

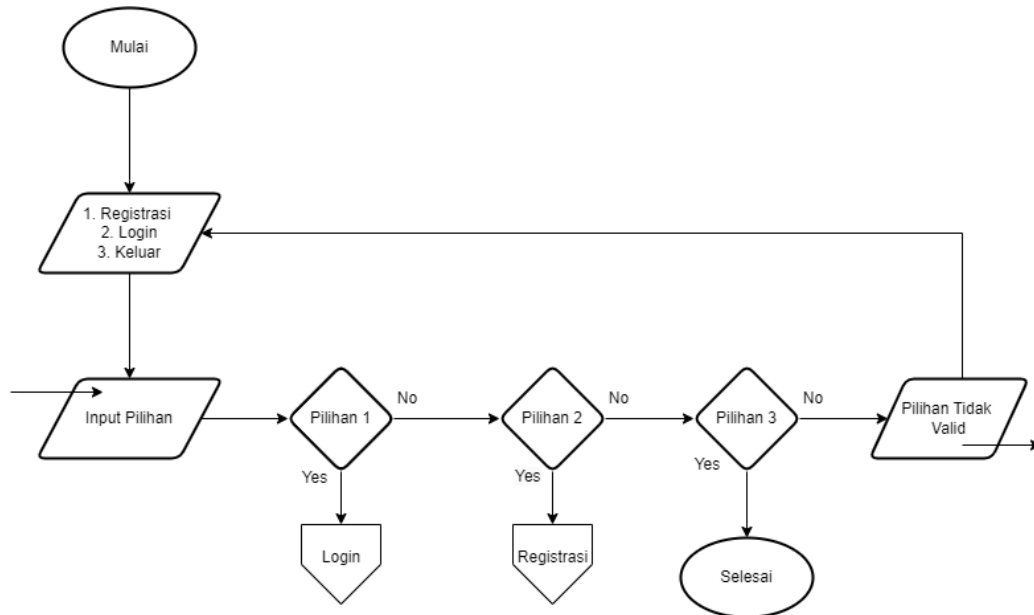
LAPORAN PRAKTIKUM
POSTTEST 7
ALGORITMA PEMROGRAMAN DASAR



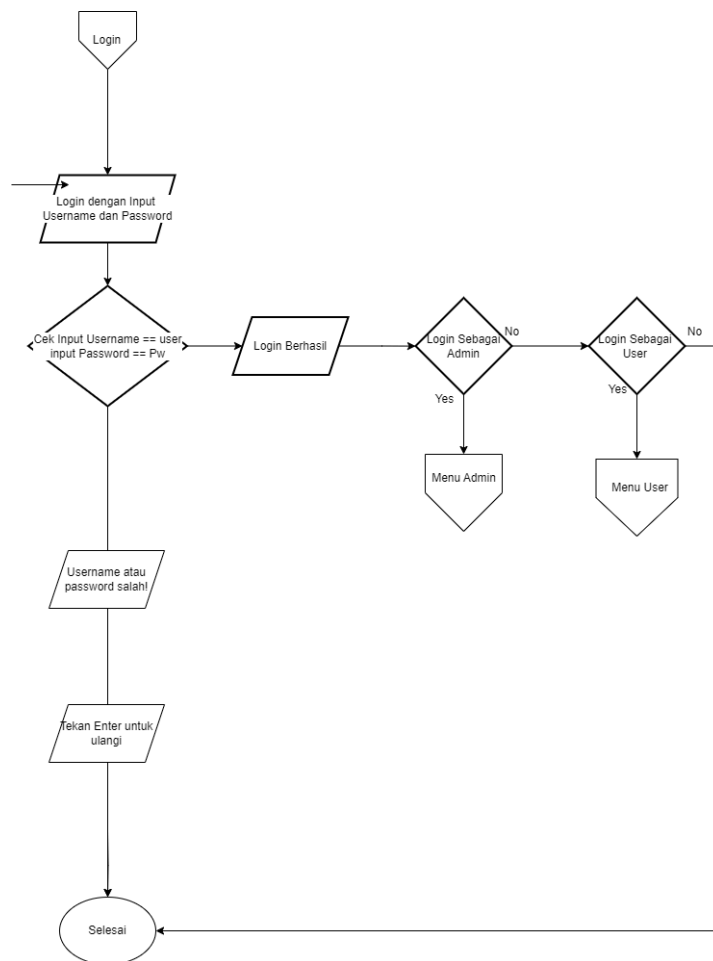
Disusun oleh:
Muhammad Geo Gilang Saputra
Kelas C2'25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

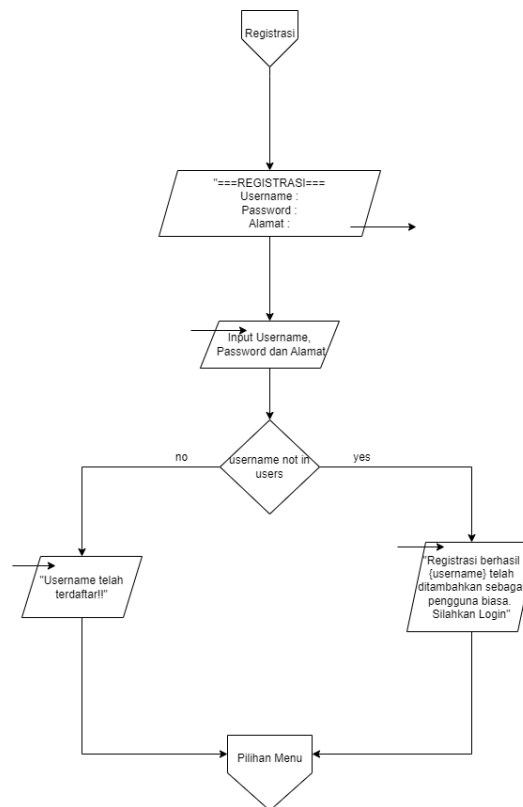
1. Flowchart



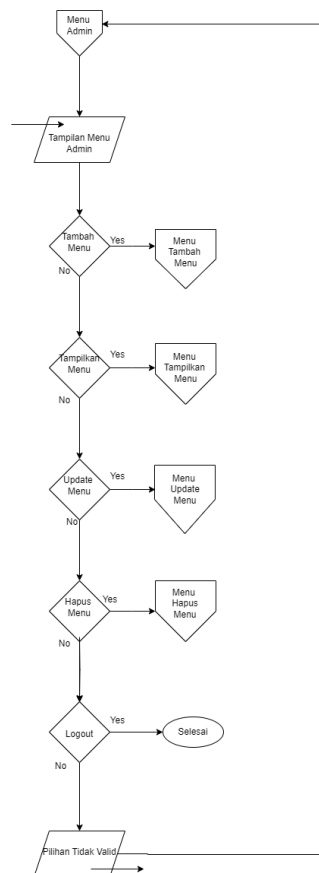
Gambar 1.1 Flowchart Halaman Awal



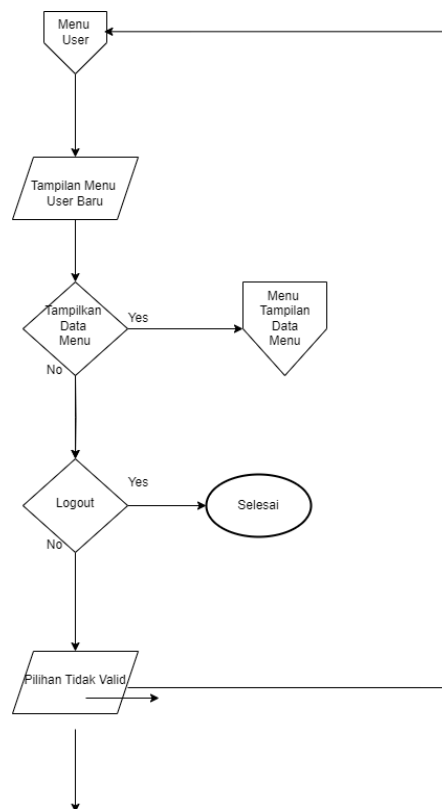
Gambar 1.2 Flowchart Halaman Login



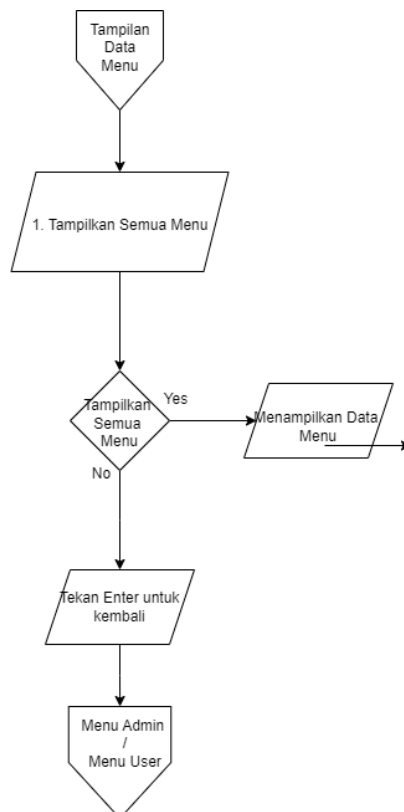
Gambar 1.3 Flowchart Halaman Registrasi



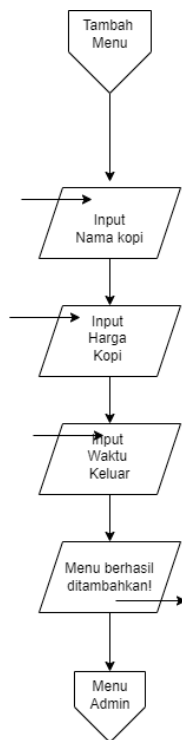
Gambar 1.4 Flowchart Halaman Menu Admin



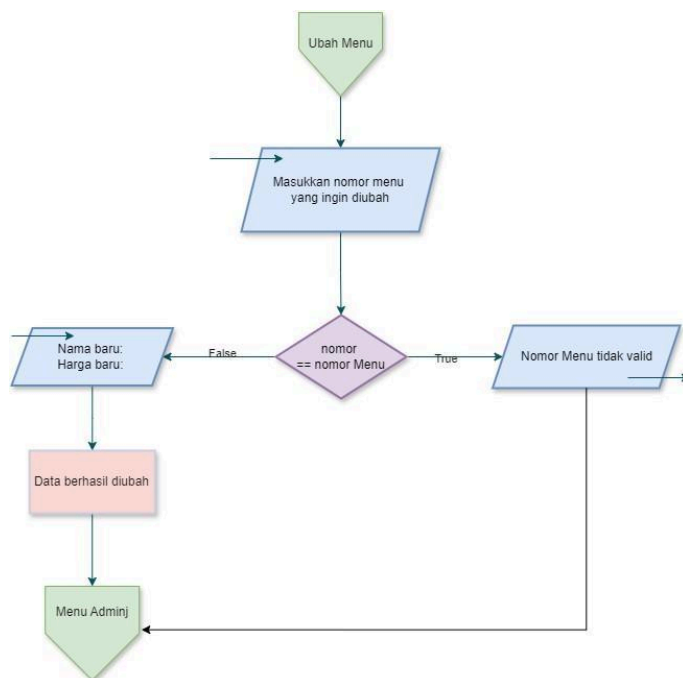
Gambar 1.5 Flowchart Halaman Menu User



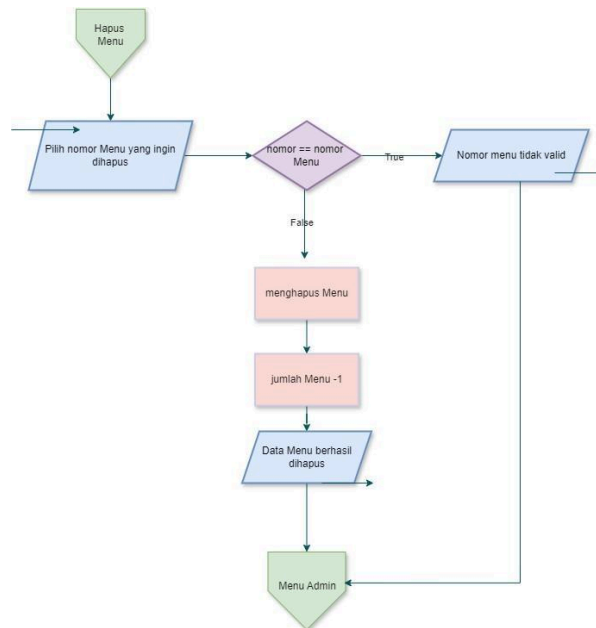
Gambar 1.6 Flowchart Halaman Tampilkan Menu



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 lanjutan Posttest 6 ini merupakan sistem manajemen **Coffee Shop** yang dikembangkan menggunakan konsep **fungsi, prosedur, variabel global dan lokal, error handling, serta fungsi rekursif** sesuai dengan modul pembelajaran. Program ini memiliki dua jenis pengguna, yaitu **admin** dan **user**, yang memiliki hak akses berbeda. Pada awal program, pengguna dapat memilih untuk **login, register**, atau **keluar**. Saat proses registrasi, sistem menerapkan *error handling* untuk memastikan username tidak kosong, tidak duplikat, dan password memiliki panjang minimal yang sesuai.

Setelah login, **admin** dapat melakukan berbagai pengelolaan menu seperti melihat daftar kopi, menambah menu baru melalui fungsi dengan parameter (Tambah menu (nama, harga)), menghapus menu dengan pengecekan kesalahan input, serta melihat total transaksi yang dicatat dalam variabel global `transaksi_total`. Sebaliknya, **user** hanya dapat melihat daftar menu dan melakukan pemesanan kopi. Proses pemesanan menggunakan **fungsi rekursif** (`Pesan_kopi(jumlah)`), yang akan memanggil dirinya sendiri hingga jumlah pesanan selesai, serta menambahkan hasil total pembelian ke dalam total transaksi global.

Selain itu, program ini juga menerapkan **dua prosedur**, yaitu `clear ()` untuk membersihkan layar dan header (teks) untuk menampilkan judul bagian, serta beberapa **fungsi tanpa parameter** seperti `tampilkan_menu()` dan `hitung_total ()`. Penggunaan **try-except-finally** memastikan program tetap berjalan walau terjadi kesalahan input, seperti

memasukkan harga bukan angka atau memilih menu yang tidak tersedia. Dengan demikian, program ini tidak hanya berfungsi sebagai sistem kasir sederhana, tetapi juga menunjukkan penerapan konsep dasar pemrograman terstruktur yang baik dan aman dari error.

3. Source Code

A. Variabel Global

```
users = {  
    "admin": {"password": "111", "role": "admin"},  
    "user": {"password": "123", "role": "user"}  
}  
  
menu_coffee = {  
    1: {"nama": "Americano", "harga": 15000},  
    2: {"nama": "Latte", "harga": 20000},  
    3: {"nama": "Cappuccino", "harga": 18000}  
}  
  
transaksi_total = 0 # variabel global tambahan
```

B. Prosedur dan Fungsi Tanpa Parameter

```
def clear():  
    os.system('cls' if os.name == 'nt' else 'clear')  
  
def header(teks):  
    print("=" * 40)  
    print(teks.center(40))  
    print("=" * 40)  
  
def tampilkan_menu():  
    header("DAFTAR MENU COFFEE SHOP")  
    if not menu_coffee:  
        print("Belum ada menu tersedia.")  
    else:  
        for i, data in menu_coffee.items():  
            print(f"{i}. {data['nama']} - Rp{data['harga']}")
```

C. Fungsi Dengan Parameter dan Error Handling

```
def tambah_menu(nama, harga):  
    try:  
        if not nama.strip():  
            raise ValueError("Nama menu tidak boleh kosong!")  
        if harga <= 0:  
            raise ValueError("Harga harus Lebih besar dari 0!")  
        id_baru = max(menu_coffee.keys()) + 1  
        menu_coffee[id_baru] = {"nama": nama, "harga": harga}  
        print("✅ Menu berhasil ditambahkan!")  
    except ValueError as e:  
        print(f"⚠️ Error: {e}")
```


D. Fungsi Rekursif (Pesan Kopi)

```
def pesan_kopi(jumlah):  
    if jumlah == 0:  
        return 0  
    else:  
        tampilkan_menu()  
        try:  
            pilih = int(input("Masukkan nomor kopi yang ingin dipesan: "))  
            if pilih not in menu_coffee:  
                raise KeyError("Menu tidak ditemukan!")  
            harga = menu_coffee[pilih]["harga"]  
            print(f"Pesanan ditambahkan: {menu_coffee[pilih]['nama']} -  
Rp{harga}")  
            return harga + pesan_kopi(jumlah - 1)  
        except (ValueError, KeyError) as e:  
            print(f"⚠ {e}")  
            return pesan_kopi(jumlah - 1)  
        finally:  
            print("Transaksi sebagian selesai.\n")
```

E. Contoh Penerapan Error Handling Saat Register

```
try:  
    if not username.strip():  
        raise ValueError("Username tidak boleh kosong!")  
    if len(password) < 3:  
        raise ValueError("Password minimal 3 karakter!")  
    if username in users:  
        raise KeyError("Username sudah terdaftar!")  
    users[username] = {"password": password, "role": "user"}  
    print("\n✅ Akun berhasil dibuat!")  
except (ValueError, KeyError) as e:  
    print(f"⚠ {e}")  
finally:  
    input("\nTekan Enter untuk kembali...")
```

F. Pemanggilan Fungsi di Menu Admin

```
elif pilih_admin == "2":  
    nama = input("Nama kopi: ")  
    try:  
        harga = int(input("Harga: "))  
        tambah_menu(nama, harga)  
    except ValueError:  
        print("Harga harus berupa angka!")
```

4. Hasil Output

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

Gambar 4.1 Hasil Output

```
=====
          MENU ADMIN COFFEE SHOP
=====
1. Lihat Menu
2. Tambah Menu
3. Hapus Menu
4. Lihat Total Transaksi
5. Logout
Pilih menu: █
```

Tampilan Menu Utama

```
=====
          DAFTAR MENU COFFEE SHOP
=====
1. Americano - Rp15000
2. Latte - Rp20000
3. Cappuccino - Rp18000
=====
Tekan Enter untuk kembali...█
```

Gambar 4.2 Hasil Output
Tampilan Menu Admin

```
=====
          TAMBAH MENU BARU
=====
Nama kopi: carmel
Harga: 25000
✅ Menu berhasil ditambahkan!
Tekan Enter untuk kembali...█
```

Gambar 4.3 Hasil Output
Tampilan Menu Admin Lihat
Menu

Gambar 4.4 Hasil Output
Tampilan Menu Admin Tambah
Menu

```
=====
                        UBAH MENU COFFEE SHOP
=====
                        DAFTAR MENU COFFEE SHOP
=====
1. Americano - Rp15000
2. Latte - Rp20000
3. Cappuccino - Rp18000
=====
Masukkan nomor menu yang ingin diubah: █
```

Gambar 4.5 Hasil Output Tampilan Menu Admin Ubah Menu

```
=====
                        DAFTAR MENU COFFEE SHOP
=====
1. Americano - Rp15000
2. Latte - Rp20000
3. Cappuccino - Rp18000
=====
Masukkan nomor menu yang ingin dihapus: █
```

Gambar 4.6 Hasil Output Tampilan Menu Admin Hapus Menu

```
=====
                        REGISTER AKUN BARU
=====
Masukkan username baru: geo
Masukkan password: 1467
✓ Akun berhasil dibuat!
Tekan Enter untuk kembali...█
```

Gambar 4.7 Hasil Output Tampilan Menu Registrasi

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

Gambar 4.8 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\Post-Test> 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\Post-Test> git commit -m "Finish Post Test"
[main cbcd958] Finish Post Test
2 files changed, 171 insertions(+)
create mode 100644 Post-Test/post-test-apd-6/2509106121-Muhammad Geo Gilang Saputra-PT-6.pdf
create mode 100644 Post-Test/post-test-apd-6/2509106121-Muhammad Geo Gilang Saputra-PT-6.py
```

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\Post-Test> 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\Post-Test> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 648.12 KiB | 16.20 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
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
72012a4..cbcd958 main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd\Post-Test> █
```

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

