

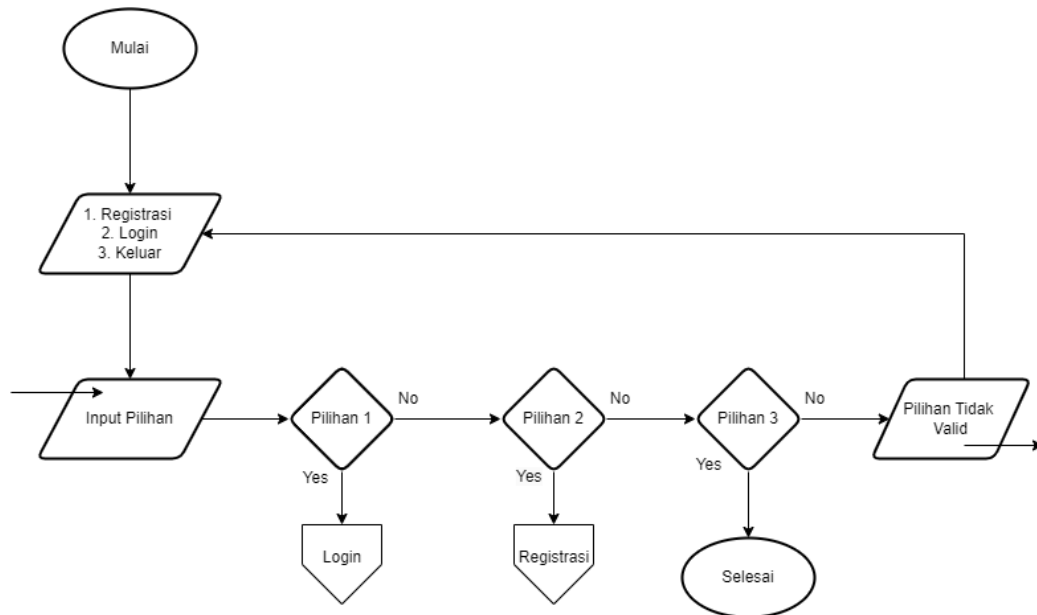
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN DASAR



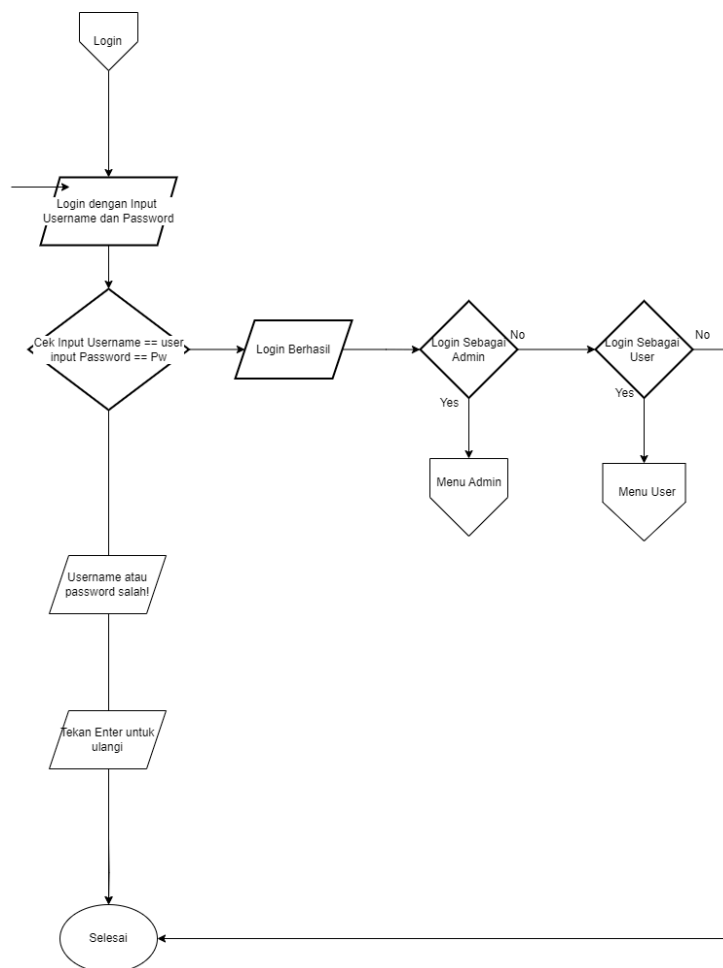
Disusun oleh:
Muhammad Geo Gilang Saputra
Kelas C2'25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

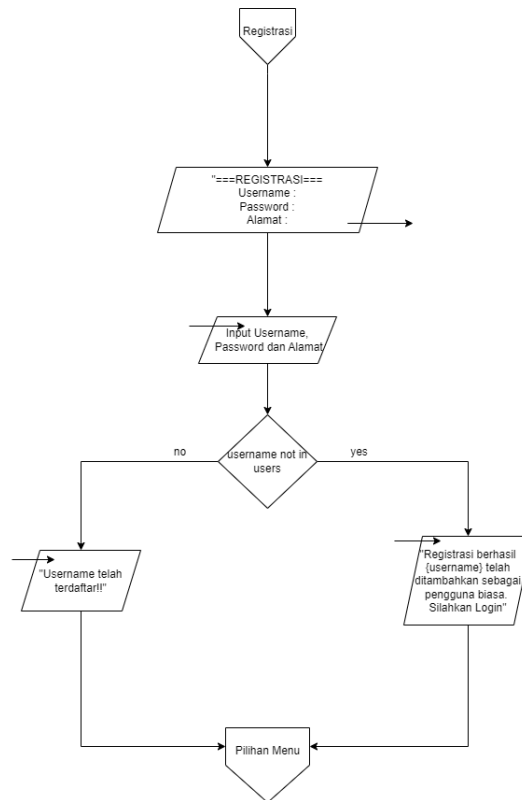
1. Flowchart



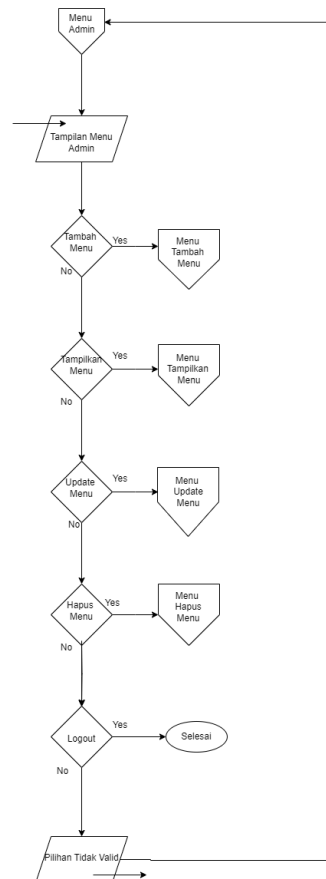
Gambar 1.1 Flowchart Halaman Awal



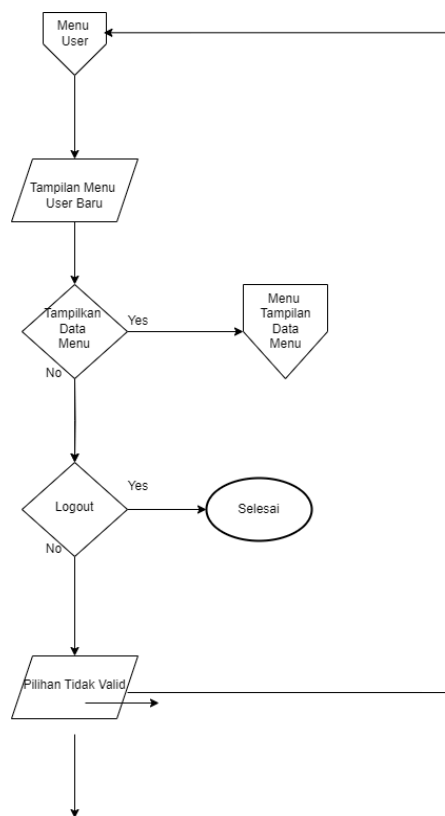
Gambar 1.2 Flowchart Halaman Login



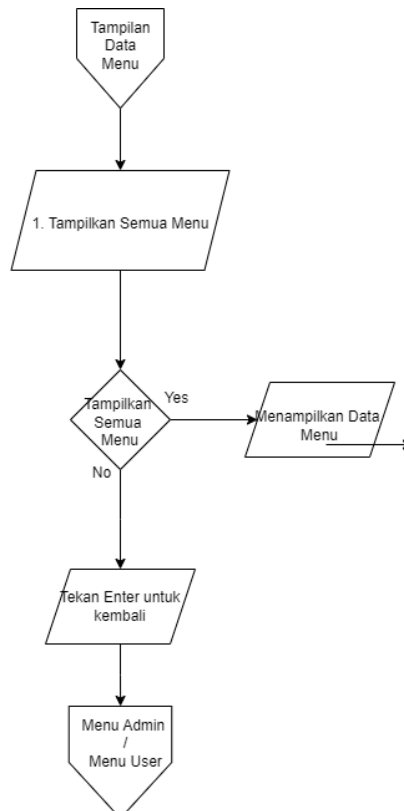
Gambar 1.3 Flowchart Halaman Registrasi



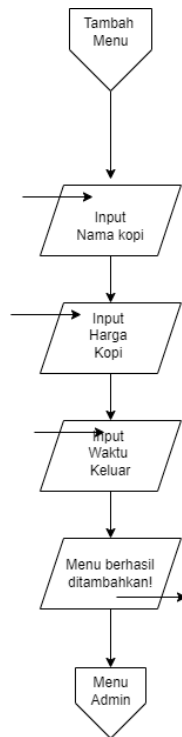
Gambar 1.4 Flowchart Halaman Menu Admin



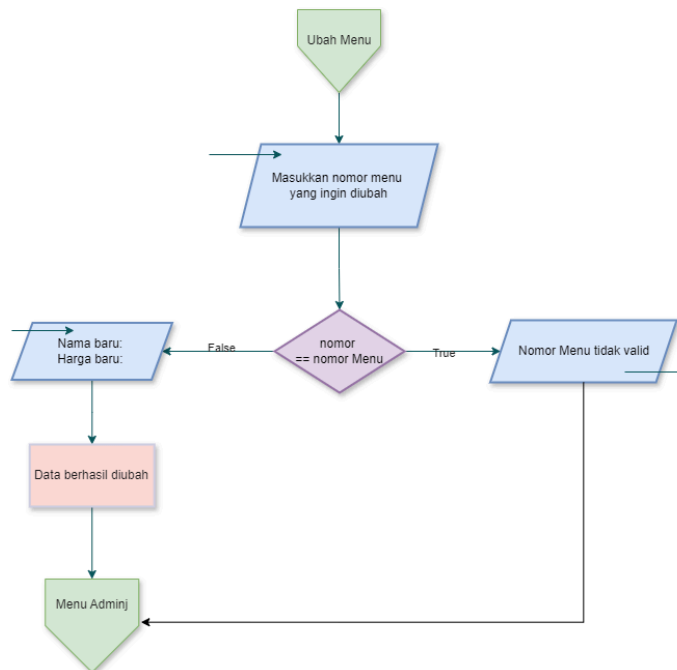
Gambar 1.5 Flowchart Halaman Menu User



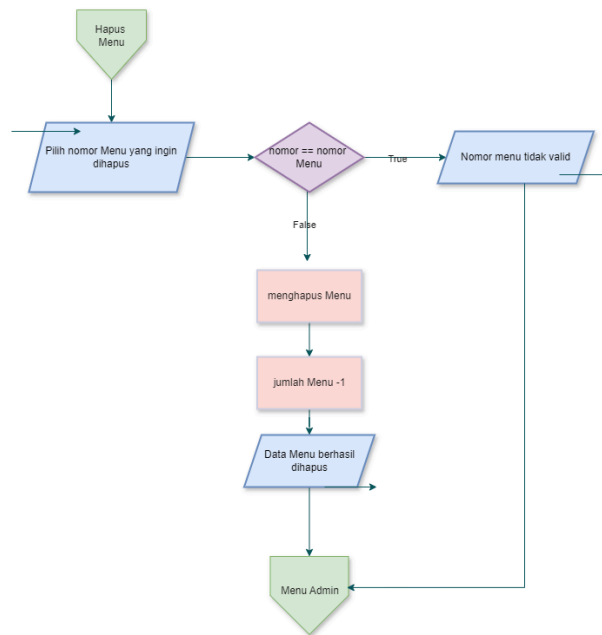
Gambar 1.6 Flowchart Halaman Tampilkan Menu



Gambar 1.7 Flowchart Halaman Tambah Menu



Gambar 1.8 Flowchart Halaman Ubah Menu



Gambar 1.8 Flowchart Halaman Hapus Menu

2. Deskripsi Singkat Program

Program **Manajemen Coffee Shop** merupakan aplikasi berbasis Python yang dirancang untuk membantu pengelolaan menu dan pengguna pada sebuah kedai kopi. Program ini menerapkan konsep **CRUD (Create, Read, Update, Delete)** menggunakan **List** dan **Nested List**, serta memiliki fitur **multiuser** yang membedakan antara **admin** dan **pengguna biasa (user)**.

Admin dapat menambah, melihat, mengubah, dan menghapus data menu kopi, sedangkan pengguna biasa hanya dapat melihat daftar menu yang tersedia.

Selain itu, program ini juga memiliki fitur **register dan login**, sehingga pengguna dapat membuat akun baru dan masuk sesuai hak aksesnya. Untuk memperjelas tampilan di terminal, digunakan perintah **clear screen** dari library **os**.

Program dibuat berdasarkan **modul pembelajaran List & Tuple**, dengan penerapan logika percabangan, perulangan, serta validasi input tanpa menggunakan try dan except.

- Register

Berfungsi untuk menambahkan akun pengguna baru.

Program akan meminta input **username** dan **password**, lalu menyimpannya ke dalam

list users. Jika username sudah ada, sistem akan menolak pendaftaran. Fungsi ini menjadikan program **multiuser** karena setiap pengguna dapat memiliki akun sendiri.

- Login

- Digunakan untuk memverifikasi akun yang sudah terdaftar.

Jika username dan password cocok, sistem akan mengembalikan **role** pengguna (admin, atau user) yang nantinya menentukan menu yang dapat diakses.

- Menu Admin

- Menampilkan dan mengatur seluruh fungsi CRUD menu kopi.

Admin dapat:

- ❖ Menambah menu baru (**Create**)
 - ❖ Melihat daftar menu (**Read**)
 - ❖ Mengubah menu (**Update**)
 - ❖ Menghapus menu (**Delete**)

Fungsi ini mewakili seluruh operasi manajemen data dalam Coffee Shop.

- Menu User

Menyediakan akses bagi pengguna biasa untuk **melihat daftar menu** tanpa bisa mengubahnya. Dengan ini, perbedaan hak akses antara **admin dan user** diterapkan.

read_menu () , create_menu () , update_menu () , dan delete_menu ()

Keempat fungsi ini merupakan implementasi langsung dari operasi CRUD:

- **create_menu ()** → menambahkan menu baru ke list.
 - **read_menu ()** → menampilkan seluruh menu kopi yang ada.
 - **update_menu ()** → mengubah nama atau harga menu tertentu.
 - **delete_menu ()** → menghapus menu kopi berdasarkan nomor indeks.

- Clear ()

- Membersihkan tampilan terminal agar program terlihat rapi setiap kali berpindah menu.

Menggunakan library os dengan perintah `os.system (' cls ' if os.name == ' nt ' else ' clear ')`.

3. Source Code

A. Tampilan Menu Admin

```
if role == "admin":
    while True:
        os.system('cls' if os.name == 'nt' else 'clear')
        print("=== MENU ADMIN COFFEE SHOP ===")
        print("1. Lihat Menu")
        print("2. Tambah Menu (Create)")
        print("3. Ubah Menu (Update)")
        print("4. Hapus Menu (Delete)")
        print("5. Logout")
        pilih_admin = input("Pilih menu: ")
```

B. Tampilan Menu User

```
elif role == "user":
    while True:
        os.system('cls' if os.name == 'nt' else 'clear')
        print("=== MENU PENGGUNA COFFEE SHOP ===")
        print("1. Lihat Menu")
        print("2. Logout")
        pilih_user = input("Pilih menu: ")

        if pilih_user == "1":
            os.system('cls' if os.name == 'nt' else 'clear')
            print("=== DAFTAR MENU COFFEE SHOP ===")
            if len(menu_coffee) == 0:
                print("Belum ada menu tersedia.")
            else:
                for i, item in enumerate(menu_coffee):
                    print(f"{i+1}. {item[0]} - Rp{item[1]}")
                input("\nTekan Enter untuk kembali...")

        elif pilih_user == "2":
            break
        else:
            print("⚠️ Pilihan tidak valid!")
            input("\nTekan Enter untuk lanjut...")
```


B. CRUD

```
# Lihat Menu

if pilih_admin == "1":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== DAFTAR MENU COFFEE SHOP ===")
    if len(menu_coffee) == 0:
        print("Belum ada menu tersedia.")
    else:
        for i, item in enumerate(menu_coffee):
            print(f"{i+1}. {item[0]} - Rp{item[1]}")
        input("\nTekan Enter untuk kembali...")

# Tambah Menu
elif pilih_admin == "2":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== TAMBAH MENU BARU ===")
    nama = input("Nama kopi: ")
    harga = input("Harga: ")
    if harga.isdigit():
        menu_coffee.append([nama, int(harga)])
        print("\n✅ Menu berhasil ditambahkan!")
    else:
        print("\n⚠️ Harga harus berupa angka!")
        input("\nTekan Enter untuk kembali...")

# Ubah Menu
elif pilih_admin == "3":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== UBAH MENU ===")
    if len(menu_coffee) == 0:
        print("Belum ada menu yang bisa diubah.")
    else:
        for i, item in enumerate(menu_coffee):
            print(f"{i+1}. {item[0]} - Rp{item[1]}")
        idx = input("Masukkan nomor menu yang ingin
diubah: ")

        if idx.isdigit():
            idx = int(idx) - 1
            if 0 <= idx < len(menu_coffee):
                nama = input("Nama baru: ")
                harga = input("Harga baru: ")
                if harga.isdigit():
                    menu_coffee[idx] = [nama,
int(harga)]

                print("\n✅ Menu berhasil diubah!")
            else:
                print("\n⚠️ Harga harus berupa
```

```

angka!")
        else:
            print("\n⚠️ Nomor menu tidak
ditemukan!")

        else:
            print("\n⚠️ Input tidak valid!")
            input("\nTekan Enter untuk kembali...")

# Hapus Menu
elif pilih_admin == "4":
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== HAPUS MENU ===")
    if len(menu_coffee) == 0:
        print("Tidak ada menu untuk dihapus.")
    else:
        for i, item in enumerate(menu_coffee):
            print(f"{i+1}. {item[0]} - Rp{item[1]}")
        idx = input("Masukkan nomor menu yang ingin
dihapus: ")

        if idx.isdigit():
            idx = int(idx) - 1
            if 0 <= idx < len(menu_coffee):
                del menu_coffee[idx]
                print("\n✅ Menu berhasil dihapus!")
            else:
                print("\n⚠️ Nomor menu tidak
ditemukan!")

        else:
            print("\n⚠️ Input tidak valid!")
            input("\nTekan Enter untuk kembali...")

```

4. Hasil Output

```
=== SISTEM MANAJEMEN COFFEE SHOP ☕ ===  
1. Login  
2. Register  
3. Keluar  
Pilih menu: █
```

Gambar 4.1 Hasil Output Tampilan Menu Utama

```
=== MENU ADMIN COFFEE SHOP ===  
1. Lihat Menu  
2. Tambah Menu (Create)  
3. Ubah Menu (Update)  
4. Hapus Menu (Delete)  
5. Logout  
Pilih menu: █
```

Gambar 4.2 Hasil Output Tampilan Menu Admin

```
=== REGISTER AKUN BARU ===  
Masukkan username baru: geo  
Masukkan password: 111  
  
✅ Akun berhasil dibuat!  
  
Tekan Enter untuk kembali... █
```

Gambar 4.3 Hasil Output Tampilan Menu Registrasi

```
=== MENU ADMIN COFFEE SHOP ===  
1. Lihat Menu  
2. Tambah Menu (Create)  
3. Ubah Menu (Update)  
4. Hapus Menu (Delete)  
5. Logout  
Pilih menu: █
```

Gambar 4.4 Hasil Output Tampilan Menu Admin

```
=== MENU PENGGUNA COFFEE SHOP ===
1. Lihat Menu
2. Logout
Pilih menu: █
```

Gambar 4.5 Hasil Output Tampilan Menu Pengguna

5. Langkah-langkah GIT

5.1 GIT Init

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git init
Reinitialized existing Git repository in C:/Users/HYPE AMD/OneDrive/Documents/praktikum-apd/.git/
```

Perintah ini adalah untuk membuat repository Git baru di folder kerja kamu. Git akan membuat folder tersembunyi bernama `.git` yang berisi semua data versi dan riwayat perubahan.

5.2 GIT Add

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git add .
```

Perintah tersebut adalah untuk menambahkan file ke staging area, yaitu tempat sementara sebelum disimpan ke riwayat versi.

5.3 GIT Commit

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git commit -m "finish post test"
[main ff53e17] finish post test
2 files changed, 175 insertions(+), 144 deletions(-)
```

Perintah tersebut adalah untuk menyimpan perubahan permanen ke dalam repository lokal dengan pesan tertentu.

5.4 GIT Remote

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git remote add origin https://github.com/geogilang14-12/praktikum-apd
error: remote origin already exists.
```

Perintah tersebut adalah untuk menghubungkan repository lokal dengan repository online (remote) seperti GitHub.

5.5 GIT Push

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 157.54 KiB | 5.83 MiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
remote: This repository moved. Please use the new location:
remote:  https://github.com/geogilang14-12/praktikum-apd.git
To https://github.com/geogilang14-12/pertemuan-1.git
   0f9e095..ff53e17  main -> main
branch 'main' set up to track 'origin/main'.
```

Perintah tersebut adalah untuk mengirim (upload) commit dari repository lokal ke repository remote (GitHub).