

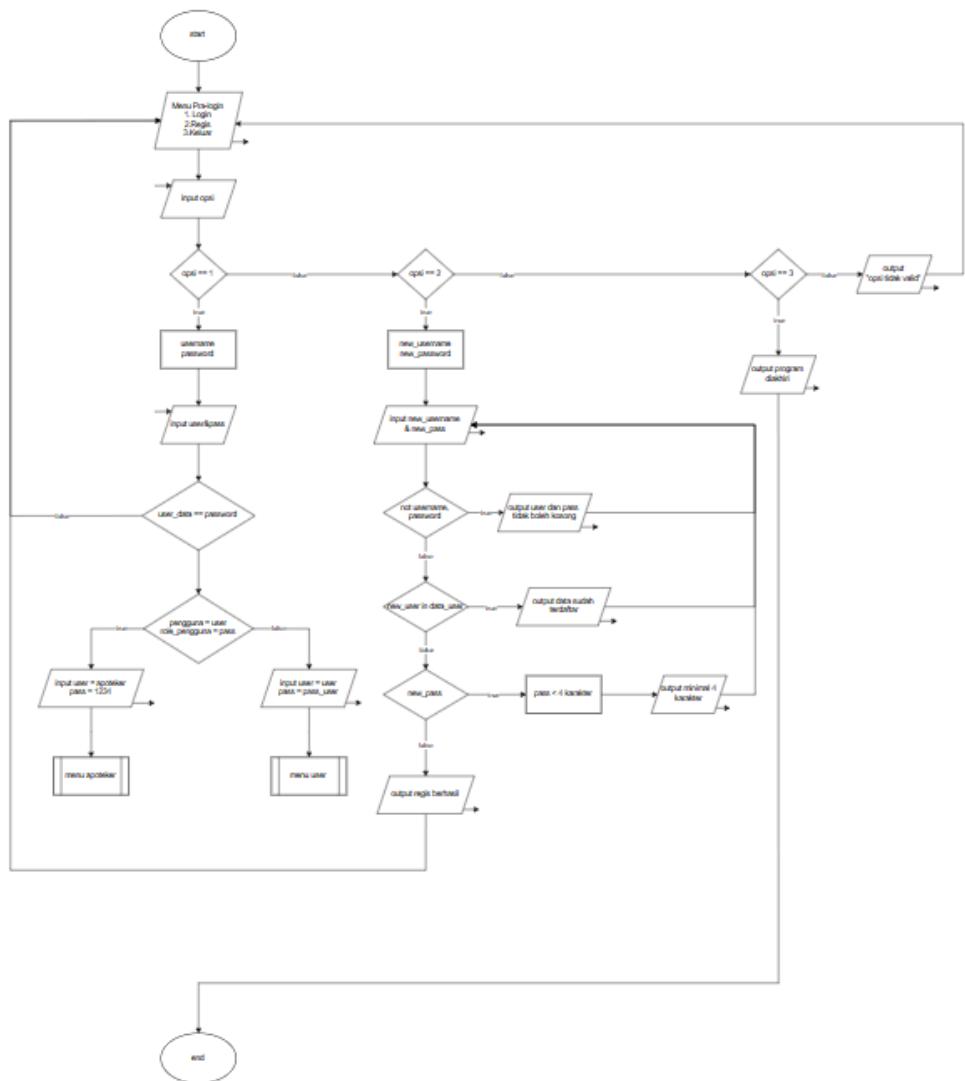
**LAPORAN PRAKTIKUM
POSTTEST (8)
ALGORITMA PEMROGRAMAN DASAR**



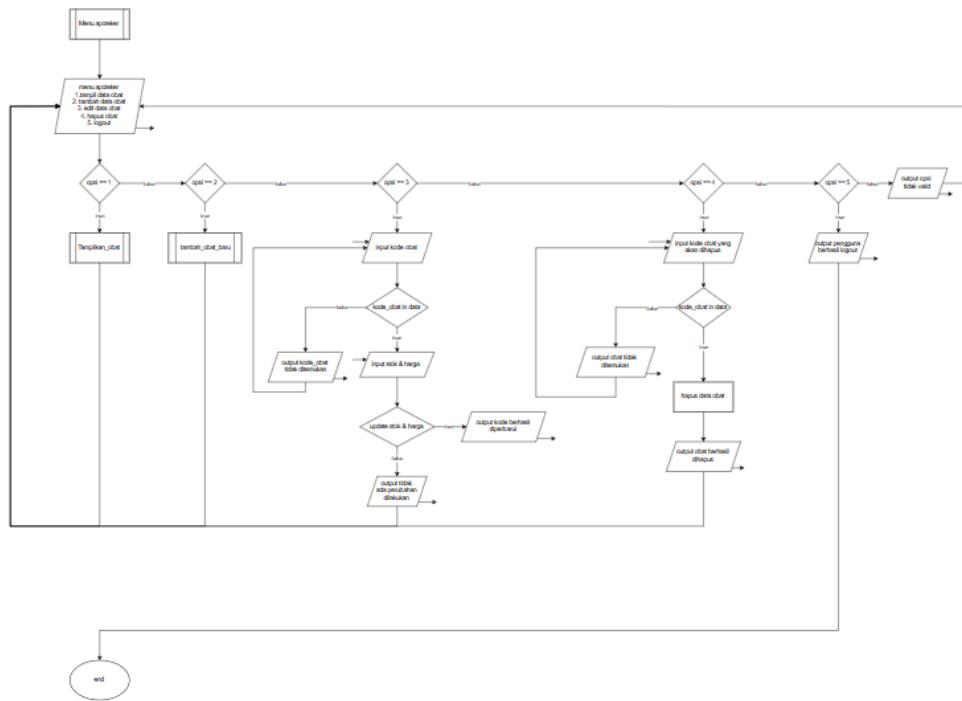
Disusun oleh:
Nayla Nur Elsyada (2509106070)
Kelas (B2'25)

**PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025**

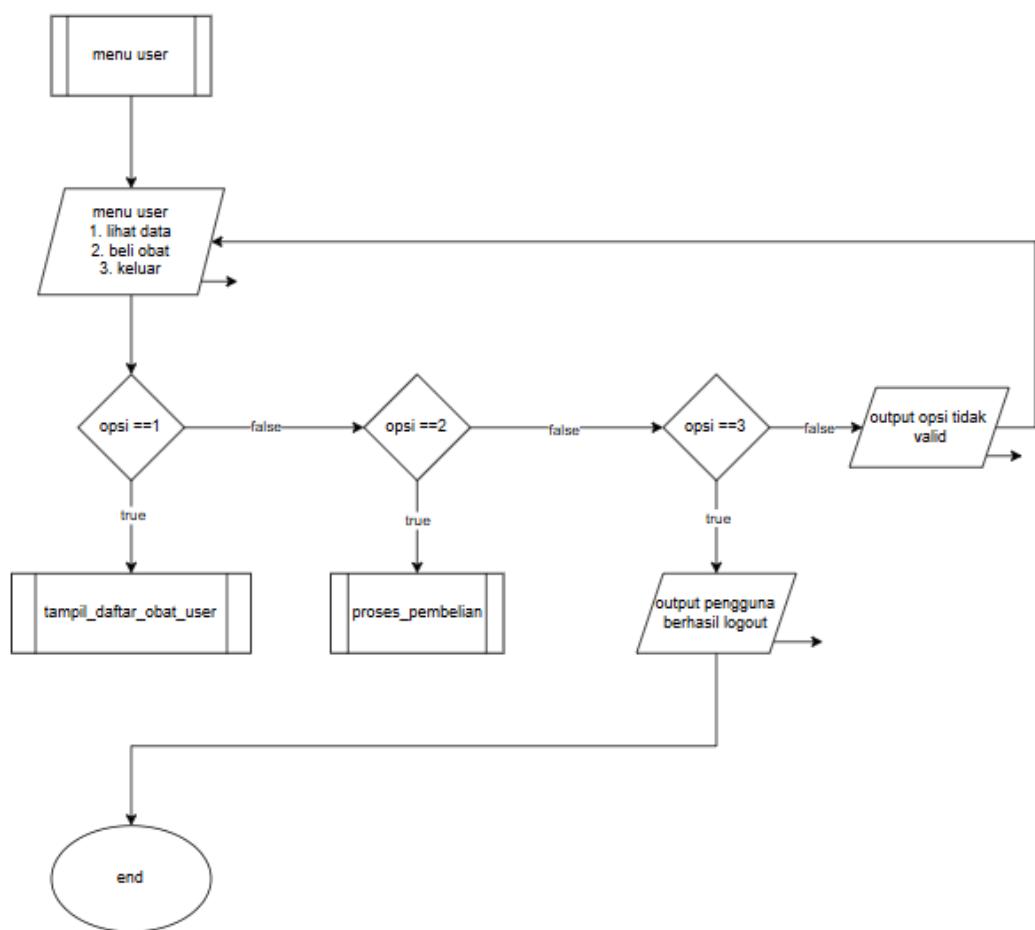
1. Flowchart



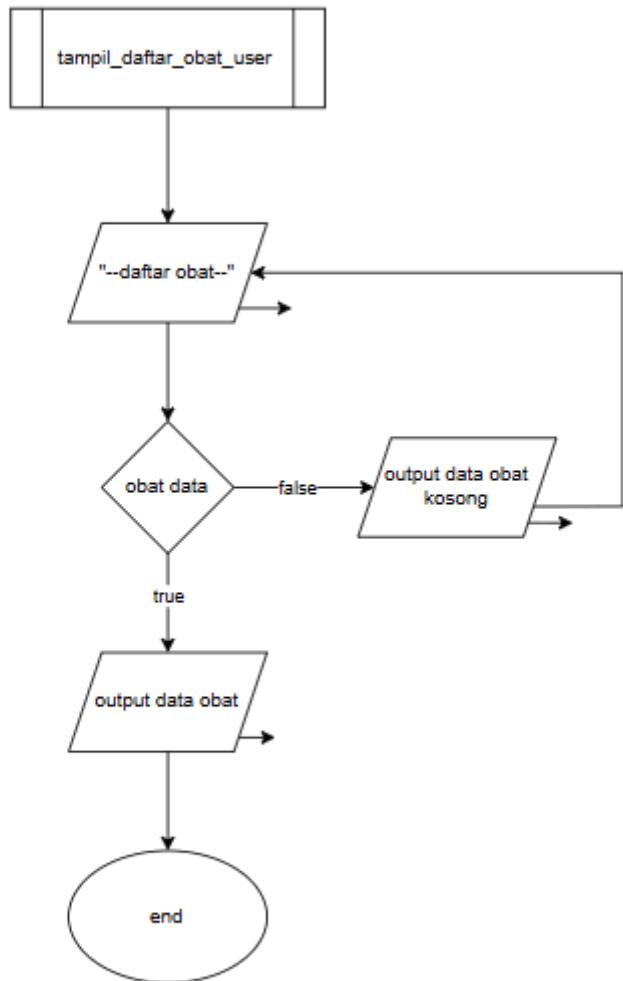
Gambar 1.1 Flowchart Pra-Login



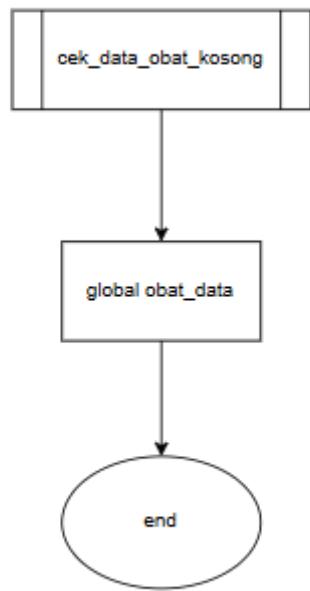
Gambar 1.2 Flowchart Menu Apoteker



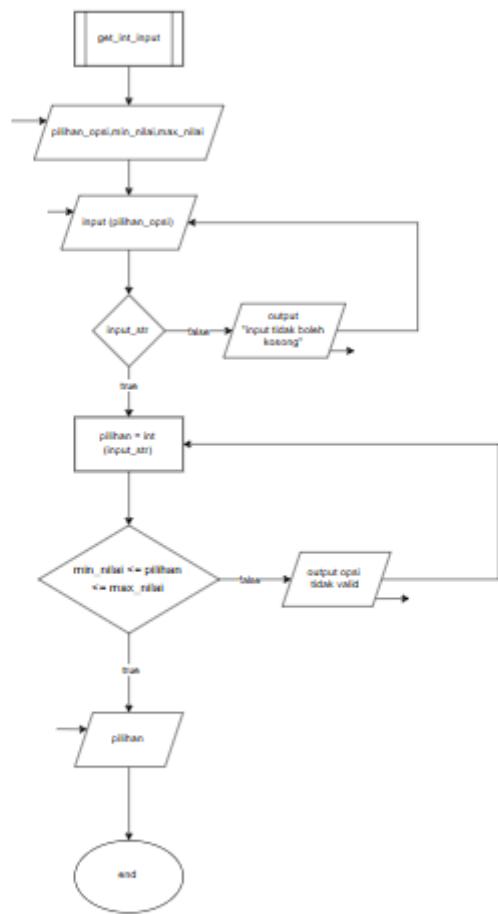
Gambar 1.3 Flowchart Menu user



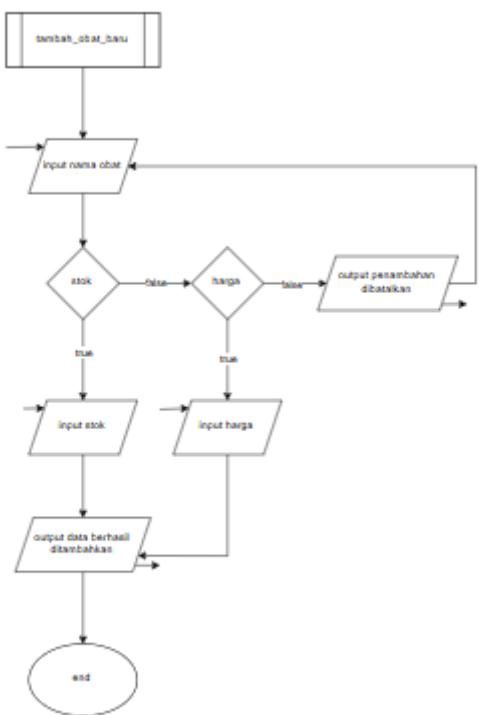
Gambar 1.4 Flowchart Fungsi Tampil Data (User)



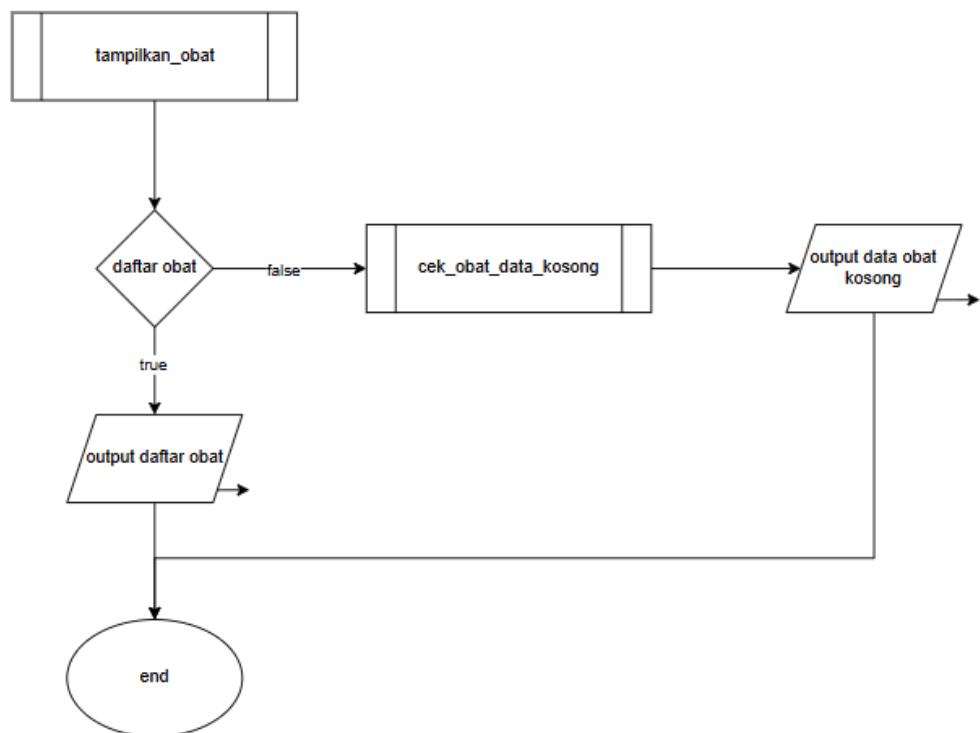
Gambar 1.5 Flowchart Cek Data



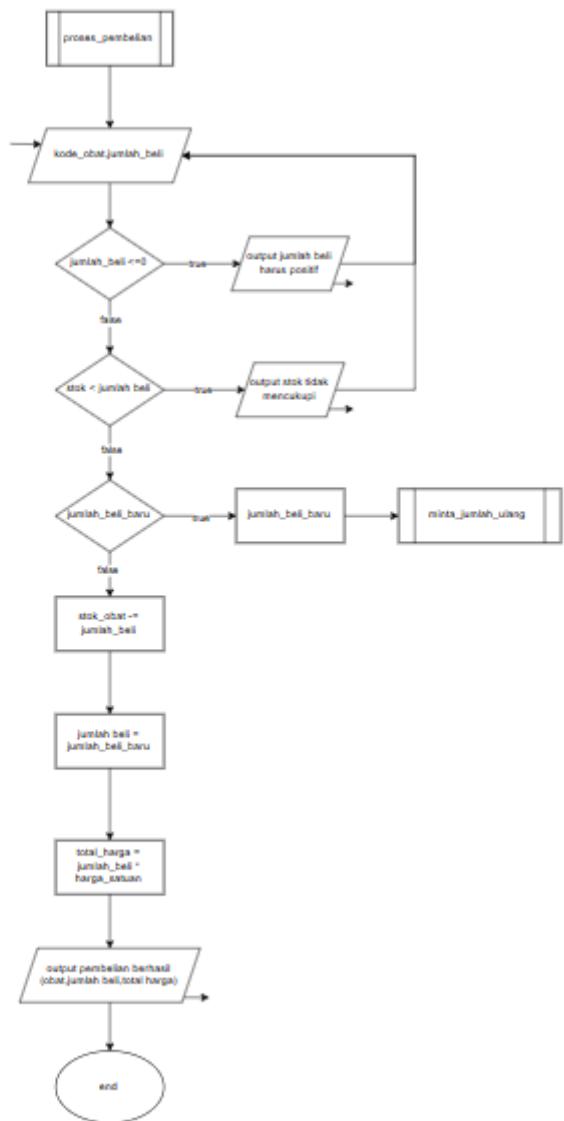
Gambar 1.6 Flowchart Input



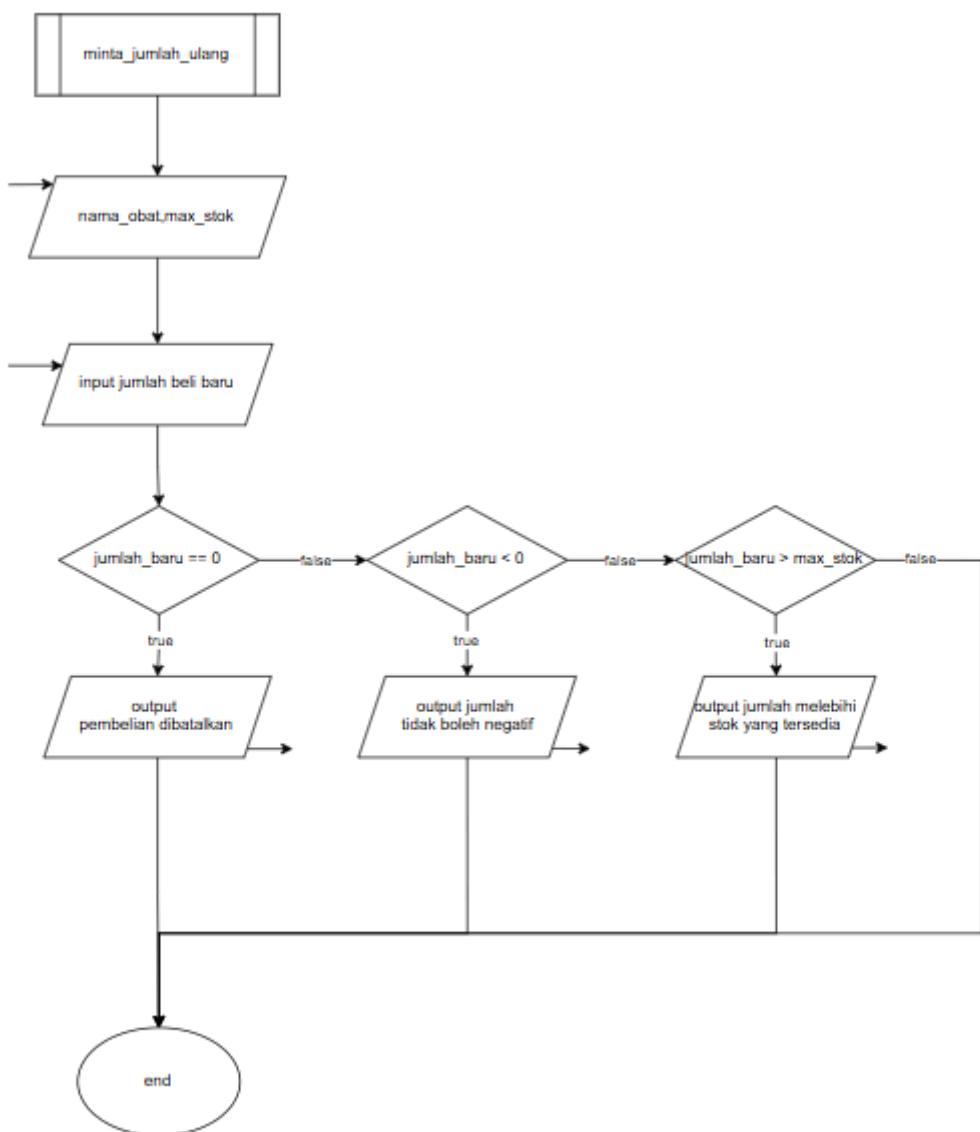
Gambar 1.7 Flowchart Tambah Obat



Gambar 1.8 Flowchart Tampilkan Obat



Gambar 1.9 Flowchart Proses Pembelian



Gambar 1.10 Flowchart Minta Jumlah Ulang

Penjelasan :

- Data Awal & Variabel
 - user_data (Menyimpan data login user beserta password)
 - obat_data (Menyimpan daftar obat berdasarkan kode obat)
 - next_kode_obat (Variabel guna memberikan kode unik secara otomatis saat obat baru ditambahkan.)

- pengguna, role_pengguna, run (Variabel untuk mencari status pengguna yang sedang aktif dan mengontrol loop utama program)
- Fungsi Utama
 - get_int_input() : fungsi validasi input, guna meminta input dari pengguna dan memastikan input tersebut adalah interger yang berada dalam rentang minimum dan maksimum yang ditentukan (misal pilih opsi 1-5).
 - tampilkan_daftar_obat_user() : guna menampilkan daftar lengkap obat, harga, dan stok yang tersedia kepada pengguna umum.
 - proses_pembelian() : guna mengelola transaksi pembelian, dengan melakukan validasi kode obat dan jumlah beli, jika stok kurang, ia memanggil fungsi rekursif (minta_jumlah_ulang()) untuk meminta input baru, jika sukses, stok dikurangi dan total harga ditampilkan.
 - cek_obat_kosong() : guna memeriksa (obat_data) tidak memiliki isi.
 - prosedur_tampilkan_obat_admin() : guna menampilkan daftar obat
 - prosedur_tambah_obat_baru() : guna menambahkan obat baru dalam (obat_data) setelah memvalidasi nama, stok, dan harga.
 - prettytable() : guna merapikan tampilan data

2. Deskripsi Singkat Program

Aplikasi manajemen inventaris dan penjualan untuk apotek atau toko obat. Menyediakan fungsionalitas dasar untuk mengelola daftar obat, memonitor stok, dan mencatat penjualan harian.

3. Source Code

3.1 Fitur Deklarasi Data Global (Modul Data)

```
# -- DATA GLOBAL --
user_data = {
    "apoteker" : {"password": "1234", "role": "apoteker"},
    "user": {"password": "2345", "role": "user"}
}
obat_data = {
    101: {"nama": "Paracetamol", "stok": 50, "harga": 5000},
    102: {"nama": "Amoxicillin", "stok": 30, "harga": 15000},
```

```

103: {"nama": "Komix", "stok": 20, "harga": 10000},
104: {"nama": "OBH", "stok": 15, "harga": 13000},
105: {"nama": "Tremenza", "stok": 20, "harga": 20000},
106: {"nama": "Omeprazol", "stok": 13, "harga": 15000}
}
next_kode_obat = 107
pengguna = None
role_pengguna = None
run = True
from prettytable import PrettyTable
import data as d
import handler as h
import os
tabel1 = PrettyTable()
tabel2 = PrettyTable()
tabel3 = PrettyTable()

```

3.2 Fitur Fungsi Input Interger

```

from prettytable import PrettyTable
import data as d

def get_int_input(pilihan_opsi, min_nilai, max_nilai):
    while True:
        input_str = input(pilihan_opsi).strip()
        if not input_str:
            print("Input tidak boleh kosong.")
            continue
        try:
            pilihan = int(input_str)
            if min_nilai <= pilihan <= max_nilai:
                return pilihan
            else:
                print(f"Opsi tidak valid. Pilih antara {min_nilai} dan {max_nilai}.")
        except ValueError:
            print("Input harus berupa angka.")

```

3.3 Fitur Fungsi Tampil Data

```

def tampilkan_daftar_obat_user():
    print("\n--- Daftar Obat ---\n")
    if cek_obat_kosong():
        print("Data obat kosong.")
        return False

```

```

table = PrettyTable()
table.field_names = ["Kode", "Nama Obat", "Harga (Rp)", "Stok"]
table.align = "l"

for kodeObat, obat in d.obat_data.items():
    harga_formatted = f"{obat['harga']:,}"
    table.add_row([kodeObat, obat['nama'], harga_formatted,
    obat['stok']])

print(table)
return True

```

3.4 Fitur Fungsi Proses Pembelian

```

def proses_pembelian(kode_obat, jumlah_beli):
    if kode_obat not in d.obat_data:
        print("Kode obat tidak valid.")
        return False

    data_obat = d.obat_data[kode_obat]
    nama_obat = data_obat['nama']
    harga_satuan = data_obat['harga']

    if jumlah_beli <= 0:
        print("Jumlah beli harus positif.")
        return False

    if data_obat['stok'] < jumlah_beli:
        print(f"Stok {nama_obat} tidak mencukupi (Tersedia: {data_obat['stok']}).")

    def minta_jumlah_ulang(nama_obat, max_stok):
        print(f"Mohon masukkan ulang jumlah pembelian untuk {nama_obat}. Maksimal {max_stok}.")
        try:
            jumlah_baru_str = input("Jumlah Beli Baru (0 untuk batal):").strip()
            if not jumlah_baru_str:
                return minta_jumlah_ulang(nama_obat, max_stok)
            jumlah_baru = int(jumlah_baru_str)
            if jumlah_baru == 0:
                print("Pembelian dibatalkan.")
                return 0
            elif jumlah_baru < 0:

```

```

        print("Jumlah tidak boleh negatif.")
        return minta_jumlah_ulang(nama_obat, max_stok)
    elif jumlah_baru > max_stok:
        print(f"Jumlah melebihi stok tersedia ({max_stok}).")
        return minta_jumlah_ulang(nama_obat, max_stok)
    else:
        return jumlah_baru
except ValueError:
    print("Input harus berupa angka.")
    return minta_jumlah_ulang(nama_obat, max_stok)

jumlah_beli_baru = minta_jumlah_ulang(nama_obat, data_obat['stok'])
if jumlah_beli_baru == 0:
    return False
jumlah_beli = jumlah_beli_baru

# Lakukan Transaksi
data_obat['stok'] -= jumlah_beli
total_harga = jumlah_beli * harga_satuan

table_transaksi = PrettyTable()
table_transaksi.field_names = ["Keterangan", "Nilai"]
table_transaksi.align = "l"
table_transaksi.add_row(["Obat", nama_obat])
table_transaksi.add_row(["Jumlah", jumlah_beli])
table_transaksi.add_row(["Total Harga (Rp)", f"{total_harga:,}"])

print("\nPembelian berhasil:")
print(table_transaksi)
return True

```

3.5 Fitur Fungsi Menu Obat

```

# Cek ketersediaan data obat
def cek_obat_kosong():
    return not d.obat_data

# Cek data obat
def prosedur_tampilkan_obat_admin():
    if cek_obat_kosong():
        print("Data obat kosong.")
        return

    print("\n--- Daftar Obat (Admin) ---")

    table = PrettyTable()
    table.field_names = ["Kode", "Nama Obat", "Stok", "Harga (Rp)"]

```

```

table.align = "l"

for kodeObat, obat in d.obat_data.items():
    harga_formatted = f"{obat['harga']:,}"
    table.add_row([kodeObat, obat['nama'], obat['stok'],
    harga_formatted])

print(table)

```

3.6 Fitur Fungsi Proses Tambah Data

```

# Proses Tambah Obat Baru
def prosedur_tambah_obat_baru():
    print("\n--- Tambah Obat Baru ---")

    nama = input("Nama Obat: ").strip()
    if not nama:
        print("Nama obat tidak boleh kosong. Penambahan dibatalkan.")
        return

    # Validasi Stok
    while True:
        try:
            stok = int(input("Stok: ").strip())
            if stok >= 0:
                break
            else:
                print("Stok tidak boleh negatif.")
        except ValueError:
            print("Stok harus berupa angka.")

    # Validasi Harga
    while True:
        try:
            harga = int(input("Harga: ").strip())
            if harga >= 0:
                break
            else:
                print("Harga tidak boleh negatif.")
        except ValueError:
            print("Harga harus berupa angka.")

    d.obat_data[d.next_kode_obat] = {"nama": nama, "stok": stok, "harga": harga}
    print(f"Obat '{nama}' berhasil ditambahkan dengan kode")

```

```
{d.next_kode_obat}.")  
    d.next_kode_obat += 1
```

3.7 Fitur Pra-Login

```
while d.run:  
    if d.pengguna is None:  
        # --- MENU PRE-LOGIN ---  
        tabel1 = PrettyTable()  
        tabel1.field_names = ["==== Selamat Datang di Toko Obat Sehat ===="]  
        tabel1.add_row(["1. Login"])  
        tabel1.add_row(["2. Registrasi"])  
        tabel1.add_row(["3. Keluar"])  
        print (tabel1)  
        pilihan = h.get_int_input("Pilih Opsi (1-3): ", 1, 3)  
        clear()
```

3.8 Fitur Login

```
if pilihan == 1:  
    # --- Login ---  
    username = input("Username: ").strip()  
    password = input("Password: ").strip()  
  
    if username in d.user_data:  
        if d.user_data[username]["password"] == password:  
            d.pengguna = username  
            d.role_pengguna = d.user_data[username]["role"]  
            print(f"Login berhasil! Selamat datang, {username}  
({d.role_pengguna}).")  
            continue  
        else:  
            print("Password salah.")  
    else:  
        print("Username tidak ditemukan.")
```

3.9 Fitur Registrasi Pengguna Baru

```

elif pilihan == 2:
    # --- Registrasi ---
    print("\n--- Registrasi Pengguna Baru ---")
    new_username = input("Masukkan Username Baru: ").strip()
    new_password = input("Masukkan Password: ").strip()

    if not new_username or not new_password:
        print("Username dan Password tidak boleh kosong.")
    elif new_username in d.user_data:
        print("Username sudah terdaftar.")
    else:
        if len(new_password) < 4:
            print("Password minimal 4 karakter.")
        else:
            d.user_data[new_username] = {"password": new_password,
"role": "user"}
            print(f"Registrasi {new_username} berhasil. Silakan Login.")

```

3.10 Fitur Logout

```

elif pilihan == 3:
    # --- Keluar Program ---
    d.run = False
    print("Terima kasih, program diakhiri.")

```

3.11 Fitur Login User Apoteker

```

elif d.pengguna is not None:
    if d.role_pengguna == "apoteker":
        # --- MENU ADMIN ---
        tabel2 = PrettyTable()
        tabel2.field_names = ["==== Menu Admin ===="]
        tabel2.add_row(["1. Lihat Daftar Obat"])
        tabel2.add_row(["2. Tambah Obat Baru"])
        tabel2.add_row(["3. Perbarui Stok/Harga Obat"])
        tabel2.add_row(["4. Hapus Obat"])
        tabel2.add_row(["5. Logout"])

```

```
    print (tabel2)
    opsi_menu = h.get_int_input("Pilih Opsi (1-5): ", 1, 5)
```

3.12 Fitur Opsi Menu 1 (Tampil Data)

```
# pilihan Menu Admin
if opsi_menu == 1:
    h.prosedur_tampilkan_obat_admin()
```

3.11 Fitur Opsi Menu 2 (Tambah Data)

```
elif opsi_menu == 2:
    h.prosedur_tambah_obat_baru()
```

3.12 Fitur Opsi Menu 3 (Update Data)

```
elif opsi_menu == 3:
    h.prosedur_perbarui_obat()
```

3.13 Fitur Opsi Menu 4 (Hapus Data)

```
elif opsi_menu == 4:
    h.prosedur_hapus_obat()
```

3.14 Fitur Opsi Menu 5 (Logout)

```
# 5. Logout
elif opsi_menu == 5:
    print(f"Pengguna {d.pengguna} berhasil logout.")
    d.pengguna = None
    d.role_pengguna = None
```

3.15 Fitur User Biasa

```

elif d.role_pengguna == "user":
    # --- MENU USER BIASA ---
    tabel3 = PrettyTable()
    tabel3.field_names = ["== Menu =="]
    tabel3.add_row(["1. Lihat Daftar Obat"])
    tabel3.add_row(["2. Beli Obat"])
    tabel3.add_row(["3. Logout"])
    print(tabel3)
    opsi_menu = h.get_int_input("Pilih Opsi (1-3): ", 1, 3)

```

3.16 Fitur Opsi Menu 1 User (Tampil Data)

```

if opsi_menu == 1:
    h.tampilkan_daftar_obat_user()

```

3.17 Fitur Opsi menu 2 User (Beli Obat)

```

elif opsi_menu == 2:
    # --- BELI OBAT ---
    if h.cek_obat_kosong():
        print("\nTidak ada obat yang tersedia untuk dibeli.")
        continue

    h.tampilkan_daftar_obat_user()
    print("\n--- Proses Pembelian Obat ---")

    kodeObat = None
    while True:
        inputKode = input("Masukkan kode obat yang akan dibeli:
").strip()
        try:
            kodeObat = int(inputKode)
            if kodeObat in d.obat_data:
                break
            else:
                print("Kode obat tidak ditemukan.")
        except ValueError:
            print("Kode obat harus berupa angka.")

    jumlahBeli = None
    while True:
        inputJumlah = input(f"Masukkan jumlah beli
{d.obat_data.get(kodeObat, {}).get('nama', 'Obat')}: ").strip()
        try:

```

```
        jumlahBeli = int(inputJumlah)
        break
    except ValueError:
        print("Jumlah beli harus berupa angka.")

    h.proses_pembelian(kodeObat, jumlahBeli)
```

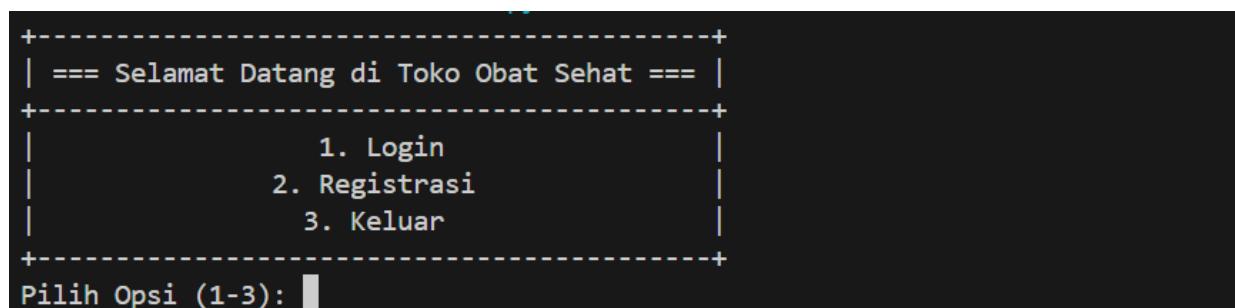
3.18 Fitur Opsi Logout

```
# 3. Logout
elif opsi_menu == 3:
    print(f"Pengguna {d.pengguna} berhasil logout.")
    d.pengguna = None
    d.role_pengguna = None
```

3.19 Fitur Clear

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

4. Hasil Output



A screenshot of a terminal window displaying a Python application's user interface. The window has a dark background with white text. At the top, it says "Selamat Datang di Toko Obat Sehat". Below that is a menu with three options: "1. Login", "2. Registrasi", and "3. Keluar". At the bottom, there is a prompt "Pilih Opsi (1-3):" followed by a cursor.

```
+-----+
| === Selamat Datang di Toko Obat Sehat === |
+-----+
|           1. Login                         |
|           2. Registrasi                     |
|           3. Keluar                        |
+-----+
Pilih Opsi (1-3): |
```

Gambar 4.1 Hasil Output Python

```
Username: apoteker
Password: 1234
Login berhasil! Selamat datang, apoteker (apoteker).
+-----+
|      === Menu Admin ===      |
+-----+
| 1. Lihat Daftar Obat       |
| 2. Tambah Obat Baru        |
| 3. Perbarui Stok/Harga Obat|
| 4. Hapus Obat              |
| 5. Logout                   |
+-----+
Pilih Opsi (1-5): 1

--- Daftar Obat (Admin) ---
+-----+-----+-----+
| Kode | Nama Obat   | Stok | Harga (Rp) |
+-----+-----+-----+
| 101  | Paracetamol | 50   | 5,000    |
| 102  | Amoxicillin | 30   | 15,000   |
| 103  | Komix        | 20   | 10,000   |
| 104  | OBH          | 15   | 13,000   |
| 105  | Tremenza     | 20   | 20,000   |
| 106  | Omeprazol    | 13   | 15,000   |
+-----+-----+-----+
```

Gambar 4.2 Hasil Output Lihat Data

```
Username: apoteker
Password: 1234
Login berhasil! Selamat datang, apoteker (apoteker).
+-----+
|     === Menu Admin === |
+-----+
| 1. Lihat Daftar Obat   |
| 2. Tambah Obat Baru    |
| 3. Perbarui Stok/Harga Obat |
|   4. Hapus Obat          |
|   5. Logout              |
+-----+
Pilih Opsi (1-5): 2

--- Tambah Obat Baru ---
Nama Obat: Vit c
Stok: 20
Harga: 2000
Obat 'Vit c' berhasil ditambahkan dengan kode 107.
```

Gambar 4.3 Hasil Output Tambah Data

```

--- Perbarui Obat ---

--- Daftar Obat (Admin) ---
+-----+-----+-----+
| Kode | Nama Obat | Stok | Harga (Rp) |
+-----+-----+-----+
| 101  | Paracetamol | 50   | 5,000
| 102  | Amoxicillin | 30   | 15,000
| 103  | Komix        | 20   | 10,000
| 104  | OBH          | 15   | 13,000
| 105  | Tremenza     | 20   | 20,000
| 106  | Omeprazol    | 13   | 15,000
| 107  | Vit c         | 20   | 2,000
+-----+-----+-----+
Masukkan kode obat yang akan diubah: 104
Obat yang dipilih: OBH
Masukkan Stok Baru (kosongkan jika tidak diubah): 12
Masukkan Harga Baru (kosongkan jika tidak diubah):
Kode obat 104 berhasil diperbarui. Data terbaru:
+-----+-----+-----+
| Kode | Nama Obat | Stok | Harga (Rp) |
+-----+-----+-----+
| 104  | OBH        | 12   | 13,000
+-----+-----+-----+

```

Gambar 4.4 Hasil Output Update Data Obat

```
--- Hapus Obat ---  
--- Daftar Obat (Admin) ---  
+-----+-----+-----+  
| Kode | Nama Obat | Stok | Harga (Rp) |  
+-----+-----+-----+  
| 101 | Paracetamol | 50 | 5,000 |  
| 102 | Amoxicillin | 30 | 15,000 |  
| 103 | Komix | 20 | 10,000 |  
| 104 | OBH | 12 | 13,000 |  
| 105 | Tremenza | 20 | 20,000 |  
| 106 | Omeprazol | 13 | 15,000 |  
| 107 | Vit c | 20 | 2,000 |  
+-----+-----+-----+  
Masukkan kode obat yang akan dihapus: 103  
Obat 'Komix' (Kode 103) berhasil dihapus.
```

Gambar 4.5 Hasil Output Hapus Obat
(Penerapan Try Except Pada Menu Hapus Data)

```
Terima kasih, program diakhiri.
```

Gambar 4.6 Hasil Output Logout

```
--- Registrasi Pengguna Baru ---  
Masukkan Username Baru: pasien  
Masukkan Password: 0987  
Registrasi pasien berhasil. Silakan Login.
```

Gambar 4.7 Hasil Output Registrasi

```

+-----+
|     === Menu === |
+-----+
| 1. Lihat Daftar Obat |
| 2. Beli Obat          |
| 3. Logout             |
+-----+
Pilih Opsi (1-3): 1

--- Daftar Obat ---

+-----+-----+-----+
| Kode | Nama Obat   | Harga (Rp) | Stok |
+-----+-----+-----+
| 101  | Paracetamol  | 5,000      | 50   |
| 102  | Amoxicillin  | 15,000     | 30   |
| 103  | Komix         | 10,000     | 20   |
| 104  | OBH           | 13,000     | 15   |
| 105  | Tremenza      | 20,000     | 20   |
| 106  | Omeprazol    | 15,000     | 13   |
+-----+-----+-----+

```

Gambar 4.8 Hasil Output Lihat Daftar Obat Menu User

```

Pilih Opsi (1-3): 2

--- Daftar Obat ---

+-----+-----+-----+
| Kode | Nama Obat   | Harga (Rp) | Stok |
+-----+-----+-----+
| 101  | Paracetamol  | 5,000      | 50   |
| 102  | Amoxicillin  | 15,000     | 30   |
| 103  | Komix         | 10,000     | 20   |
| 104  | OBH           | 13,000     | 15   |
| 105  | Tremenza      | 20,000     | 20   |
| 106  | Omeprazol    | 15,000     | 13   |
+-----+-----+-----+

--- Proses Pembelian Obat ---
Masukkan kode obat yang akan dibeli: 103
Masukkan jumlah beli Komix: 3

Pembelian berhasil:
+-----+-----+
| Keterangan | Nilai  |
+-----+-----+
| Obat       | Komix  |
| Jumlah     | 3       |
| Total Harga (Rp) | 30,000 |
+-----+-----+

```

Gambar 4.10 Hasil Output Beli Obat

Terima kasih, program diakhiri.

Gambar 4.11 Hasil Output Logout User

5. Langkah-langkah GIT

5.1 GIT Add

Fungsi `git add` adalah untuk memindahkan perubahan dari direktori kerja (working directory) ke area staging (staging area), yang merupakan langkah awal untuk mempersiapkan berkas agar dimasukkan ke dalam commit berikutnya dalam repositori Git.

```
PS C:\Users\ASUS-GK\Documents\PRAKTIKUM_APD_B2'25> git add .
```

Gambar 5.1 git add

5.2 GIT Commit

Fungsi `git commit` adalah untuk menyimpan "snapshot" atau rekam jejak perubahan yang telah Anda buat pada proyek Anda dan menyimpannya ke dalam riwayat repositori Git. Setiap kali Anda melakukan `commit`, Git akan menyimpan perubahan tersebut beserta informasi tentang siapa yang membuat perubahan, kapan, dan catatan singkat mengenai perubahan tersebut, menciptakan titik penyimpanan yang dapat Anda temukan kembali di masa mendatang.

```
PS C:\Users\ASUS-GK\Documents\PRAKTIKUM_APD_B2'25> git commit -m "2509106070_NAYLANURELSYADA_PT_8"
[main 73441d2] 2509106070_NAYLANURELSYADA_PT_8
 6 files changed, 411 insertions(+)
 create mode 100644 Kelas/pertemuan8_2509106070.py
 create mode 100644 POSTEST/POST-TEST-APD-8/__pycache__/_data.cpython-313.pyc
 create mode 100644 POSTEST/POST-TEST-APD-8/__pycache__/_handler.cpython-313.pyc
 create mode 100644 POSTEST/POST-TEST-APD-8/data.py
 create mode 100644 POSTEST/POST-TEST-APD-8/handler.py
 create mode 100644 POSTEST/POST-TEST-APD-8/main.py
```

Gambar 5.2 git commit

5.3 GIT Push

Fungsi `git push` adalah untuk mengunggah (mendorong) commit atau perubahan yang telah dibuat di repositori lokal Anda ke repositori jarak jauh, seperti repositori di GitHub atau GitLab. Tujuan utamanya adalah membagikan kode Anda kepada tim, menyimpannya sebagai cadangan, dan memungkinkan kolaborasi.

```
PS C:\Users\ASUS-GK\Documents\PRAKTIKUM_APD_B2'25> git push -f origin main
Enumerating objects: 88, done.
Counting objects: 100% (88/88), done.
Delta compression using up to 4 threads
Compressing objects: 100% (85/85), done.
Writing objects: 100% (88/88), 2.26 MiB | 941.00 KiB/s, done.
Total 88 (delta 21), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (21/21), done.
To https://github.com/elsyadaa-del/Praktikum_APD_B2-25.git
 + d287345...73441d2 main -> main (forced update)
PS C:\Users\ASUS-GK\Documents\PRAKTIKUM_APD_B2'25> █
```

Gambar 5.3 git push