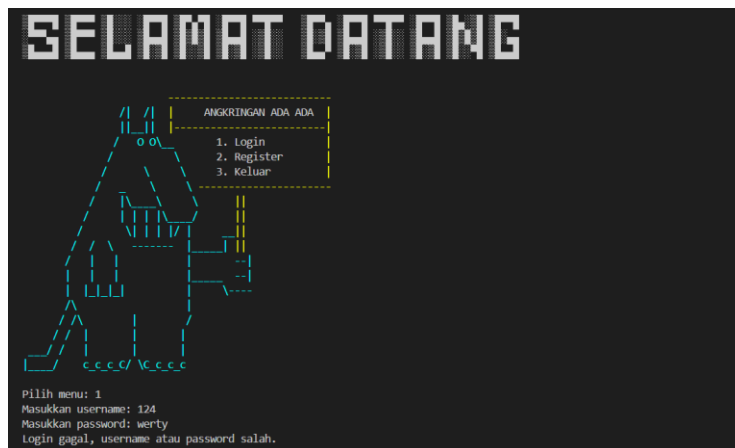
3.1.2 Tampilan Login



Gambar 3 Berikut tampilan output setelah login berhasil untuk pembeli.

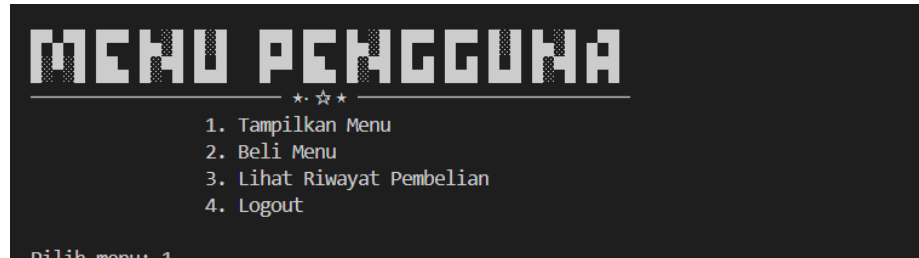


Gambar 4 Berikut adalah tampilan output setelah login sebagai admin.



Gambar 5 Jika login gagal, program akan menampilkan pesan kesalahan seperti pada gambar.

3.1.3 Tampilan Menu Pembeli



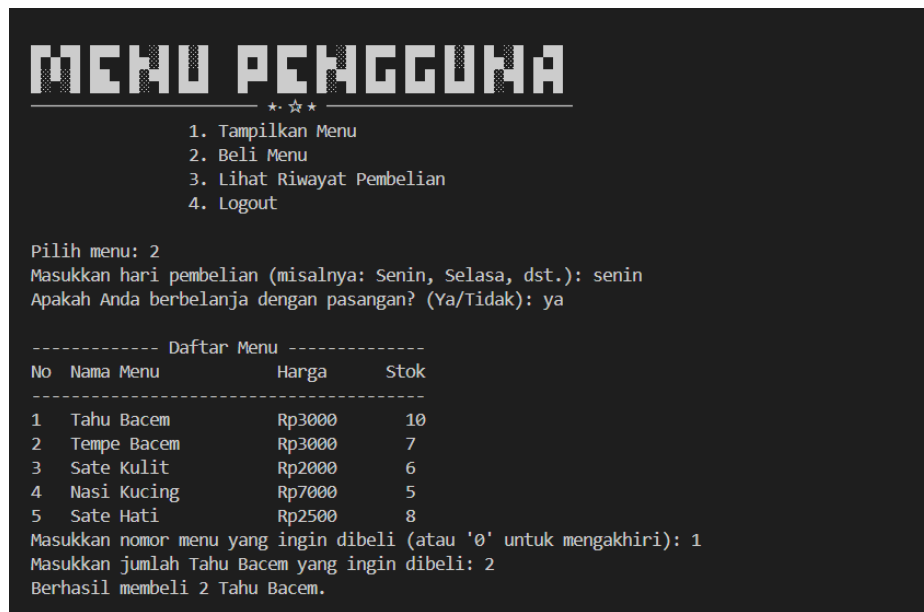
Gambar 6 Pada tampilan ini, pembeli dapat memilih menu untuk melihat daftar menu yang tersedia, membeli barang, melihat riwayat pembelian, atau logout.

3.1.4 Tampilan Daftar Menu

----- Daftar Menu -----			
No	Nama Menu	Harga	Stok
1	Tahu Bacem	Rp3000	10
2	Tempe Bacem	Rp3000	7
3	Sate Kulit	Rp2000	6
4	Nasi Kucing	Rp7000	5
5	Sate Hati	Rp2500	8

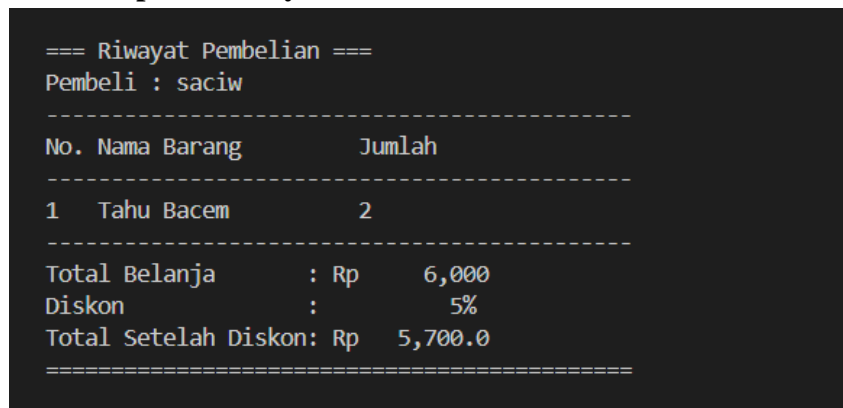
Gambar 7 Ketika pembeli memilih opsi "Tampilkan Menu", program akan menampilkan daftar menu yang tersedia

3.1.5 Tampilan Pembelian Menu



Gambar 8 Setelah memilih menu "Beli Menu", pembeli akan diminta untuk memilih barang yang ingin dibeli dan memasukkan jumlah yang diinginkan.

3.1.6 Tampilan Riwayat Pembelian



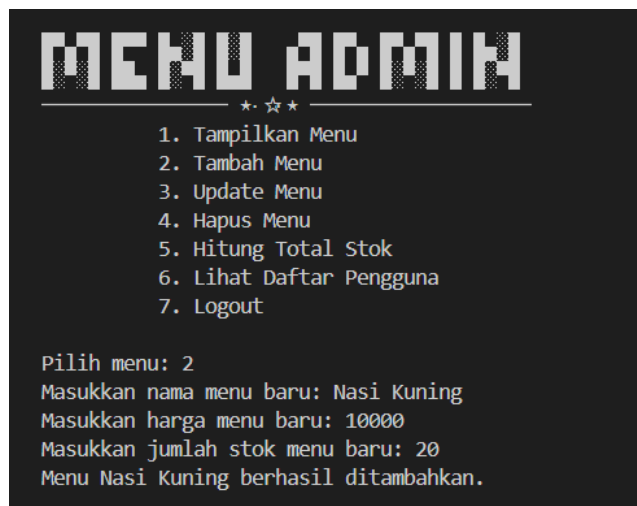
Gambar 9 Setelah membeli barang, pengguna dapat melihat riwayat pembelian mereka.

3.1.7 Tampilan Admin Menu



Gambar 10 Untuk pengguna dengan role "admin", ada pilihan menu admin yang dapat mengelola menu, menambah, mengubah, atau menghapus menu.

3.1.8 Tampilan Tambah Menu (Admin)



Gambar 11 Jika admin memilih untuk menambah menu baru, tampilan akan meminta input untuk nama menu, harga, dan stok..

3.1.9 Tampilan Update Menu (Admin)

```
Pilih menu: 3

--- Update Menu ---

Daftar Menu:
No  Nama Menu      Harga      Stok
-----
1   Tahu Bacem     Rp3000     8
2   Tempe Bacem    Rp3000     7
3   Sate Kulit     Rp2000     6
4   Nasi Kucing   Rp7000     5
5   Sate Hati      Rp2500     8
6   Nasi Kuning    Rp10000    20

Masukkan nomor menu yang ingin diupdate: 1
Masukkan harga baru: 1500
Masukkan stok baru: 12
Menu 'Tahu Bacem' berhasil diperbarui!
```

Gambar 12 Admin dapat memilih untuk mengupdate menu yang ada. Program akan menampilkan daftar menu yang ada dan meminta admin untuk memilih menu yang ingin diperbarui, termasuk harga dan stok baru

3.1.10 Tampilan Hapus Menu (Admin)

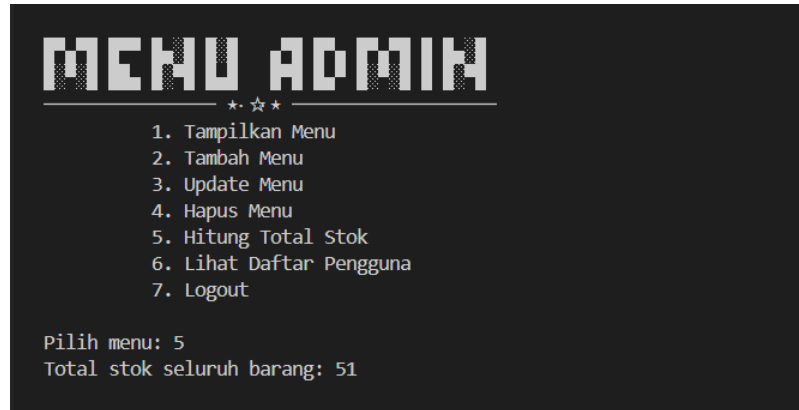
```
Pilih menu: 4

--- Hapus Menu ---
No  Nama Menu      Harga      Stok
-----
1   Tahu Bacem     Rp1500     12
2   Tempe Bacem    Rp3000     7
3   Sate Kulit     Rp2000     6
4   Nasi Kucing   Rp7000     5
5   Sate Hati      Rp2500     8
6   Nasi Kuning    Rp10000    20

Masukkan nomor menu yang ingin dihapus: 2
Apakah Anda yakin ingin menghapus menu 'Tempe Bacem'? (y/n): y
Menu 'Tempe Bacem' berhasil dihapus.
```

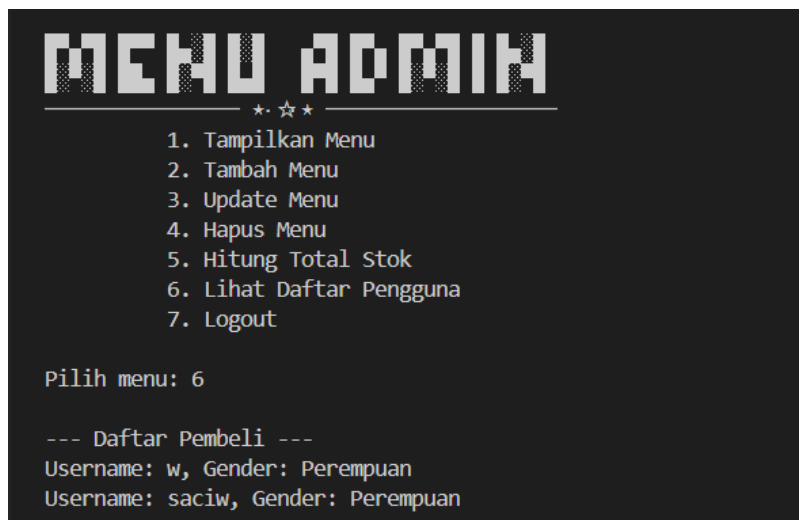
Gambar 13 Admin dapat memilih untuk menghapus menu yang ada. Program akan menampilkan daftar menu dan meminta admin memilih menu yang akan dihapus.

3.1.11 Tampilan Hitung Total Stok (Admin)



Gambar 14 Admin dapat melihat total stok seluruh barang.

3.1.12 Tampilan Lihat Daftar Pengguna (Admin)



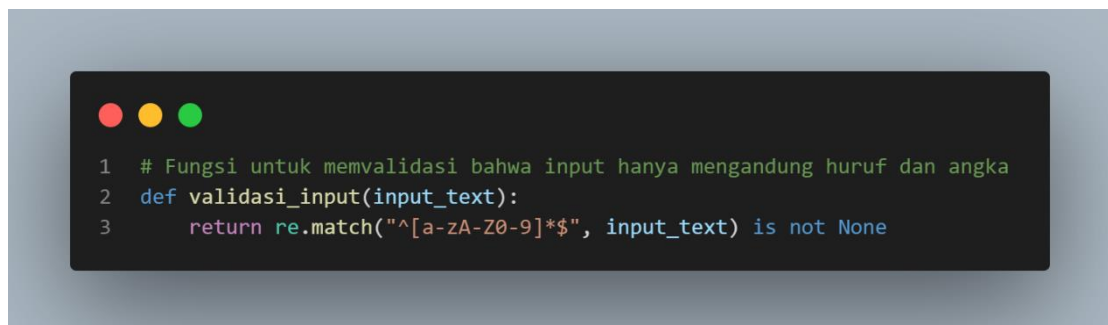
Gambar 15 Admin dapat melihat daftar pengguna.

3.1.13 Tampilkan Logout




Gambar 16 Tampilan logout admin.

3.2 Source Code




Gambar 17 fungsi untuk memvalidasi bahwa input hanya mengandung huruf dan angka.




```
1 def pembersihan():
2     os.system('cls' if os.name == 'nt' else 'clear')
```

Gambar 18 fungsi untuk memvalidasi bahwa input hanya mengandung huruf dan angka.



```
1 # Fungsi untuk membaca data pengguna dari file CSV
2 def baca_user_csv():
3     try:
4         with open('users.csv', mode='r', encoding='utf-8') as file:
5             reader = csv.reader(file)
6             next(reader) # Lewati header
7             for row in reader:
8                 if len(row) == 4:
9                     username, password, role, gender = row
10                    users[username] = {"password": password, "role": role, "gender": gender}
11     except FileNotFoundError:
12         pass # Jika file tidak ditemukan, biarkan daftar users tetap seperti awal
```

Gambar 19 fungsi untuk membaca data pengguna dari file CSV.




```

1 # Fungsi menampilkan menu angkringan
2 def tampilkan_menu():
3     print("\n----- Daftar Menu -----")
4     print(f"{'No':<3} {'Nama Menu':<20} {'Harga':<10} {'Stok':<5}")
5     print("-" * 40)
6     for i, (nama, info) in enumerate(menu.items(), start=1):
7         print(f"{i:<3} {nama:<20} Rp{info['harga']:<10} {info['stok']:<5}")
8     input("\nTekan Enter untuk kembali...")
9

```

Gambar 20 fungsi menampilkan menu angkringan.

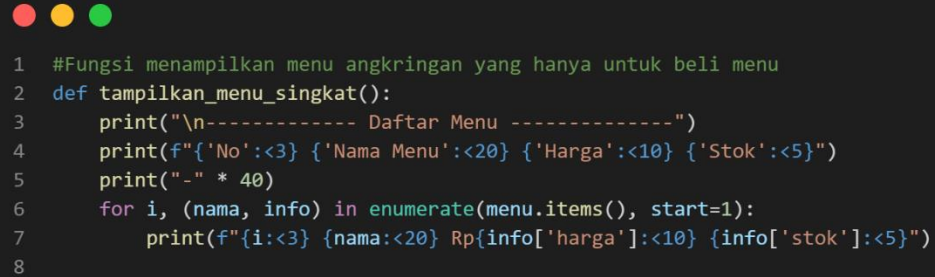


```

1 # Fungsi menampilkan menu angkringan
2 def tampilkan_menu():
3     print("\n----- Daftar Menu -----")
4     print(f"{'No':<3} {'Nama Menu':<20} {'Harga':<10} {'Stok':<5}")
5     print("-" * 40)
6     for i, (nama, info) in enumerate(menu.items(), start=1):
7         print(f"{i:<3} {nama:<20} Rp{info['harga']:<10} {info['stok']:<5}")
8     input("\nTekan Enter untuk kembali...")
9

```

Gambar 21 fungsi menampilkan menu anfkringan.

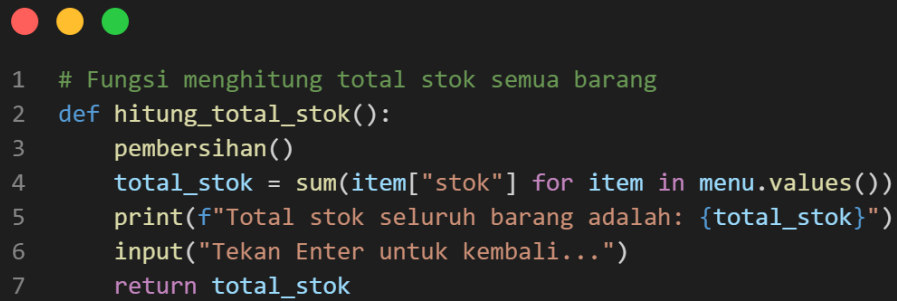


```

1 #Fungsi menampilkan menu angkringan yang hanya untuk beli menu
2 def tampilkan_menu_singkat():
3     print("\n----- Daftar Menu -----")
4     print(f"{'No':<3} {'Nama Menu':<20} {'Harga':<10} {'Stok':<5}")
5     print("-" * 40)
6     for i, (nama, info) in enumerate(menu.items(), start=1):
7         print(f"{i:<3} {nama:<20} Rp{info['harga']:<10} {info['stok']:<5}")
8

```

Gambar 22 fungsi menampilkan menu angkringan yang hanya untuk beli menu.

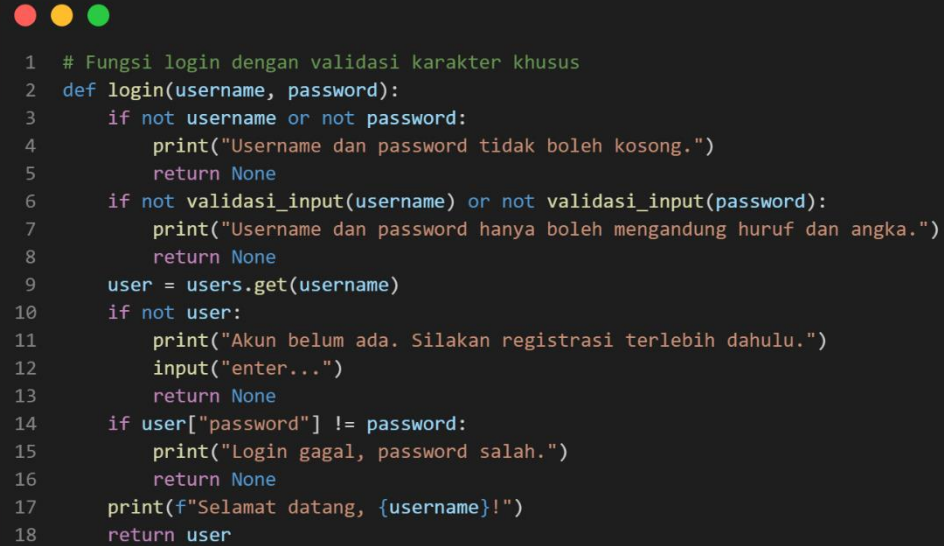


```

1 # Fungsi menghitung total stok semua barang
2 def hitung_total_stok():
3     pembersihan()
4     total_stok = sum(item["stok"] for item in menu.values())
5     print(f"Total stok seluruh barang adalah: {total_stok}")
6     input("Tekan Enter untuk kembali...")
7     return total_stok

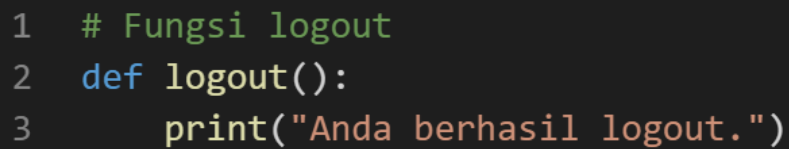
```

Gambar 23 fungsi menghitung total stok semua barang.



```
1 # Fungsi login dengan validasi karakter khusus
2 def login(username, password):
3     if not username or not password:
4         print("Username dan password tidak boleh kosong.")
5         return None
6     if not validasi_input(username) or not validasi_input(password):
7         print("Username dan password hanya boleh mengandung huruf dan angka.")
8         return None
9     user = users.get(username)
10    if not user:
11        print("Akun belum ada. Silakan registrasi terlebih dahulu.")
12        input("enter...")
13        return None
14    if user["password"] != password:
15        print("Login gagal, password salah.")
16        return None
17    print(f"Selamat datang, {username}!")
18    return user
```

Gambar 24 fungsi login dengan validasi karakter khusus.



```
1 # Fungsi logout
2 def logout():
3     print("Anda berhasil logout.")
```

Gambar 25 fungsi logout.

```

1  # Fungsi menghitung diskon
2  def hitung_diskon(total_belanja, hari, dengan_pasangan):
3      diskon = 0
4      if hari.lower() == "minggu":
5          diskon += 0.03
6      if dengan_pasangan.lower() == "ya":
7          diskon += 0.05
8      if total_belanja > 100000:
9          diskon += 0.07
10     return diskon

```

Gambar 26 fungsi menghitung diskon.

```

1  # Fungsi menyimpan riwayat pembelian
2  def lihat_riwayat_pembelian(username):
3      if username in riwayat_pembelian:
4          print("\n--- Riwayat Pembelian ---")
5          print(f"{'No':<3} {'Tanggal':<15} {'Items Dibeli':<25} {'Total Belanja':<15} {'Diskon':<10} {'Total Setelah Diskon':<20}")
6          print("-" * 90)
7
8          for idx, transaksi in enumerate(riwayat_pembelian[username], 1):
9              # Menyusun item-item yang dibeli
10             items = ", ".join([f'{item}: {jumlah} pcs' for item, jumlah in transaksi["items_dibeli"].items()])
11             print(f"{'idx':<3} {transaksi['tanggal']:<15} {items:<25} Rp{transaksi['total_belanja']:<15} {int(transaksi['diskon'] * 100):<10}% Rp{transaksi['total_setelah_diskon']:<20}")
12         else:
13             print("Belum ada riwayat pembelian.")
14
15     # Tunggu hingga pengguna menekan enter sebelum melanjutkan
16     input("Tekan enter untuk kembali ke menu...")

```

Gambar 27 fungsi menyimpan Riwayat pembelian.

```

1  # Fungsi menampilkan riwayat pembelian
2  def lihat_riwayat_pembelian(username, items_dibeli, total_belanja, diskon):
3      print("\n=== Riwayat Pembelian ===")
4      print(f"Pembeli : {username}")
5      print("-----")
6
7      if items_dibeli:
8          print(f'No.:<4{'Nama Barang':<20{'Jumlah':<10}')
9          print("-----")
10         for i, (item, jumlah) in enumerate(items_dibeli.items(), 1):
11             print(f'i:<4{item:<20{jumlah:<10}')
12             print("-----")
13             print(f'Total Belanja':<20): Rp{total_belanja:>10,}")
14             print(f'Diskon':<20): {int(diskon * 100):>10}%")
15             print(f'Total Setelah Diskon':<20): Rp{(total_belanja - (total_belanja * diskon)):>10,}")
16         else:
17             print("Belum ada riwayat pembelian.")
18
19     input("Tekan Enter untuk kembali ke menu...")
20     print("=====")
21

```

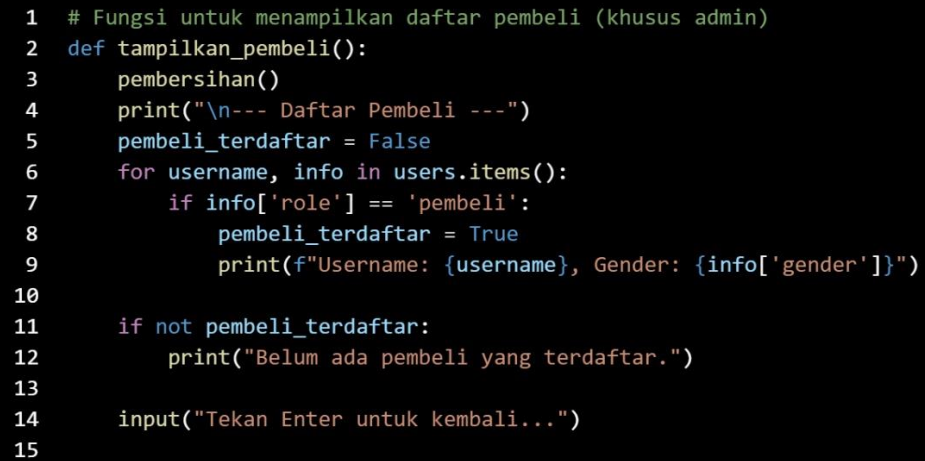
Gambar 28 fungsi menampilkan riwayat pembelian.

```

1 # Fungsi registrasi pengguna baru dengan validasi karakter khusus
2 def registrasi():
3     while True:
4         pembersihan() # Membersihkan layar
5         username = input("Masukkan username baru (hanya huruf dan angka): ").strip()
6
7         # Cek apakah username kosong
8         if not username:
9             print("Username tidak boleh kosong. Silakan coba lagi.")
10            continue # Kembali ke awal loop jika username kosong
11
12        # Cek apakah username hanya mengandung huruf dan angka
13        if not validasi_input(username):
14            print("Username hanya boleh mengandung huruf dan angka. Silakan coba lagi.")
15            continue # Kembali ke awal loop jika username tidak valid
16
17        # Cek apakah username sudah terdaftar
18        if username in users:
19            print("Nama pengguna sudah ada. Silakan gunakan username lain.")
20            input("enter...")
21            continue # Kembali ke awal loop jika username sudah terdaftar
22
23        # Jika username valid, lanjutkan untuk meminta password dan gender
24        password = input("Masukkan password baru (hanya huruf dan angka): ").strip()
25        if not password:
26            print("Password tidak boleh kosong. Silakan coba lagi.")
27            continue # Kembali ke awal loop jika password kosong
28
29        if not validasi_input(password):
30            print("Password hanya boleh mengandung huruf dan angka. Silakan coba lagi.")
31            continue # Kembali ke awal loop jika password tidak valid
32
33        gender = input("Masukkan gender (L/P): ").strip().upper()
34
35        if gender not in ["L", "P"]:
36            print("Gender tidak valid. Silakan coba lagi.")
37            continue # Kembali ke awal loop jika gender tidak valid
38
39        # Tentukan gender lengkap
40        gender_full = "Laki-Laki" if gender == "L" else "Perempuan"
41
42        # Simpan data pengguna ke dalam dictionary users
43        users[username] = {"password": password, "role": "pembeli", "gender": gender_full}
44        print(f"Pengguna {username} berhasil didaftarkan sebagai pembeli.")
45
46        # Simpan ke file CSV setelah registrasi
47        simpan_user_csv()
48        break # Keluar dari loop setelah pendaftaran berhasil
49

```

Gambar 29 fungsi registrasi pengguna baru dengan validasi karakter khusus.



```
1 # Fungsi untuk menampilkan daftar pembeli (khusus admin)
2 def tampilkan_pembeli():
3     pembersihan()
4     print("\n--- Daftar Pembeli ---")
5     pembeli_terdaftar = False
6     for username, info in users.items():
7         if info['role'] == 'pembeli':
8             pembeli_terdaftar = True
9             print(f"Username: {username}, Gender: {info['gender']}")
10
11     if not pembeli_terdaftar:
12         print("Belum ada pembeli yang terdaftar.")
13
14     input("Tekan Enter untuk kembali...")
15
```

Gambar 30 fungsi menampilkan daftar pembeli khusus admin.

```

1  # Fungsi membeli barang
2  def beli_barang(username):
3      pembersihan()
4      hari = input("Masukkan hari pembelian (misalnya: Senin, Selasa, dst.): ")
5      pasangan = input("Apakah Anda berbelanja dengan pasangan? (Ya/Tidak): ").strip().lower()
6      items_dibeli = {}
7      total_belanja = 0
8
9      while True:
10         tampilkan_menu_singkat() # Ganti dengan menu tanpa input
11         try:
12             pilihan = int(input("Masukkan nomor menu yang ingin dibeli (atau '0' untuk mengakhiri): ").strip())
13
14             if pilihan == 0:
15                 break
16
17             if pilihan < 1 or pilihan > len(menu):
18                 print("Pilihan tidak valid.")
19                 continue
20
21             nama_barang = list(menu.keys())[pilihan - 1]
22             jumlah = int(input(f"Masukkan jumlah {nama_barang} yang ingin dibeli: "))
23             if jumlah <= menu[nama_barang]["stok"]:
24                 menu[nama_barang]["stok"] -= jumlah
25                 items_dibeli[nama_barang] = items_dibeli.get(nama_barang, 0) + jumlah
26                 total_belanja += menu[nama_barang]["harga"] * jumlah
27                 print(f"Berhasil membeli {jumlah} {nama_barang}.")
28             else:
29                 print(f"Stok {nama_barang} tidak mencukupi.")
30         except ValueError:
31             print("Input tidak valid. Pastikan memasukkan angka dengan benar.")
32
33     if items_dibeli:
34         diskon = hitung_diskon(total_belanja, hari, pasangan)
35         return items_dibeli, diskon
36
37     # Jika tidak ada barang yang dibeli
38     return {}, 0

```

Gambar 31 fungsi membeli menu.

SECRET ORANGE

Kelompok 1 B2 2024 | 25

```

1  # Fungsi menghapus menu
2  def hapus_menu():
3      pembersihan()
4      print("\n--- Hapus Menu ---")
5      # Tampilkan daftar menu terlebih dahulu
6      if not menu:
7          print("Menu kosong. Tidak ada yang bisa dihapus.")
8          return
9
10     # Tampilkan daftar menu dengan format yang konsisten
11     print(f"{'No':<3} {'Nama Menu':<20} {'Harga':<10} {'Stok':<5}")
12     print("-" * 40)
13
14     menu_list = list(menu.keys())
15     for i, nama in enumerate(menu_list, start=1):
16         info = menu[nama]
17         print(f"{i:<3} {nama:<20} Rp{info['harga']:<10} {info['stok']:<5}")
18
19     try:
20         # Pilih menu berdasarkan angka
21         pilihan_menu = int(input("\nMasukkan nomor menu yang ingin dihapus: ").strip())
22         if pilihan_menu < 1 or pilihan_menu > len(menu_list):
23             print("Pilihan tidak valid.")
24             return
25
26         nama_menu = menu_list[pilihan_menu - 1]
27
28         # Konfirmasi penghapusan
29         konfirmasi = input(f"Apakah Anda yakin ingin menghapus menu '{nama_menu}'? (y/n): ").strip().lower()
30         if konfirmasi == 'y':
31             del menu[nama_menu]
32             print(f"Menu '{nama_menu}' berhasil dihapus.")
33         else:
34             print("Penghapusan dibatalkan.")
35     except ValueError:
36         print("Input tidak valid. Pastikan memasukkan angka dengan benar.")
37

```

Gambar 33 fungsi menghapus menu.

```

1  # Fungsi menu admin
2  def menu_admin():
3      while True:
4          pembersihan()
5          print("""
6
7  MENU ADMIN
8
9  ***
10         1. Tampilkan Menu
11         2. Tambah Menu
12         3. Update Menu
13         4. Hapus Menu
14         5. Hitung Total Stok
15         6. Lihat Daftar Pengguna
16         7. Logout
17 """)
18     pilihan = input("Pilih menu: ")
19
20     if pilihan == "1":
21         tampilkan_menu()
22     elif pilihan == "2":
23         tambah_menu()
24     elif pilihan == "3":
25         update_menu()
26     elif pilihan == "4":
27         hapus_menu()
28     elif pilihan == "5":
29         print(f"Total stok seluruh barang: {hitung_total_stok()}")
30     elif pilihan == "6":
31         tampilkan_pembeli()
32     elif pilihan == "7":
33         logout()
34         break
35     else:
36         print("Pilihan tidak valid.")
37

```

Gambar 34 fungsi menu admin.

```

1  # Fungsi untuk mengupdate menu
2  def update_menu():
3      pembersihan()
4      print("\n--- Update Menu ---")
5      # Tampilkan daftar menu terlebih dahulu
6      if not menu:
7          print("Menu kosong. Tidak ada yang bisa diupdate.")
8          return
9
10     # Tampilkan daftar menu dengan angka, konsisten dengan tampilkan_menu
11     print("\nDaftar Menu:")
12     print(f"{'No':<3} {'Nama Menu':<20} {'Harga':<10} {'Stok':<5}")
13     print("-" * 40)
14     menu_list = list(menu.keys())
15     for i, nama in enumerate(menu_list, start=1):
16         info = menu[nama]
17         print(f"{'i':<3} {'nama':<20} Rp{info['harga']:<10} {info['stok']:<5}")
18
19     try:
20         # Pilih menu berdasarkan angka
21         pilihan = int(input("\nMasukkan nomor menu yang ingin diupdate: ").strip())
22         if pilihan < 1 or pilihan > len(menu_list):
23             print("Pilihan tidak valid.")
24             return
25
26         nama_menu = menu_list[pilihan - 1]
27
28         # Input pembaruan
29         harga_baru = int(input("Masukkan harga baru: ").strip())
30         stok_baru = int(input("Masukkan stok baru: ").strip())
31
32         if harga_baru <= 0 or stok_baru < 0:
33             print("Harga harus lebih besar dari 0, dan stok tidak boleh negatif.")
34             return
35
36         # Update data menu
37         menu[nama_menu]["harga"] = harga_baru
38         menu[nama_menu]["stok"] = stok_baru
39         print(f"Menu '{nama_menu}' berhasil diperbarui!")
40     except ValueError:
41         print("Input tidak valid. Pastikan memasukkan angka dengan benar.")
42

```

Gambar 35 fungsi untuk mengupdate menu.

```

1  # Fungsi menu pembeli
2  def menu_pembeli(username):
3      items_dibeli = {} # Variabel untuk menyimpan barang yang dibeli
4      total_belanja = 0 # Total belanja awal
5      diskon = 0 # Diskon awal
6
7      while True:
8          pembersihan()
9          print("""
10
11  MENU PENGGUNA
12  ***
13
14          1. Tampilkan Menu
15          2. Beli Menu
16          3. Lihat Riwayat Pembelian
17          4. Logout
18  """)
19          pilihan = input("Pilih menu: ").strip()
20
21          if pilihan == "1":
22              tampilkan_menu()
23          elif pilihan == "2":
24              hasil_belanja, diskon_pembelian = beli_barang(username)
25              if hasil_belanja: # Jika ada barang yang dibeli
26                  for item, jumlah in hasil_belanja.items():
27                      items_dibeli[item] = items_dibeli.get(item, 0) + jumlah
28                      total_belanja += sum(menu[item]["harga"] * jumlah for item, jumlah in hasil_belanja.items())
29                      diskon = diskon_pembelian
30          elif pilihan == "3":
31              lihat_riwayat_pembelian(username, items_dibeli, total_belanja, diskon)
32          elif pilihan == "4":
33              logout()
34              break
35          else:
36              print("Pilihan tidak valid.")
37

```

Gambar 36 fungsi menu pembeli.

```
1 # Fungsi menambah menu baru
2 def tambah_menu():
3     pembersihan()
4     nama = input("Masukkan nama menu baru: ")
5
6     while True:
7         try:
8             harga = int(input("Masukkan harga menu baru: "))
9             break # Keluar dari loop jika input valid
10        except ValueError:
11            print("Harga harus berupa angka. Silakan coba lagi.")
12
13    while True:
14        try:
15            stok = int(input("Masukkan jumlah stok menu baru: "))
16            break # Keluar dari loop jika input valid
17        except ValueError:
18            print("Stok harus berupa angka. Silakan coba lagi.")
19
20    if nama not in menu:
21        menu[nama] = {"harga": harga, "stok": stok}
22        print(f"Menu {nama} berhasil ditambahkan.")
23    else:
24        print("Menu tersebut sudah ada.")
25
```

Gambar 37 fungsi menambah menu baru.

BAB IV

PENUTUP

4.1 Kesimpulan

Program aplikasi sistem angkringan berbasis python ini berhasil menyediakan berbagai fitur yang mendukung manajemen operasional angkringan. Fitur-fitur seperti login dengan validasi karakter khusus, registrasi pengguna, pengelolaan menu (tambah, hapus, dan update), pembelian menu dengan perhitungan diskon berdasarkan hari dan kondisi tertentu, serta riwayat pembelian memberikan kemudahan bagi pengguna maupun admin dalam menjalankan kegiatan terkait angkringan. Program ini juga mengintegrasikan pengelolaan data pengguna dan menu melalui file CSV untuk memastikan data tetap tersimpan dengan aman dan mudah diakses kembali. Dengan struktur yang sistematis, program ini dapat dijadikan Solusi untuk kebutuhan manajemen angkringan skala kecil hingga menengah.

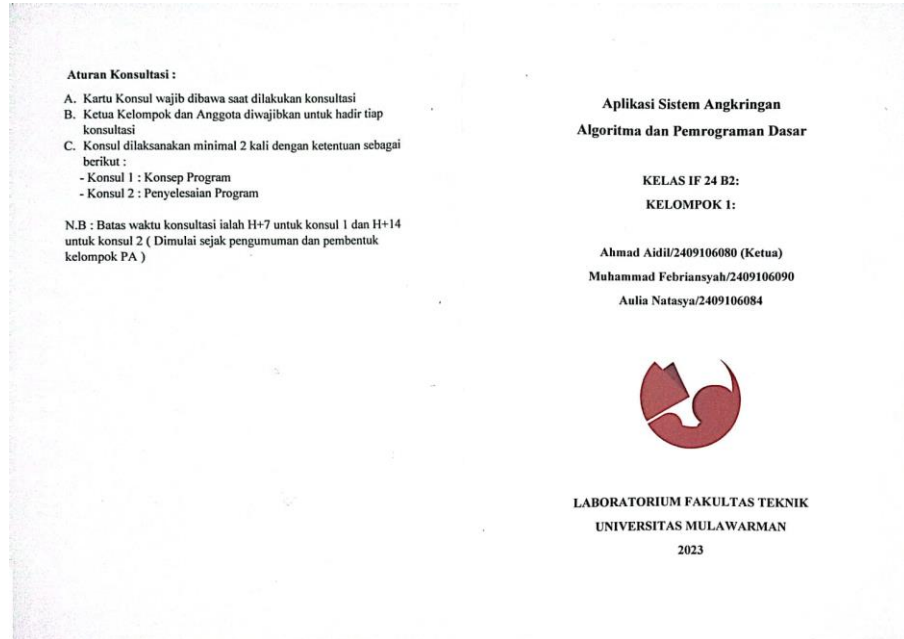
4.2 Saran

Untuk pengembangan ke depannya, ada beberapa hal yang bisa ditambahkan, seperti fitur pencarian menu supaya pembeli lebih mudah mencari item yang diinginkan, atau sistem pembayaran digital supaya lebih praktis dan sesuai dengan kebutuhan zaman sekarang. Tampilan program juga bisa dipikirkan untuk dibuat lebih menarik, supaya pengalaman pengguna semakin baik. Selain itu, optimasi kode juga bisa dilakukan agar program lebih cepat dan tidak terlalu banyak pengulangan.

DAFTAR PUSTAKA



Aulia, F. B., Wibowo, C. M., Ratu, M. A. L., Wiratsongko, R. T., Mufida, A. W., Sakurdin, L. & Wulansari, F. (2023). Pengembangan UMKM Angkringan Untuk Mendukung Perkembangan Ekonomi di Kemantren Tegalrejo Yogyakarta. *CITAKARYA Jurnal Pengabdian Masyarakat*.



LAMPIRAN



Gambar 38 Lembar Konsultasi

Gambar 39 Lembar Konsultasi

Tanggal Konsultasi : 11 November 2024	
Uraian / Pembahasan : <ul style="list-style-type: none"> • admin dapat melihat data user • user dapat melihat riwayat pemesanan seperti suhu pembelian • implementasi list jangan lupa 	
Asisten Lab  Nama: Muhammad Afrizal	Ketua Kelompok  Nama: Ahmad Aidil

Tanggal Konsultasi : 13 Desember 2024	
Uraian / Pembahasan : <ul style="list-style-type: none"> • username harus dibuat tidak sama • riwayat pemesanan di simpan di file external • struktur pembelian dirapikan dan dibuat lebih rinci seperti nama pembeli • tampilannya dirapikan • semua inputan jangan sampai kosong (untuk error handling) • harga jangan sampai 0 atau minus • semua inputan dibuat uppercase • nama makanan jangan sampai sama 	
Asisten Lab  Nama: Muhammad Afrizal	Ketua Kelompok  Nama: Ahmad Aidil

Gambar 40 Lembar Konsultasi

Gambar 41 Lembar Konsultasi