

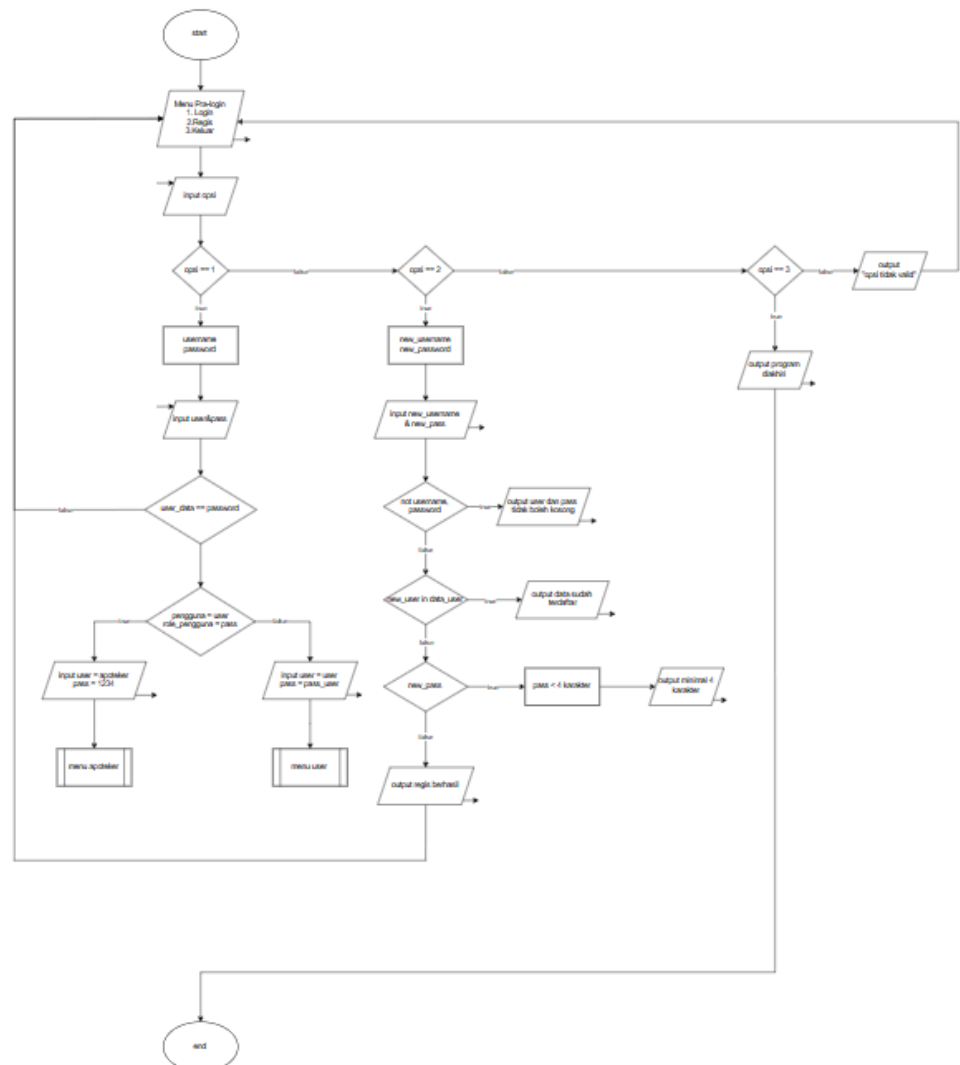
LAPORAN PRAKTIKUM
POSTTEST (7)
ALGORITMA PEMROGRAMAN DASAR



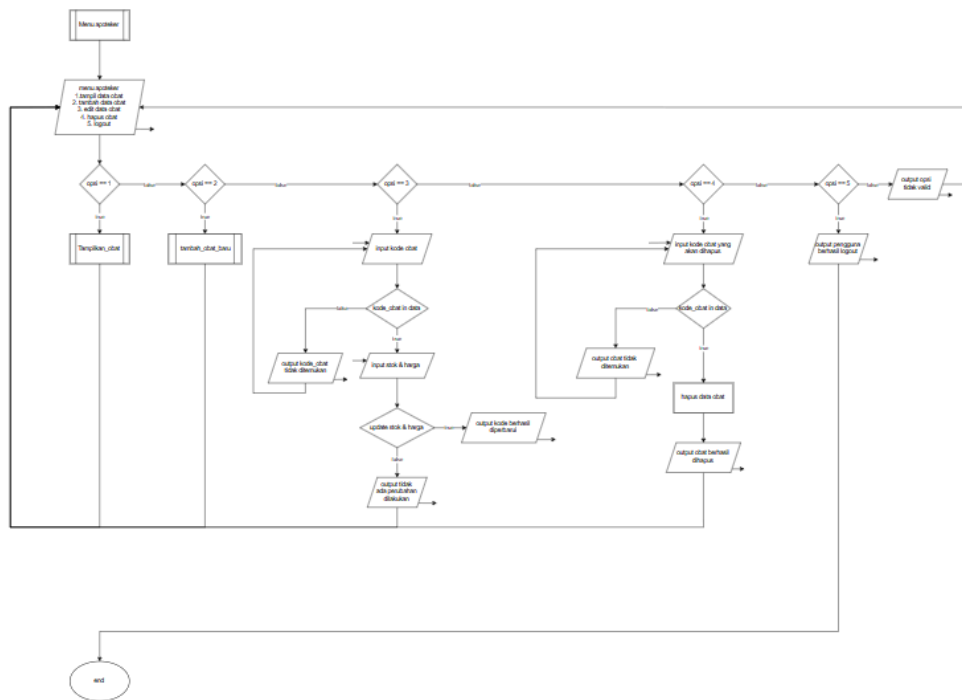
Disusun oleh:
Nayla Nur Elsyada (2509106070)
Kelas (B2'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

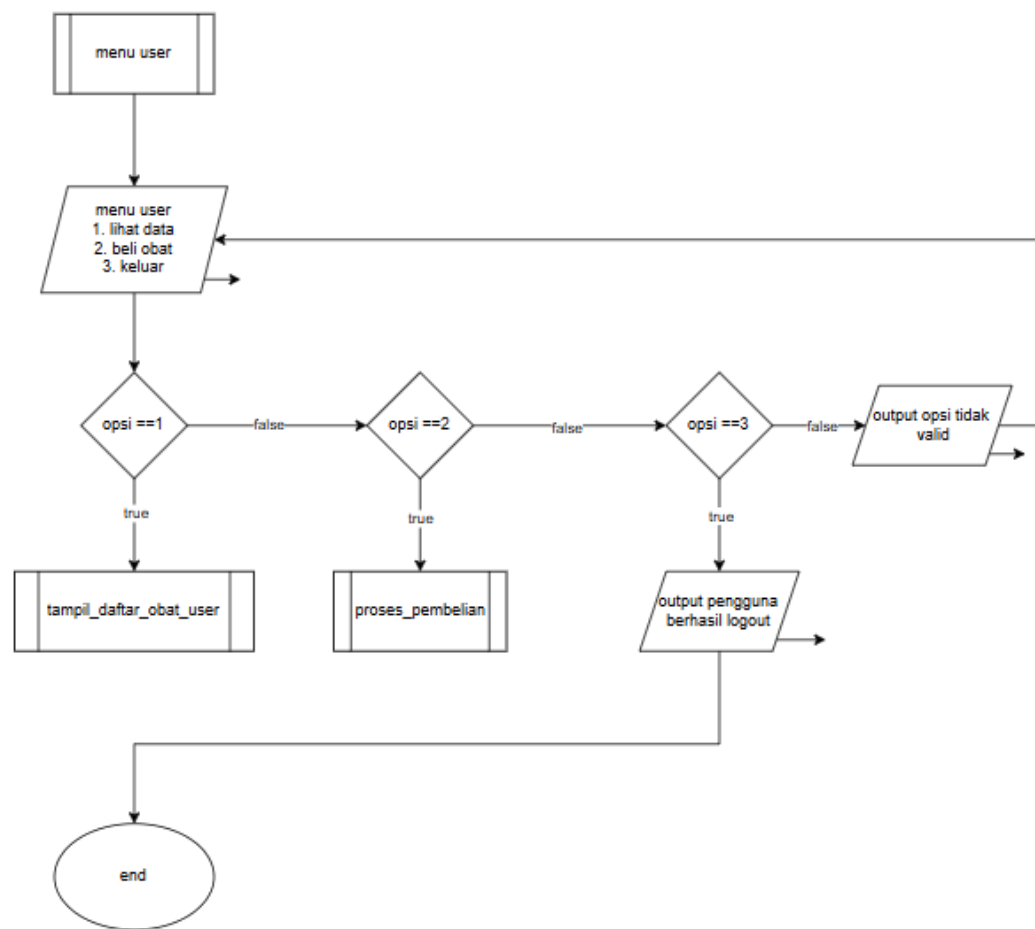
1. Flowchart



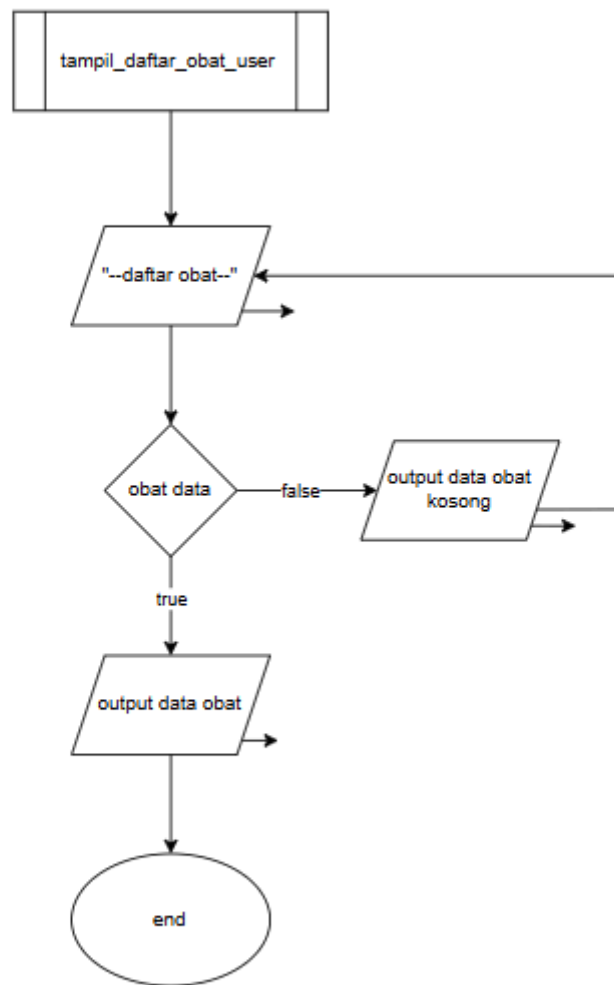
Gambar 1.1 Flowchart Pra-Login



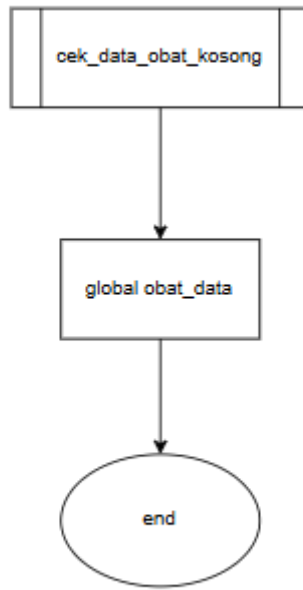
Gambar 1.2 Flowchart Menu Apoteker



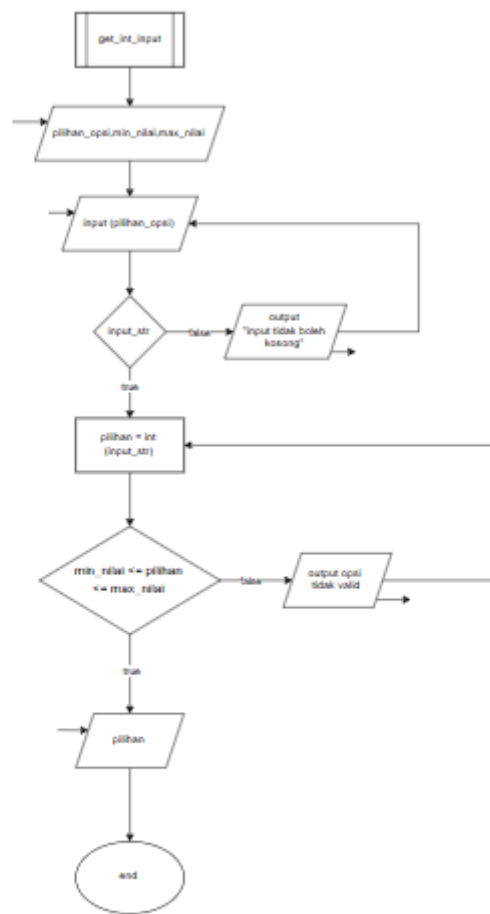
Gambar 1.3 Flowchart Menu user



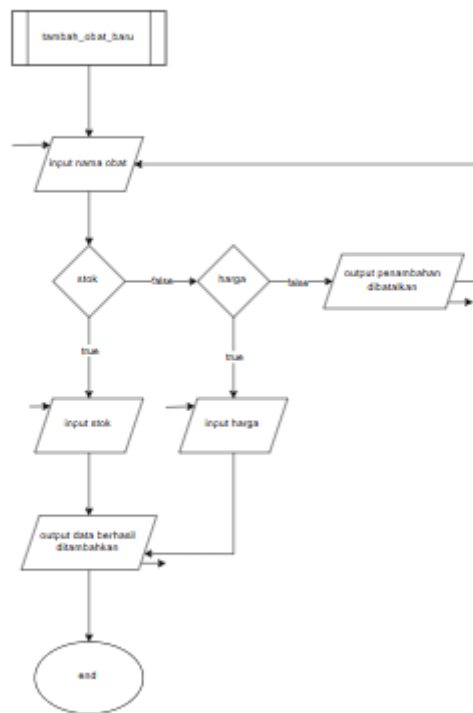
Gambar 1.4 Flowchart Fungsi Tampil Data (User)



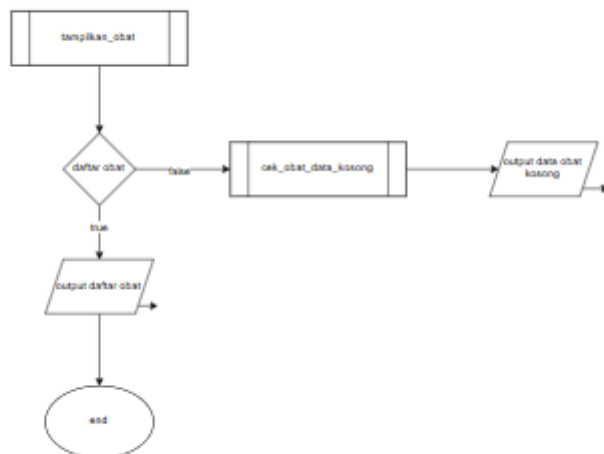
Gambar 1.5 Flowchart Cek Data



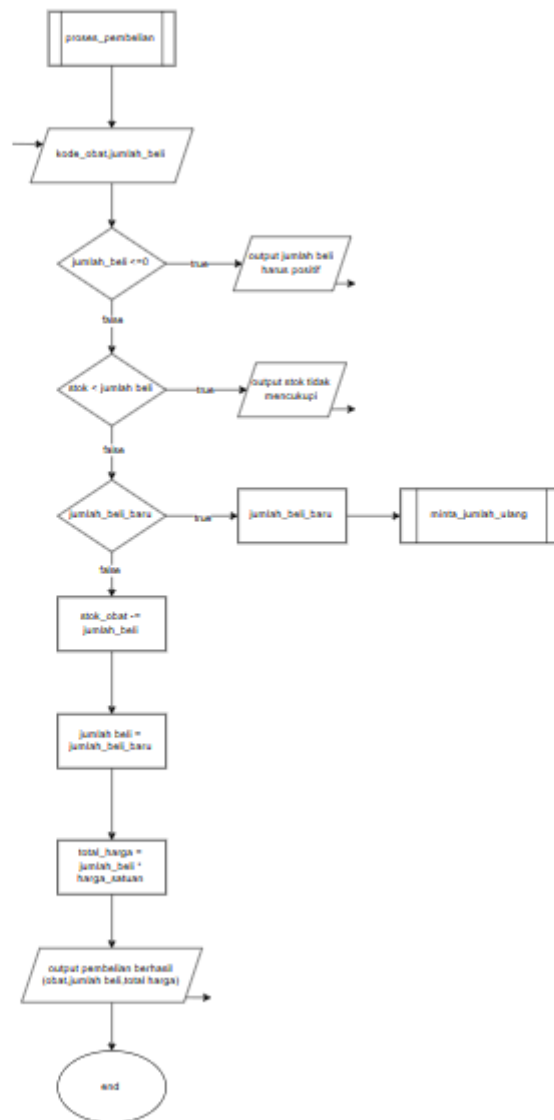
Gambar 1.6 Flowchart Input



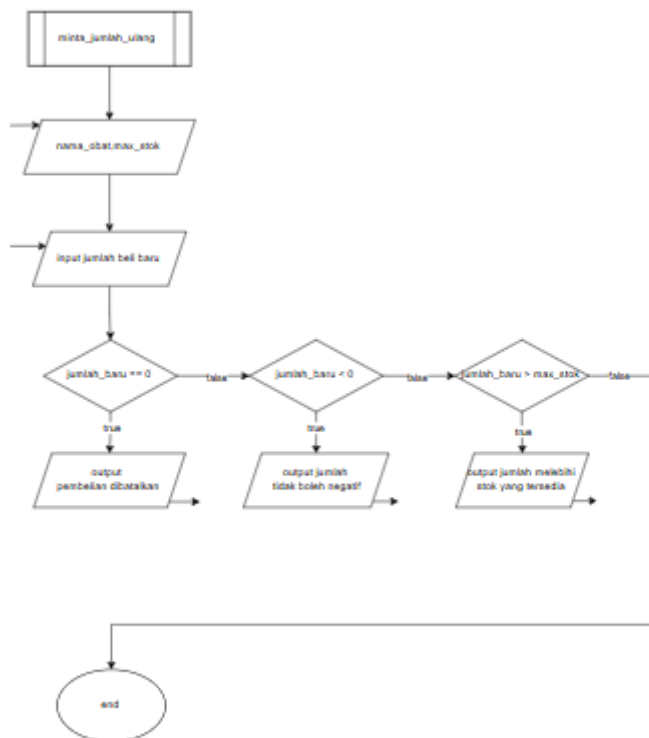
Gambar 1.7 Flowchart Tambah Obat



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.

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

```
# -- 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
```

3.2 Fitur Fungsi Input Interger

```
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():
    output = ""
    print( "\n--- Daftar Obat ---\n")
    if not obat_data:
        output += "Data obat kosong."
    else:
        for kodeObat, data in obat_data.items():
            output += f"Kode: {kodeObat} | Nama: {data['nama']} | Harga: {data['harga']} | Stok: {data['stok']}\n"
    print(output)
    return True
```

3.4 Fitur Fungsi Proses Pembelian

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

    data_obat = 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
print(f"\nPembelian berhasil:")
print(f"- Obat: {nama_obat}")
print(f"- Jumlah: {jumlah_beli}")
print(f"- Total Harga: {total_harga:,}")
return True

```

3.5 Fitur Fungsi Menu Obat

```
# Cek ketersediaan data obat
def cek_obat_kosong():
    global obat_data
    return not obat_data

# Menampilkan Daftar Obat Lengkap
def prosedur_tampilkan_obat_admin():
    if cek_obat_kosong():
        print("Data obat kosong.")
        return

    print("\n--- Daftar Obat (Admin) ---")
    for kodeObat, data in obat_data.items():
        print(f"Kode: {kodeObat} | Nama: {data['nama']} | Stok: {data['stok']} |  

        Harga: {data['harga']:,}")
```

3.6 Fitur Fungsi Proses Tambah Data

```
# Proses Tambah Obat Baru
def prosedur_tambah_obat_baru():
    global next_kode_obat, obat_data
    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 (angka): ").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 (angka): ").strip())
            if harga >= 0:
```

```

        break
    else:
        print("Harga tidak boleh negatif.")
except ValueError:
    print("Harga harus berupa angka.")

obat_data[next_kode_obat] = {"nama": nama, "stok": stok, "harga": harga}
print(f"Obat '{nama}' berhasil ditambahkan dengan kode {next_kode_obat}.")
next_kode_obat += 1

```

3.7 Fitur Pra-Login

```

while run:
    if pengguna is None:
        # --- MENU PRE-LOGIN ---
        print("\n=== Selamat Datang di Toko Obat Sehat ===")
        print("1. Login")
        print("2. Registrasi")
        print("3. Keluar")

        pilihan = get_int_input("Pilih Opsi (1-3): ", 1, 3)

```

3.8 Fitur Login

```

if pilihan == 1:
    # --- Login ---
    username = input("Username: ").strip()
    password = input("Password: ").strip()

    if username in user_data:
        if user_data[username]["password"] == password:
            pengguna = username
            role_pengguna = user_data[username]["role"]
            print(f"Login berhasil! Selamat datang, {username} ({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 user_data:
        print("Username sudah terdaftar.")
    else:
        if len(new_password) < 4:
            print("Password minimal 4 karakter.")
        else:
            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 ---
    run = False
    print("Terima kasih, program diakhiri.")
```

3.11 Fitur Login User Apoteker

```
elif pengguna is not None:
    if role_pengguna == "apoteker":
        # --- MENU ADMIN (Apoteker) ---
        print(f"\n=== Menu Admin ({pengguna}) ===")
        print("1. Lihat Daftar Obat")
        print("2. Tambah Obat Baru")
        print("3. Perbarui Stok/Harga Obat")
        print("4. Hapus Obat")
```



```
print("5. Logout")

opsi_menu = get_int_input("Pilih Opsi (1-5): ", 1, 5)
```

3.12 Fitur Opsi Menu 1 (Tampil Data)

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

3.11 Fitur Opsi Menu 2 (Tambah Data)

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

3.12 Fitur Opsi Menu 3 (Update Data)

```
elif opsi_menu == 3:
    # Memperbarui obat
    print("\n--- Perbarui Obat ---")

    # Input Kode Obat
    kodeObat = None
    while True:
        inputKode = input("Masukkan kode obat yang akan diubah: ")
        inputKode = inputKode.strip()

        try:
            kodeObat = int(inputKode)
            break
        except ValueError:
            print("Kode obat harus berupa angka.")

    if kodeObat in obat_data:
        print(f"Obat yang dipilih: {obat_data[kodeObat]['nama']}")

        stok_baru_str = input("Masukkan Stok Baru (kosongkan jika tidak diubah): ").strip()
        harga_baru_str = input("Masukkan Harga Baru (kosongkan jika tidak diubah): ").strip()

        updated = False
```

```

        # Cek dan update stok
        if stok_baru_str:
            try:
                stokBaru = int(stok_baru_str)
                if stokBaru >= 0:
                    obat_data[kodeObat]['stok'] = stokBaru
                    updated = True
                else:
                    print("Stok tidak boleh negatif. Perubahan
stok dibatalkan.")
            except ValueError:
                print("Stok harus berupa angka. Perubahan stok
dibatalkan.")

        # Cek dan update harga
        if harga_baru_str:
            try:
                hargaBaru = int(harga_baru_str)
                if hargaBaru >= 0:
                    obat_data[kodeObat]['harga'] = hargaBaru
                    updated = True
                else:
                    print("Harga tidak boleh negatif. Perubahan
harga dibatalkan.")
            except ValueError:
                print("Harga harus berupa angka. Perubahan harga
dibatalkan.")

        if updated:
            print(f"Kode obat {kodeObat} berhasil diperbarui.")
        elif not stok_baru_str and not harga_baru_str:
            print("Tidak ada perubahan dilakukan.")
        else:
            print(f"Kode obat {kodeObat} tidak ditemukan.")

```

3.13 Fitur Opsi Menu 4 (Hapus Data)

```

elif opsi_menu == 4:
    # Menghapus obat
    print("\n--- Hapus Obat ---")

    kodeObat = None
    while True:
        inputKode = input("Masukkan kode obat yang akan dihapus: ")
        inputKode = inputKode.strip()

        try:
            kodeObat = int(inputKode)
            break
        except ValueError:
            print("Kode obat harus berupa angka.")

    if kodeObat in obat_data:
        nama_obat_hapus = obat_data[kodeObat]['nama']
        del obat_data[kodeObat]
        print(f"Obat '{nama_obat_hapus}' (Kode {kodeObat}) berhasil dihapus.")
    else:
        print(f"Kesalahan: Kode obat {kodeObat} tidak ditemukan.")

```

3.14 Fitur Opsi Menu 5 (Logout)

```

# 5. Logout

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

```

3.15 Fitur User Biasa

```

elif role_pengguna == "user":
    # --- MENU USER BIASA ---
    print(f"\n=== Menu Pengguna ({pengguna}) ===")
    print("1. Lihat Daftar Obat")
    print("2. Beli Obat")
    print("3. Logout")

    opsi_menu = get_int_input("Pilih Opsi (1-3): ", 1, 3)

```

3.16 Fitur Opsi Menu 1 User (Tampil Data)

```
# Pilihan Menu User
    if opsi_menu == 1:
        tampilkan_daftar_obat_user()
```

3.17 Fitur Opsi menu 2 User (Beli Obat)

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

        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 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
{obat_data[kodeObat]['nama']}: ").strip()
            try:
                jumlahBeli = int(inputJumlah)
                break
            except ValueError:
                print("Jumlah beli harus berupa angka.")
        proses_pembelian(kodeObat, jumlahBeli)
```

3.18 Fitur Logout User

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

4. Hasil Output

```
=== Selamat Datang di Toko Obat Sehat ===
1. Login
2. Registrasi
3. Keluar
Pilih Opsi (1-3): apoteker
Input harus berupa angka.
Pilih Opsi (1-3): 1234
Opsi tidak valid. Pilih antara 1 dan 3.
Pilih Opsi (1-3): 1
Username: apoteker
Password: 1234
Login berhasil! Selamat datang, apoteker (apoteker).
```

Gambar 4.1 Hasil Output Python
(Penerapan Try Except Pada Menu Pra-login)

```
=== Menu Admin (apoteker) ===
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: 101 | Nama: Paracetamol | Stok: 50 | Harga: 5,000
Kode: 102 | Nama: Amoxicillin | Stok: 30 | Harga: 15,000
Kode: 103 | Nama: Komix | Stok: 20 | Harga: 10,000
Kode: 104 | Nama: OBH | Stok: 15 | Harga: 13,000
Kode: 105 | Nama: Tremenza | Stok: 20 | Harga: 20,000
Kode: 106 | Nama: Omeprazol | Stok: 13 | Harga: 15,000
```

Gambar 4.2 Hasil Output Lihat Data

```

=== Menu Admin (apoteker) ===
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: OBH Batuk
Stok: dua belas
Stok harus berupa angka.
Stok: 12
Harga: lima belas ribu
Harga harus berupa angka.
Harga: 15000
Obat 'OBH Batuk' berhasil ditambahkan dengan kode 107.

```

Gambar 4.3 Hasil Output Tambah Data
(Penerapan Try Except Pada Menu Tambah Data)

```

=== Menu Admin (apoteker) ===
1. Lihat Daftar Obat
2. Tambah Obat Baru
3. Perbarui Stok/Harga Obat
4. Hapus Obat
5. Logout
Pilih Opsi (1-5): 3

--- Perbarui Obat ---
Masukkan kode obat yang akan diubah: seratus tiga
Kode obat harus berupa angka.
Masukkan kode obat yang akan diubah: 103
Obat yang dipilih: Komix
Masukkan Stok Baru (kosongkan jika tidak diubah): dua puluh
Masukkan Harga Baru (kosongkan jika tidak diubah): dua bels ribu
Stok harus berupa angka. Perubahan stok dibatalkan.
Harga harus berupa angka. Perubahan harga dibatalkan.

```

```

=== Menu Admin (apoteker) ===
1. Lihat Daftar Obat
2. Tambah Obat Baru
3. Perbarui Stok/Harga Obat
4. Hapus Obat
5. Logout
Pilih Opsi (1-5): 3

--- Perbarui Obat ---
Masukkan kode obat yang akan diubah: 103
Obat yang dipilih: Komix
Masukkan Stok Baru (kosongkan jika tidak diubah): 20
Masukkan Harga Baru (kosongkan jika tidak diubah): 12000
Kode obat 103 berhasil diperbarui.

```

Gambar 4.4 Hasil Output Update Data Obat
(Penerapan Try Except Pada Menu Update Data)

```
=== Menu Admin (apoteker) ===
1. Lihat Daftar Obat
2. Tambah Obat Baru
3. Perbarui Stok/Harga Obat
4. Hapus Obat
5. Logout
Pilih Opsi (1-5): 4

--- Hapus Obat ---
Masukkan kode obat yang akan dihapus: seratus lima
Kode obat harus berupa angka.
Masukkan kode obat yang akan dihapus: 105
Obat 'Tremenza' (Kode 105) berhasil dihapus.
```

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

```
=== Menu Admin (apoteker) ===
1. Lihat Daftar Obat
2. Tambah Obat Baru
3. Perbarui Stok/Harga Obat
4. Hapus Obat
5. Logout
Pilih Opsi (1-5): 5
Pengguna apoteker berhasil logout.
```

Gambar 4.6 Hasil Output Logout

```
--- Registrasi Pengguna Baru ---
Masukkan Username Baru: User3
Masukkan Password: 890
Password minimal 4 karakter.

--- Registrasi Pengguna Baru ---
Masukkan Username Baru: User2
Masukkan Password: 6789
Registrasi User2 berhasil. Silakan Login.

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

Gambar 4.7 Hasil Output Registrasi
(Penerapan Try Except Pada Menu Register)

```

=== Selamat Datang di Toko Obat Sehat ===
1. Login
2. Registrasi
3. Keluar
Pilih Opsi (1-3): 1
Username: User2
Password: 6789
Login berhasil! Selamat datang, User2 (user).

```

Gambar 4.8 Hasil Output Login Kembali Setelah Regis

```

=== Menu Pengguna (User2) ===
1. Lihat Daftar Obat
2. Beli Obat
3. Logout
Pilih Opsi (1-3): 1

--- Daftar Obat ---

Kode: 101 | Nama: Paracetamol | Harga: 5000 | Stok: 50
Kode: 102 | Nama: Amoxicillin | Harga: 15000 | Stok: 30
Kode: 103 | Nama: Komix | Harga: 12000 | Stok: 20
Kode: 104 | Nama: OBH | Harga: 13000 | Stok: 15
Kode: 106 | Nama: Omeprazol | Harga: 15000 | Stok: 13
Kode: 107 | Nama: OBH Batuk | Harga: 15000 | Stok: 12

```

Gambar 4.9 Hasil Output Lihat Daftar Obat Menu User

```

=== Menu Pengguna (User2) ===
1. Lihat Daftar Obat
2. Beli Obat
3. Logout
Pilih Opsi (1-3): 2

--- Daftar Obat ---

Kode: 101 | Nama: Paracetamol | Harga: 5000 | Stok: 50
Kode: 102 | Nama: Amoxicillin | Harga: 15000 | Stok: 30
Kode: 103 | Nama: Komix | Harga: 12000 | Stok: 20
Kode: 104 | Nama: OBH | Harga: 13000 | Stok: 15
Kode: 106 | Nama: Omeprazol | Harga: 15000 | Stok: 13
Kode: 107 | Nama: OBH Batuk | Harga: 15000 | Stok: 12

--- Proses Pembelian Obat ---
Masukkan kode obat yang akan dibeli: seratus empat
Kode obat harus berupa angka.
Masukkan kode obat yang akan dibeli: 104
Masukkan jumlah beli OBH: 2

Pembelian berhasil:
- Obat: OBH
- Jumlah: 2
- Total Harga: 26,000

```

Gambar 4.10 Hasil Output Beli Obat


```
=== Menu Pengguna (User2) ===  
1. Lihat Daftar Obat  
2. Beli Obat  
3. Logout  
Pilih Opsi (1-3): 3  
Pengguna User2 berhasil logout.
```

Gambar 4.11 Hasil Output Logout User

```
=== Selamat Datang di Toko Obat Sehat ===  
1. Login  
2. Registrasi  
3. Keluar  
Pilih Opsi (1-3): 3  
Terima kasih, program diakhiri.  
PS C:\Users\ASUS-GK\Documents\PRAKTIKUM_APD_B2'25>
```

Gambar 4.12 Hasil Output Akhir

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_NAYLANURELSYA  
DA_PT_7"  
[main 3e1c322] 2509106070_NAYLANURELSYADA_PT_7  
1 file changed, 343 insertions(+)  
create mode 100644 POSTEST/POST_TEST_APD_7/2509106070_NAYLANURELSYADA_PT_7.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: 76, done.
Counting objects: 100% (76/76), done.
Delta compression using up to 4 threads
Compressing objects: 100% (73/73), done.
Writing objects: 100% (76/76), 2.25 MiB | 1.04 MiB/s, done.
Total 76 (delta 18), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (18/18), done.
To https://github.com/elsyadaa-del/Praktikum_APD_B2-25.git
+ 33349f9...3e1c322 main -> main (forced update)
PS C:\Users\ASUS-GK\Documents\PRAKTIKUM_APD_B2'25>
```

Gambar 5.3 git push