

LAPORAN PROYEK AKHIR
DASAR-DASAR PEMROGRAMAN
SISTEM MANAJEMEN APOTEK



Disusun Oleh:
KELOMPOK 5

Fitri Yanti	2309116016
Satria Alfiandi R. Akbar	2509116089
Muhammad Risqy Alpianur	2509116101

Asisten Laboratorium:

Taufik Ramadhani
2409116001

Dwi Pebriyanto Pradana
2409116012

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
2025

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas rahmat, taufik, serta hidayah-Nya sehingga penulis dapat menyelesaikan laporan proyek akhir mata kuliah *Dasar Pemrograman* yang berjudul “Sistem Manajemen Apotek” dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu syarat untuk memenuhi penilaian akhir dalam mata kuliah *Dasar Pemrograman*. Melalui proyek ini, penulis berupaya untuk menerapkan berbagai konsep dasar dalam pemrograman seperti variabel, tipe data, operator, percabangan, perulangan, fungsi, serta struktur data seperti list, tuple, dan dictionary.

Tema “Sistem Manajemen Apotek” dipilih karena dinilai relevan dengan kebutuhan dunia nyata dalam pengelolaan data apotek, seperti pengelolaan stok obat, transaksi pembelian, serta pencatatan pelanggan. Sistem ini diharapkan dapat membantu memahami penerapan logika pemrograman dalam konteks pengelolaan bisnis sederhana.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan demi penyempurnaan laporan dan pengembangan kemampuan di masa mendatang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu mata kuliah *Dasar Pemrograman*, teman-teman, serta semua pihak yang telah membantu dalam penyusunan laporan ini. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi bagi pembaca yang ingin mempelajari penerapan dasar-dasar pemrograman dalam pembuatan sistem manajemen sederhana.

Samarinda, 26 Oktober 2025

Penyusun

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL	iv
DAFTAR LAMPIRAN.....	v
BAB I PENDAHULUAN	1
1.1 Deskripsi Masalah.....	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
BAB II PERANCANGAN.....	3
2.1 Analisis Program	3
2.2 Flowchart “Sistem Manajemen Apotek”	4
Gambar 2.1 Flowchart 1.1	4
BAB III HASIL DAN PEMBAHASAN.....	5
3.1 Implementasi Program.....	5
3.2 Alur Program	5
BAB IV PENUTUP	28
4.1 Kesimpulan.....	28
4.2 Saran.....	28
DAFTAR PUSTAKA	29
LAMPIRAN	30

DAFTAR GAMBAR

Gambar 2. 1 Flowchart	4
Gambar 3. 1.....	5
Gambar 3. 2.....	5
Gambar 3. 3.....	6
Gambar 3. 4.....	6
Gambar 3. 5.....	6
Gambar 3. 6.....	7
Gambar 3. 7.....	7
Gambar 3. 8.....	8
Gambar 3. 9.....	8
Gambar 3. 10.....	8
Gambar 3. 11.....	9
Gambar 3. 12.....	9
Gambar 3. 13.....	10
Gambar 3. 14.....	10
Gambar 3. 15.....	11
Gambar 3. 16.....	11
Gambar 3. 17.....	11
Gambar 3. 18.....	12
Gambar 3. 19.....	12
Gambar 3. 20.....	13
Gambar 3. 21.....	13
Gambar 3. 22.....	13
Gambar 3. 23.....	14
Gambar 3. 24.....	14
Gambar 3. 25.....	14
Gambar 3. 26.....	15
Gambar 3. 27.....	15
Gambar 3. 28.....	16
Gambar 3. 29.....	16
Gambar 3. 30.....	17
Gambar 3. 31.....	18
Gambar 3. 32.....	18

DAFTAR TABEL

Tabel 3. 1 Source Code	13
-------------------------------------	----

DAFTAR LAMPIRAN

Lampiran 1 : Tabel Kontribusi.....	24
---	-----------

BAB I

PENDAHULUAN

1.1 Deskripsi Masalah

Apotek merupakan salah satu komponen penting dalam bidang kesehatan yang berfungsi sebagai tempat penyaluran obat-obatan dan kebutuhan medis kepada masyarakat. Dalam kegiatan operasionalnya, apotek memerlukan sistem yang terorganisir untuk mengelola berbagai data, seperti data obat, data transaksi penjualan, serta informasi pelanggan. Namun, dalam praktiknya masih banyak apotek yang melakukan pencatatan secara manual, baik di buku catatan maupun lembar kerja sederhana.

Pencatatan manual tersebut sering kali menimbulkan berbagai kendala, seperti kesalahan dalam penghitungan stok obat, keterlambatan dalam pencatatan transaksi, serta kesulitan dalam mencari data obat atau pelanggan. Hal ini dapat berdampak pada pelayanan yang kurang optimal kepada konsumen dan menghambat efisiensi kerja karyawan apotek.

Oleh karena itu, dibutuhkan sebuah sistem berbasis komputer yang mampu membantu proses manajemen apotek agar lebih efektif, efisien, dan akurat. Sistem ini diharapkan dapat melakukan pengelolaan data obat, pencatatan transaksi, serta pelaporan yang terstruktur seperti pada saat user (pembeli) melihat daftar obat yang akan ditampilkan.

Melalui proyek “*Sistem Manajemen Apotek*” ini, penulis berupaya untuk merancang dan mengimplementasikan program sederhana menggunakan bahasa pemrograman Python yang dapat mensimulasikan proses manajemen apotek secara digital. Sistem ini akan membantu pengguna dalam menambah, menghapus, memperbarui, serta menampilkan data obat dan transaksi secara otomatis, sehingga kegiatan operasional apotek menjadi lebih terkontrol dan efisien.

1.2 Rumusan Masalah

Berdasarkan deskripsi masalah yang telah dijelaskan sebelumnya, maka dapat dirumuskan beberapa masalah yang akan dibahas dalam proyek akhir ini, yaitu sebagai berikut:

1. Bagaimana cara membuat sistem sederhana yang dapat mengelola data obat di apotek menggunakan bahasa pemrograman Python?
2. Bagaimana sistem dapat membantu mencatat dan menampilkan transaksi penjualan obat dengan cepat dan akurat?
3. Bagaimana membuat role admin yang dapat menambahkan, menghapus dan memperbarui daftar obat yang telah ditampilkan?
4. Bagaimana sistem dapat menampilkan data obat dalam format tabel yang terstruktur dan mudah dibaca oleh pengguna?

1.3 Batasan Masalah

Agar pembahasan dalam proyek akhir ini lebih terarah dan tidak meluas, maka diperlukan adanya batasan masalah. Adapun batasan masalah dalam pembuatan sistem manajemen apotek ini adalah sebagai berikut:

1. Sistem yang kami buat hanya berfokus pada pengelolaan data obat dan transaksi penjualan sederhana di apotek.
2. Sistem hanya menggunakan bahasa pemrograman Python dengan modul standar seperti CSV, pwinput, dan prettytable.
3. Data obat hanya disimpan dalam file CSV tanpa menggunakan basis data eksternal seperti MySQL atau SQLite.
4. Sistem hanya memiliki dua peran utama pengguna, yaitu Admin dan User.

4.4 Tujuan

Adapun tujuan dari pembuatan proyek akhir ini adalah untuk menjawab rumusan masalah yang telah dirumuskan sebelumnya, yaitu sebagai berikut:

1. Untuk membuat sistem manajemen apotek sederhana yang dapat mengelola data obat seperti menambah, menghapus, memperbarui, dan melihat stok obat.
2. Untuk membantu mencatat dan menampilkan transaksi penjualan obat dengan cepat dan akurat.
3. Agar admin dapat menambahkan, menghapus dan memperbarui daftar obat yang telah ditampilkan.
4. Dapat menampilkan data obat dalam format tabel yang terstruktur dan mudah dibaca oleh pengguna.

1.5 Manfaat

Proyek akhir pembuatan sistem manajemen apotek ini diharapkan dapat memberikan berbagai manfaat. Adapun manfaat dari pembuatan sistem ini adalah sebagai berikut:

1. Bagi mahasiswa, proyek akhir ini dapat menjadi sarana untuk menerapkan pengetahuan dasar pemrograman ke dalam pembuatan aplikasi nyata yang bermanfaat di bidang kesehatan.
2. Bagi pengguna atau pembeli, sistem ini membantu dalam memperoleh informasi obat dengan lebih cepat dan akurat tanpa harus mencatat secara manual.
3. Bagi pihak apotek, sistem ini memudahkan dalam pengelolaan data stok obat, pencatatan transaksi penjualan, dan laporan keuangan secara lebih efisien.
4. Secara umum, sistem ini dapat meningkatkan efektivitas dan efisiensi operasional apotek serta mengurangi kesalahan dalam pengelolaan data obat dan transaksi.

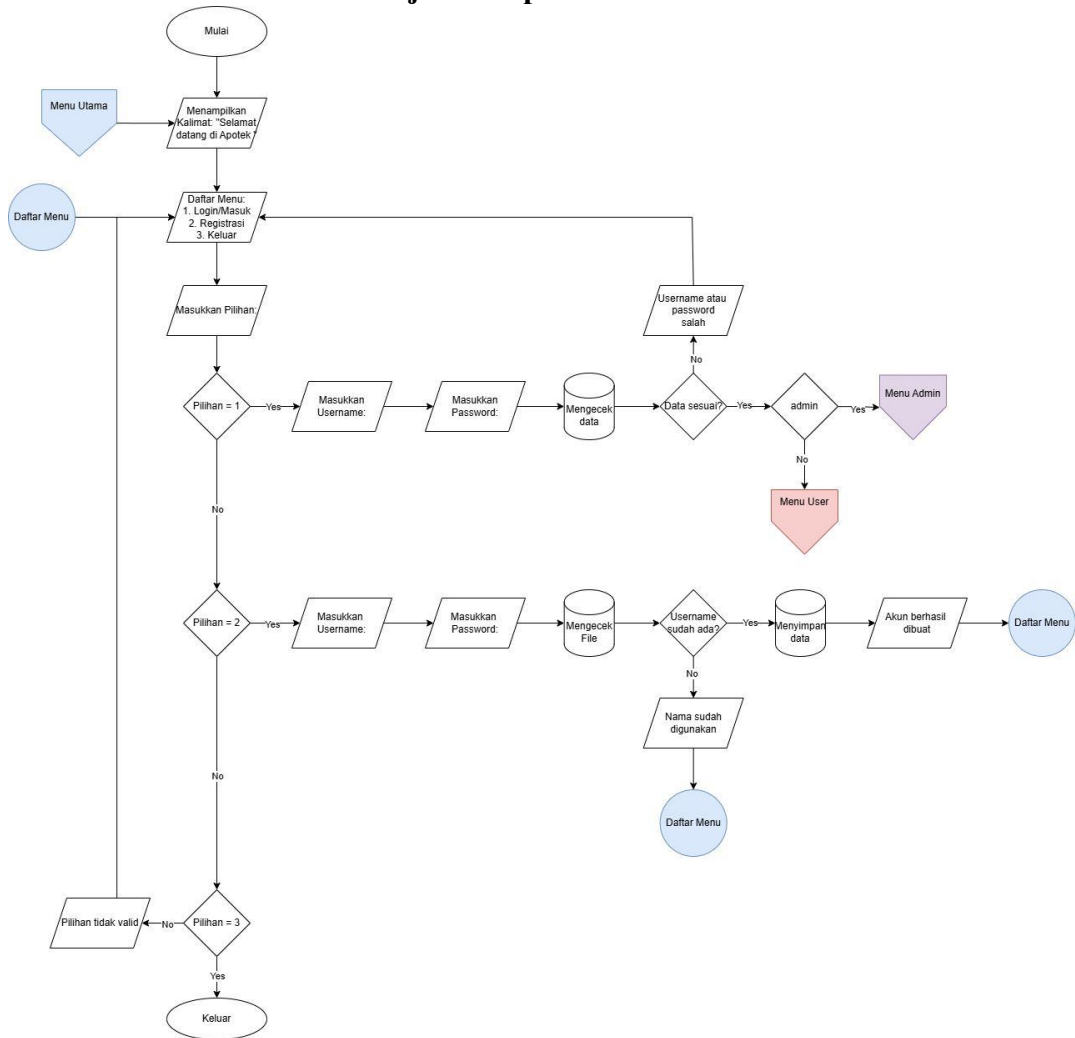
BAB II

PERANCANGAN

2.1 Analisis Program

1. “os” dalam python adalah Pustaka bawaan yang memungkinkan program berinteraksi langsung dengan sistem operasi seperti menjalankan perintah sistem. Kami menggunakan “os” untuk berinteraksi atau langsung tulis perintah tanpa perlu membuka panel di program komputer .
2. “csv” dalam python merujuk pada file “Comma Separated Values” yang merupakan format teks untuk menyimpan data tabular, di mana setiap baris mewakili satu baris data dan nilai-nilai di setiap baris dipisahkan oleh koma (atau delimiter lain). Dalam program ini kami gunakan ntuk menyimpan data obat, meliputi nama obat, harga, jumlah stok, dan tingkat obat. Kami menggunakan “os” untuk berinteraksi atau langsung tulis perintah tanpa perlu membuka panel di program computer.
- 3.“prettytable” dalam python ialah Pustaka (library) yang memungkinkan anda membuat dan mencetak table dalam format yang menarik dan mudah dibaca langsung di konsol. Pustaka ini menyediakan cara mudah untuk mengubah data tabular, seperti dari daftar atau file CSV, menjadi table ASCII yang terformat dengan baik, lengkap dengan opsi kustomisasi seperti perataan teks, gaya batas, dan lebar kolom. Dalam program kami, kami menggunakan “prettytable” ini untuk menampilkan data obat dari file CSV dalam bentuk tabel yang rapi di terminal. Tanpa library ini, data hanya akan muncul dalam bentuk teks biasa.
4. “pwininput” dalam python adalah singkatan dari "password input," yang merujuk pada cara aman untuk mendapatkan input kata sandi dari pengguna tanpa menampilkannya di layer. Kami menggunakan “pwininput” agar password tidak dapat terlihat di layer.

2.2 Flowchart “Sistem Manajemen Apotek”



Gambar 2. 1 Flowchart

BAB III

HASIL DAN PEMBAHASAN

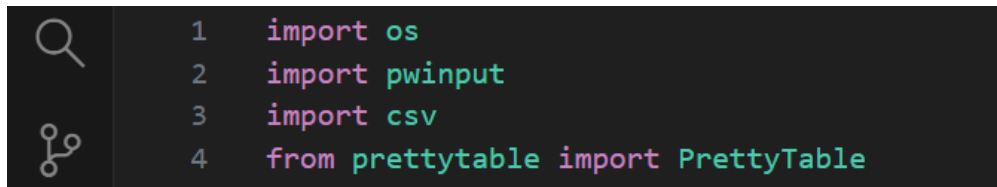
3.1 Implementasi Program

Dalam implementasinya di real life (dunia nyata), program “Sistem Manajemen Apotek” kami dirancang untuk membantu proses pengelolaan data obat dan akun pengguna secara lebih efisien, terstruktur, dan mudah digunakan. Secara real-life, program ini dapat diterapkan di lingkungan apotek kecil hingga menengah sebagai sistem administrasi dasar untuk membantu pemilik atau karyawan apotek dalam:

1. Mencatat data obat yang tersedia
2. Menampilkan daftar stok obat yang tersedia
3. Menampilkan daftar stok obat dalam bentuk obat
4. Meningkatkan efisiensi kerja

3.2 Alur Program

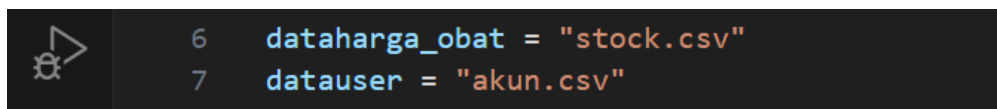
Alur Program yang kami buat dapat dijalankan sebagai berikut :



```
1 import os
2 import pwininput
3 import csv
4 from prettytable import PrettyTable
```

Gambar 3. 1

- **import os** : kami menggunakan os untuk menjalankan perintah sistem seperti cls (membersihkan layar).
- **import pwininput** : kami menggunakan pwininput agar input password tidak terlihat saat diketik.
- **import csv** : kami menggunakan csv untuk untuk membaca dan menulis data ke file.csv (data obat dan akun pengguna).
- **from prettytable import PrettyTable** : kami menggunakan prettytable untuk menampilkan tabel dengan tampilan yang rapi di terminal.



```
6 dataharga_obat = "stock.csv"
7 datauser = "akun.csv"
```

Gambar 3. 2

- **stock.csv** : Menyimpan data obat (ID, nama, harga, jumlah, tingkatan).
- **akun.csv** : Menyimpan data user, password, role, saldo.

```

8
9 def dtuser ():
10     if not os.path.exists(datauser):
11         with open(datauser, mode="w", newline="") as f:
12             writer = csv.writer(f)
13             writer.writerow(["username", "password", "role", "saldo"])
14

```

Gambar 3. 3

- **dtuser()**, membuat file akun.csv beserta header (username, password, role, saldo) jika belum ada. **Tujuannya** menjamin file data utama tersedia agar program tidak error saat pertama kali dijalankan.

```

14
15 def hrgaobt():
16     if not os.path.exists(dataharga_obat):
17         with open(dataharga_obat, mode="w", newline="") as f:
18             writer = csv.writer(f)
19             writer.writerow(["ID", "Nama", "Harga", "Jumlah", "Tingkatan"])
20

```

Gambar 3. 4

- **hrgaobt()**, membuat file stock.csv dengan header (ID, Nama, Harga, Jumlah, Tingkatan) jika belum ada. **Tujuannya** menjamin file data utama tersedia agar program tidak error saat pertama kali dijalankan.

```

20
21 def clear():
22     os.system("cls")
23

```

Gambar 3. 5

1. **clear()**, membersihkan tampilan layar konsol agar tampilan lebih rapi saat berpindah menu. **Tujuannya** memperindah tampilan antarmuka CLI (Command Line Interface).

```

24 def register():
25     clear()
26     dtuser()
27     username = input("Masukkan username baru: ")
28     if len(username) > 10:
29         print("=====+")
30         print("|                      Nama Tidak Boleh Lebih Dari 10 Karakter                      |")
31         print("=====+")
32         return
33
34     password = pwinput.pwinput("Masukkan password: ")
35     if len(password) < 8:
36         print("=====+")
37         print("|                      Harus Lebih Dari 8 Angka atau Huruf                      |")
38         print("=====+")
39         return
40
41     with open(datauser, mode="r") as f:
42         reader = csv.DictReader(f)
43         for row in reader:
44             if row["username"] == username:
45                 print("=====+")
46                 print("|                      Username Sudah Tendaftar, Silahkan Coba Nama Lain                      |")
47                 print("=====+")
48                 return
49
50     with open(datauser, mode="a", newline="") as f:
51         writer = csv.writer(f)
52         writer.writerow([username, password, "User", "0"])
53     print(f"Akun '{username}' berhasil dibuat.\n")
54

```

Gambar 3. 6

2. **register()**, memanggil dtuser() agar file akun siap digunakan. Meminta username dan password dari pengguna baru. Mengecek panjang username/password agar sesuai aturan (username maksimal 10 karakter dan password minimal 8 karakter), dan mengecek apakah username sudah ada di CSV. Jika belum, menyimpan data baru ke file akun.csv dengan role default "User" dan saldo awal 0. **Tujuannya** menambahkan akun baru agar pengguna bisa login dan menggunakan sistem (tanpa register, user tidak bisa berinteraksi dengan sistem). **Outputnya** jika dijalankan sebagai berikut:

```

>  TERMINAL

Masukkan username baru: akbar
Masukkan password: *****
Akun 'akbar' berhasil dibuat.

Tekan Enter untuk melanjutkan...

```

Gambar 3. 7

```

55 def login():
56     clear()
57     if not os.path.exists(datauser):
58         print("Belum ada akun. Silakan registrasi dulu.")
59         return
60
61     username = input("Masukkan username: ")
62     password = pwininput.pwininput("Masukkan password: ")
63
64     found = False
65     with open(datauser, mode="r") as f:
66         reader = csv.DictReader(f)
67         for row in reader:
68             if row["username"] == username and row["password"] == password:
69                 found = True
70                 role = row["role"]
71                 saldo = float(row["saldo"])
72                 print(f"\nBerhasil login sebagai {role}")
73                 print(f"Selamat datang, {username} | Saldo: Rp {saldo:,.0f}\n")
74
75                 if role.lower() == "admin":
76                     menu_admin()
77                 else:
78                     menu_user(username, saldo)
79                 break
80     if not found:
81         print("Nama atau password salah.\n")

```

Gambar 3. 8

3. **login()**, mengecek apakah file akun.csv ada, minta input username & password, dan mencocokkan data dari file CSV. Jika cocok maka menampilkan role dan saldo user. Jika Admin maka masuk ke menu_admin(), jika User maka masuk ke menu_user(). **Tujuannya** sebagai gerbang masuk sistem bagi pengguna terdaftar, sekaligus membedakan hak akses antara Admin dan User. **Outputnya** sebagai berikut:

```

> TERMINAL
Masukkan username: akbar
Masukkan password: *****

```

Gambar 3. 9

```

=====+
|                                     Menu User                                     |
|=====+
| [1]. Lihat Daftar Obat |
| [2]. Isi Saldo         |
| [3]. Beli Obat         |
| [4]. Lihat Saldo       |
| [0]. Keluar            |
|=====+
Masukkan pilihan (0-4): 

```

Gambar 3. 10

```

83 def tambahkan():
84     try:
85         aid = input("Masukkan ID: ")
86         if int(aid) <= 0:
87             print("Masukan ID harus berupa angka dan lebih dari 0.")
88             return
89
90         if os.path.exists(dataharga_obat):
91             with open(dataharga_obat, mode="r") as file:
92                 reader = csv.reader(file)
93                 next(reader, None)
94                 for row in reader:
95                     if row and row[0] == aid:
96                         print("\n+=====+")
97                         print("|                                ID sudah Terdaftar                                |")
98                         print("+=====+\n")
99                         return
100
101         nama = input("Masukkan nama obat: ")
102         if not nama.strip():
103             print("\n+=====+")
104             print("|                                Nama Tidak Boleh Kosong                                |")
105             print("+=====+\n")
106             return
107
108         harga = float(input("Masukkan harga obat: "))
109         if harga <= 0:
110             print("\n+=====+")
111             print("|                                Harga tidak boleh kurang dari atau sama dengan 0                                |")
112             print("+=====+\n")
113             return
114
115
116

```

Gambar 3. 11

```

116
117     jumlah = int(input("Masukkan jumlah stok obat: "))
118     if jumlah < 0:
119         print("\n+=====+")
120         print("|                                Jumlah tidak boleh negatif                                |")
121         print("+=====+\n")
122         return
123     elif len(str(jumlah)) > 10:
124         print("\n+=====+")
125         print("|                                Jumlah tidak boleh lebih dari 10 digit                                |")
126         print("+=====+\n")
127         return
128
129     tingkat = input("Masukkan tingkatan obat (biasa/keras): ").lower()
130     if tingkat not in ["biasa", "keras"]:
131         print("\n+=====+")
132         print("|                                Tingkatan hanya boleh 'biasa' atau 'keras'                                |")
133         print("+=====+\n")
134         return
135
136     with open(dataharga_obat, mode="a", newline="") as file:
137         writer = csv.writer(file)
138         writer.writerow([aid, nama, harga, jumlah, tingkat])
139
140     print("\n+=====+")
141     print("|                                Obat Berhasil Ditambahkan                                |")
142     print("+=====+\n")
143     except ValueError:
144         print("\n+=====+")
145         print("|                                Input tidak valid, pastikan harga dan jumlah berupa angka                                |")
146         print("+=====+\n")
147     except KeyboardInterrupt:
148         (""")
149     except EOFError:
150         (""")
151

```

Gambar 3. 12

4. **tambahkan()**, hanya bisa diakses oleh Admin, menambah data obat baru ke stock.csv, mengecek agar ID obat unik, harga positif, jumlah tidak negatif, dan tingkatan hanya “biasa” atau “keras”. **Tujuannya** menambah stok obat baru dalam database apotek. Ini adalah fitur utama manajemen stok oleh Admin.

```

151
152     def lihat():
153         try:
154             with open(dataharga_obat, mode="r") as file:
155                 reader = csv.reader(file)
156                 next(reader)
157                 table = PrettyTable(["ID", "Nama", "Harga", "Jumlah", "Tingkatan"])
158                 for row in reader:
159                     table.add_row(row)
160                 print(table)
161         except FileNotFoundError:
162             print("Obat belum terdaftar, silakan tambahkan obat terlebih dahulu.")
163

```

Gambar 3. 13

5. **lihat()**, menampilkan seluruh data obat dalam bentuk tabel rapi menggunakan PrettyTable. **Tujuannya** agar User dan Admin dapat melihat daftar obat yang tersedia beserta harganya dan stoknya. **Outputnya** jika dijalankan sebagai berikut:

```

+-----+
|                                     |
|                               Menu User                               |
|-----+-----+-----+-----+-----+
| [1]. Lihat Daftar Obat                                             |
| [2]. Isi Saldo                                                       |
| [3]. Beli Obat                                                       |
| [4]. Lihat Saldo                                                     |
| [0]. Keluar                                                         |
|-----+-----+-----+-----+-----+
Masukkan pilihan (0-4): 1
+-----+-----+-----+-----+-----+
| ID | Nama      | Harga | Jumlah | Tingkatan |
+-----+-----+-----+-----+-----+
| 01 | Paracetamol | 5000 | 100    | biasa     |
| 02 | Amoxicillin | 8000 | 131    | keras     |
| 03 | Vitamin C   | 3000 | 100    | biasa     |
| 04 | Ibuprofen   | 7000 | 90     | keras     |
| 05 | Antalgin    | 6000 | 70     | biasa     |
| 06 | Sanmol      | 4000 | 150    | biasa     |
| 07 | Cetirizine  | 9000 | 60     | keras     |
| 08 | Omeprazole  | 10000 | 50    | keras     |
| 09 | Oralit      | 2000 | 200    | biasa     |
| 10 | Betadine    | 12000 | 40     | biasa     |
+-----+-----+-----+-----+-----+

Tekan Enter untuk lanjut...

```

Gambar 3. 14


```

163
164 def update():
165     lihat()
166     id_nomor = input("\nMasukkan ID yang mau diupdate: ")
167     update_rows = []
168     found = False
169
170     try:
171         with open(dataharga_obat, mode="r") as file:
172             reader = csv.reader(file)
173             header = next(reader, None)
174             if not header:
175                 print("File kosong, tidak ada data untuk diupdate.")
176                 return
177
178             for row in reader:
179                 if row[0] == id_nomor:
180                     nama = input("Nama baru obat (Enter jika tidak ingin mengubah): ") or row[1]
181                     harga_input = input("Harga baru (Enter jika tidak ingin mengubah): ")
182                     jumlah_input = input("Jumlah stok baru (Enter jika tidak ingin mengubah): ")
183                     tingkat = input("Masukkan tingkatan obat ('biasa'/'keras'), (Enter jika tidak ingin mengubah): ").lower() or row[4]
184
185                     if tingkat not in ["biasa", "keras"]:
186                         print("\n+=====+")
187                         print("|          Tingkatan hanya boleh 'biasa' atau 'keras'          |")
188                         print("+=====+\n")
189                         return
190

```

Gambar 3. 15

```

190
191
192         try:
193             harga = float(harga_input) if harga_input.strip() != "" else float(row[2])
194             jumlah = int(jumlah_input) if jumlah_input.strip() != "" else int(row[3])
195             except ValueError:
196                 print("Input angka tidak valid. Pembaruan dibatalkan.")
197                 return
198
199             update_rows.append([id_nomor, nama, harga, jumlah, tingkat])
200             found = True
201         else:
202             update_rows.append(row)
203
204     except FileNotFoundError:
205         print("Obat belum terdaftar.")
206         return
207
208     except ValueError:
209         print("\n+=====+")
210         print("|          Input tidak valid, pastikan harga dan jumlah berupa angka          |")
211         print("+=====+\n")
212         return
213

```

Gambar 3. 16

```

213
214     if found:
215         with open(dataharga_obat, mode="w", newline="") as file:
216             writer = csv.writer(file)
217             writer.writerow(["ID", "Nama", "Harga", "Jumlah", "Tingkatan"])
218             writer.writerows(update_rows)
219             print("\n+=====+")
220             print("|          Data berhasil diperbarui          |")
221             print("+=====+\n")
222         else:
223             print("\n+=====+")
224             print("|          ID obat tidak ditemukan          |")
225             print("+=====+\n")

```

Gambar 3. 17

6. **update()**, admin dapat memperbarui data obat berdasarkan ID. Dapat mengubah: nama, harga, stok, dan tingkatan. Validasi input agar tidak salah memasukkan data. **Tujuannya** memungkinkan Admin memperbarui informasi obat tanpa harus menghapus dan menambah ulang.

```

226 def hapus():
227     lihat()
228     id_hapus = input("\nMasukkan ID obat yang mau dihapus: ")
229     rows = []
230     found = False
231
232     try:
233         with open(dataharga_obat, mode="r") as file:
234             reader = csv.reader(file)
235             header = next(reader)
236             for row in reader:
237                 if row[0] == id_hapus:
238                     found = True
239                 else:
240                     rows.append(row)
241     except FileNotFoundError:
242         print("Obat belum terdaftar.")
243         return
244
245     if found:
246         with open(dataharga_obat, mode="w", newline="") as file:
247             writer = csv.writer(file)
248             writer.writerow(header)
249             writer.writerows(rows)
250         print(f"\nData obat dengan ID '{id_hapus}' berhasil dihapus.\n")
251     else:
252         print("\n+=====+")
253         print("|                                ID obat belum terdaftar                                |")
254         print("+=====+\n")
255

```

Gambar 3. 18

7. **hapus()**, admin memasukkan ID obat yang ingin dihapus. Program menghapus baris data tersebut dari stock.csv. **Tujuannya** agar Admin dapat menghapus obat yang sudah tidak dijual atau datanya salah.

```

255
256 def update_saldo(username, saldo):
257     rows = []
258     with open(datauser, mode="r") as f:
259         reader = csv.DictReader(f)
260         for row in reader:
261             if row["username"] == username:
262                 row["saldo"] = str(saldo)
263                 rows.append(row)
264     with open(datauser, mode="w", newline="") as f:
265         writer = csv.DictWriter(f, fieldnames=["username", "password", "role", "saldo"])
266         writer.writeheader()
267         writer.writerows(rows)
268

```

Gambar 3. 19

8. **update_saldo()** menyimpan saldo baru user ke file CSV.

```

269 def isi_saldo(username, saldobaru):
270     try:
271         tambah = float(input("Masukkan jumlah saldo yang ingin ditambahkan: "))
272
273         if tambah <= 0:
274             print("\n+=====+")
275             print("|          Jumlah saldo yang ditambahkan harus lebih dari 0          |")
276             print("+=====+\n")
277             return saldobaru
278
279         if tambah < 10000:
280             print("\n+=====+")
281             print("|          Jumlah saldo yang ditambahkan harus lebih dari Rp.10.000    |")
282             print("+=====+\n")
283             return saldobaru
284
285         if tambah > 5000000:
286             print("\n+=====+")
287             print("|          Tidak boleh menambahkan saldo lebih dari Rp 5.000.000    |")
288             print("+=====+\n")
289             return saldobaru
290
291         saldo = saldobaru + tambah
292         update_saldo(username, saldo)
293
294         print(f"\nSaldo berhasil ditambahkan! Saldo sekarang: Rp {saldo:,.0f}")
295         return saldo
296

```

Gambar 3. 20

```

296
297     except ValueError:
298         print("\n+=====+")
299         print("|          Harap masukkan angka yang valid, bukan huruf atau simbol.    |")
300         print("+=====+\n")
301         return saldobaru
302
303     except KeyboardInterrupt:
304         print("")
305         return saldobaru
306

```

Gambar 3. 21

9. **isi_saldo()**, user dapat menambah saldo dengan Batasan Minimal Rp10.000 dan Maksimal Rp5.000.000. **Outputnya** jika dijalankan sebagai berikut:

```

> TERMINAL
=====+
|                                     Menu User                                     |
+=====+
| [1]. Lihat Daftar Obat               |
| [2]. Isi Saldo                      |
| [3]. Beli Obat                     |
| [4]. Lihat Saldo                   |
| [0]. Keluar                         |
+=====+
Masukkan pilihan (0-4): 2
Masukkan jumlah saldo yang ingin ditambahkan: 100000

Saldo berhasil ditambahkan! Saldo sekarang: Rp 100,000

Tekan Enter untuk lanjut...

```

Gambar 3. 22

```

306
307 def lihat_saldo(username):
308     with open(datauser, mode="r") as file:
309         reader = csv.DictReader(file)
310         for row in reader:
311             if row["username"] == username:
312                 print(f"Saldo Anda saat ini: Rp {float(row['saldo']):,.0f}")
313                 return
314         print("Akun tidak ditemukan.")
315

```

Gambar 3. 23

10. `lihat_saldo()` menampilkan saldo user saat ini. **Outputnya** jika dijalankan sebagai berikut:

```

+=====+
|                                     |
|                               Menu User                               |
|=====+
| [1]. Lihat Daftar Obat                                     |
| [2]. Isi Saldo                                           |
| [3]. Beli Obat                                           |
| [4]. Lihat Saldo                                         |
| [0]. Keluar                                              |
|=====+
Masukkan pilihan (0-4): 4
Saldo Anda saat ini: Rp 100,000

Tekan Enter untuk lanjut...

```

Gambar 3. 24

```

316 def beli_obat(username, saldo):
317     lihat()
318     try:
319         id_obat = input("\nMasukkan ID obat yang ingin dibeli: ")
320         jumlah_beli = int(input("Masukkan jumlah yang ingin dibeli: "))
321
322         if jumlah_beli <= 0:
323             print("\n+=====+")
324             print("|                               Jumlah pembelian harus lebih dari 0                               |")
325             print("+=====+\n")
326             return saldo
327
328         rows = []
329         total_harga = 0
330         found = False
331
332         with open(dataharga_obat, mode="r") as file:
333             reader = csv.reader(file)
334             header = next(reader)
335             for row in reader:
336                 if row[0] == id_obat:
337                     found = True
338                     harga = float(row[2])
339                     stok = int(row[3])
340                     if jumlah_beli > stok:
341                         print("Stok tidak cukup.")
342                         return saldo
343                     total_harga = harga * jumlah_beli
344                     stok -= jumlah_beli
345                     row[3] = str(stok)
346                     rows.append(row)

```

Gambar 3. 25

```

347
348     if not found:
349         print("ID obat tidak ditemukan.")
350         return saldo
351
352     if saldo < total_harga:
353         print("Saldo tidak cukup.")
354         return saldo
355
356     saldo -= total_harga
357     update_saldo(username, saldo)
358
359     with open(dataharga_obat, mode="w", newline="") as file:
360         writer = csv.writer(file)
361         writer.writerow(header)
362         writer.writerows(rows)
363
364     print(f"\nPembelian berhasil! Total harga: Rp {total_harga:,.0f}")
365     print(f"Sisa saldo: Rp {saldo:,.0f}")
366     return saldo
367
368 except ValueError:
369     print("\n+=====+")
370     print("|      Input tidak valid! Masukkan jumlah dalam bentuk angka saja      |")
371     print("+=====+\n")
372     return saldo
373

```

Gambar 3. 26

11. **beli_obat()**, menampilkan daftar obat, meminta ID obat & jumlah pembelian, mengecek stok dan saldo cukup atau tidak. Jika valid maka bisa mengurangi stok di stock.csv. dan mengurangi saldo user di akun.csv. **Tujuannya** sebagai fitur utama bagi User untuk melakukan transaksi pembelian obat menggunakan saldo mereka. **Outputnya** jika dijalankan sebagai berikut:

ID	Nama	Harga	Jumlah	Tingkatan
01	Paracetamol	5000	100	biasa
02	Amoxicillin	8000	131	keras
03	Vitamin C	3000	100	biasa
04	Ibuprofen	7000	90	keras
05	Antalgin	6000	70	biasa
06	Sanmol	4000	150	biasa
07	Cetirizine	9000	60	keras
08	Omeprazole	10000	50	keras
09	Oralit	2000	200	biasa
10	Betadine	12000	40	biasa

Tekan Enter untuk lanjut...

Gambar 3. 27

```

374 def menu_user(username, saldo):
375     while True:
376         clear()
377         print("\n=====+=====")
378         print("|                               Menu User                               |")
379         print("+=====+=====")
380         print("| [1]. Lihat Daftar Obat                                             |")
381         print("| [2]. Isi Saldo                                                       |")
382         print("| [3]. Beli Obat                                                       |")
383         print("| [4]. Lihat Saldo                                                     |")
384         print("| [0]. Keluar                                                         |")
385         print("+=====+=====")
386         try:
387             pilihan = input("Masukkan pilihan (0-4): ")
388             if pilihan == "1":
389                 lihat()
390             elif pilihan == "2":
391                 saldo = isi_saldo(username, saldo)
392             elif pilihan == "3":
393                 saldo = beli_obat(username, saldo)
394             elif pilihan == "4":
395                 lihat_saldo(username)
396             elif pilihan == "0":
397                 print("Logout berhasil.")
398                 break
399             else:
400                 print("Masukkan angka yang sesuai.")
401                 input("\nTekan Enter untuk lanjut...")
402         except ValueError:
403             print("+=====+=====")
404             print("|                               Masukkan huruf dan angka, bukan karakter simbol.                               |")
405             print("+=====+=====")
406         except KeyboardInterrupt:
407             print("")
408         except EOFError:
409             print("")
410
411

```

Gambar 3. 28

12. **menu_user**, menu yang muncul setelah user login:

1. Lihat daftar obat
2. Isi saldo
3. Beli obat
4. Lihat saldo
5. Keluar

Setiap pilihan memanggil fungsi terkait di atas. **Tujuannya** memberikan antarmuka interaktif bagi pengguna biasa untuk melakukan transaksi. **Outputnya** sebagai berikut:

```

=====+=====
|                               Menu User                               |
=====+=====
| [1]. Lihat Daftar Obat                                             |
| [2]. Isi Saldo                                                       |
| [3]. Beli Obat                                                       |
| [4]. Lihat Saldo                                                     |
| [0]. Keluar                                                         |
=====+=====
Masukkan pilihan (0-4): █

```

Gambar 3. 29

```

412 def menu_admin():
413     clear()
414     while True:
415         clear()
416         print("\n+=====+")
417         print("|                               |")
418         print("|                               |")
419         print("| [1]. Tambahkan Obat           |")
420         print("| [2]. Lihat Daftar Obat        |")
421         print("| [3]. Updet daftar Obat        |")
422         print("| [4]. Hapus Obat              |")
423         print("| [0]. Keluar                   |")
424         print("|                               |")
425         print("+=====+")
426         try:
427             pilihan = input("Masukkan pilihan 0-4: ")
428             if pilihan == "1":
429                 tambah()
430             elif pilihan == "2":
431                 lihat()
432             elif pilihan == "3":
433                 update()
434             elif pilihan == "4":
435                 hapus()
436             elif pilihan == "0":
437                 print("Logout berhasil.")
438                 break
439             else:
440                 print("Masukkan angka yang sesuai.")
441                 input("\nTekan Enter untuk lanjut...")
442         except ValueError:
443             print("+=====+")
444             print("|                               |")
445             print("| Masukkan huruf dan angka, bukan karakter simbol. |")
446             print("+=====+")
447         except KeyboardInterrupt:
448             print("")
449         except EOFError:
450             print("")

```

Gambar 3. 30

13. **menu_admin**, menu yang hanya bisa diakses oleh Admin, berisi:

1. Tambah obat
2. Lihat obat
3. Update obat
4. Hapus obat
5. Keluar

Tujuannya sebagai panel manajemen data obat, agar Admin bisa mengatur stok dan harga dengan mudah.

```

449
450 def regis():
451     while True:
452         clear()
453         print("\n+=====+")
454         print("|          Selamat Datang di Sistem Management Apotek          |")
455         print("+=====+")
456         print("|                          Menu Utama                          |")
457         print("+=====+")
458         print("| [1]. Login                                                    |")
459         print("| [2]. Registrasi                                              |")
460         print("| [3]. Keluar                                                  |")
461         print("+=====+")
462         try:
463             pilihan = input("Pilih opsi: ")
464             if pilihan == "1":
465                 login()
466             elif pilihan == "2":
467                 register()
468                 input('Tekan Enter untuk melanjutkan...')
469             elif pilihan == "3":
470                 print("+=====+")
471                 print("|  Anda berhasil keluar. Terimakasih telah menggunakan pogram ini!  |")
472                 print("+=====+")
473                 break
474             else:
475                 print("+=====+")
476                 print("|                          Pilihan tidak valid!                          |")
477                 print("+=====+")
478                 input("Tekan enter untuk melanjutkan....")
479         except ValueError:
480             print("+=====+")
481             print("|          Masukkan huruf dan angka, bukan karakter simbol.          |")
482             print("+=====+")
483         except KeyboardInterrupt:
484             (""")
485         except EOFError:
486             (""")

```

Gambar 3. 31

14. **menu utama (regis)**, menampilkan utama program. Tiga opsi utama:

- a. Login
- b. Registrasi
- c. Keluar

Tujuannya menjadi titik awal seluruh sistem dijalankan, menjadi pintu utama program dan menentukan alur pengguna: apakah dia ingin login, daftar, atau keluar. **Outputnya** sebagai berikut:

```

>  TERMINAL
+=====+
|          Selamat Datang di Sistem Management Apotek          |
+=====+
|                          Menu Utama                          |
+=====+
| [1]. Login                                                    |
| [2]. Registrasi                                              |
| [3]. Keluar                                                  |
+=====+
Pilih opsi: 

```

Gambar 3. 32

3.3 Source Code

Tabel 3.1 Source Code

```
import os
import pwininput
import csv
from prettytable import PrettyTable

dataharga_obat = "stock.csv"
datauser = "akun.csv"

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

def register():
    clear()
    username = input("Masukkan username baru: ")
    password = pwininput.pwininput("Masukkan password: ")

    if not os.path.exists(datauser):
        with open(datauser, mode="w", newline="") as f:
            writer = csv.writer(f)
            writer.writerow(["username", "password", "role",
"saldo"])

        with open(datauser, mode="r") as f:
            reader = csv.DictReader(f)
            for row in reader:
                if row["username"] == username:
                    print("Nama sudah terdaftar, silakan coba nama
lain.")

        with open(datauser, mode="a", newline="") as f:
            writer = csv.writer(f)
            writer.writerow([username, password, "User", "0"])

        print(f"Akun '{username}' berhasil dibuat.\n")

def login():
    clear()
```

```

if not os.path.exists(datauser):
    print("Belum ada akun. Silakan registrasi dulu.")
    return

username = input("Masukkan username: ")
password = pwininput.pwininput("Masukkan password: ")

found = False
with open(datauser, mode="r") as f:
    reader = csv.DictReader(f)
    for row in reader:
        if row["username"] == username and row["password"] ==
password:
            found = True
            role = row["role"]
            saldo = float(row["saldo"])
            print(f"\nBerhasil login sebagai {role}")
            print(f"Selamat datang, {username} | Saldo: Rp
{saldo:,.0f}\n")

            if role.lower() == "admin":
                menu_admin()
            else:
                menu_user(username,saldo)
            break
if not found:
    print("Nama atau password salah.\n")

def tambahkan():
    try:
        aid = input("Masukkan ID: ")
        nama = input("Masukkan nama obat: ")
        harga = float(input("Masukkan harga obat: "))
        jumlah = int(input("Masukkan jumlah stok obat: "))
        tingkat = input("Masukkan tingkatan obat: ")

        if not os.path.exists(dataharga_obat):
            with open(dataharga_obat, mode="w", newline="") as f:
                writer = csv.writer(f)
                writer.writerow(["ID", "Nama", "Harga", "Jumlah",
"Tingkatan"])

            with open(dataharga_obat, mode="a", newline="") as file:
                writer = csv.writer(file)
                writer.writerow([aid, nama, harga, jumlah, tingkat])
            print("\n+=====+
=====+")

```

```

        print("|" + "Obat Berhasil" + "|")
    Ditambahkan
    print("+=====+\\n")
    =====+\\n")
    except ValueError:
        print("Input tidak valid, pastikan harga dan jumlah berupa
angka.")

def lihat():
    try:
        with open(dataharga_obat, mode="r") as file:
            reader = csv.reader(file)
            next(reader)
            table = PrettyTable(["ID", "Nama", "Harga", "Jumlah",
"Tingkatan"])
            for row in reader:
                table.add_row(row)
            print(table)
    except FileNotFoundError:
        print("Obat belum terdaftar, silakan tambahkan obat
terlebih dahulu.")

def update():
    lihat()
    id_nomor = input("\\nMasukkan ID yang mau diupdate: ")
    update_rows = []
    found = False

    try:
        with open(dataharga_obat, mode="r") as file:
            reader = csv.reader(file)
            header = next(reader)
            for row in reader:
                if row[0] == id_nomor:

                    nama_baru = input("Nama baru obat: ") or row[1]
                    harga_input = input("Harga baru: ")
                    jumlah_input = input("Jumlah stok baru: ")
                    tingkat_baru = input("Tingkatan obat baru: ")
or row[4]

                    try:
                        harga_baru = float(harga_input) if
harga_input.strip() != "" else float(row[2])
                        jumlah_baru = int(jumlah_input) if
jumlah_input.strip() != "" else int(row[3])
                    except ValueError:
                        print("Input angka tidak valid. Pembaruan
dibatalkan.")

```

```

        return

        update_rows.append([id_nomor, nama_baru,
harga_baru, jumlah_baru, tingkat_baru])
        found = True
    else:
        update_rows.append(row)
except FileNotFoundError:
    print("Obat belum terdaftar.")
    return

if found:
    with open(dataharga_obat, mode="w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow(["ID", "Nama", "Harga", "Jumlah",
"Tingkatan"])
        writer.writerows(update_rows)
        print("\n+=====+
=====+")
        print("|                               Data berhasil
diperbarui                               |")
        print("+=====+
=====+\n")
    else:
        print("\n+=====+
=====+")
        print("|                               ID obat tidak
ditemukan                               |")
        print("+=====+
=====+\n")

def hapus():
    lihat()
    id_hapus = input("\nMasukkan ID obat yang mau dihapus: ")
    rows = []
    found = False

    try:
        with open(dataharga_obat, mode="r") as file:
            reader = csv.reader(file)
            header = next(reader)
            for row in reader:
                if row[0] == id_hapus:
                    found = True
                else:
                    rows.append(row)
    except FileNotFoundError:
        print("Obat belum terdaftar.")

```

```

        return

    if found:
        with open(dataharga_obat, mode="w", newline="") as file:
            writer = csv.writer(file)
            writer.writerow(header)
            writer.writerows(rows)
        print(f"\nIDData obat dengan ID '{id_hapus}' berhasil
dihapus.\n")
    else:
        print("\n+=====
=====+")
        print("|                  ID obat belum
terdaftar                  |")
        print("+=====
=====+\n")
def update_saldo(username, saldo_baru):
    rows = []
    with open(datauser, mode="r") as f:
        reader = csv.DictReader(f)
        for row in reader:
            if row["username"] == username:
                row["saldo"] = str(saldo_baru)
                rows.append(row)
    with open(datauser, mode="w", newline="") as f:
        writer = csv.DictWriter(f, fieldnames=["username",
"password", "role", "saldo"])
        writer.writeheader()
        writer.writerows(rows)

def isi_saldo(username, saldo_sekarang):
    try:
        tambah = float(input("Masukkan jumlah saldo yang ingin
ditambahkan: "))
        saldo_baru = saldo_sekarang + tambah
        update_saldo(username, saldo_baru)

        print(f"Saldo berhasil ditambahkan! Saldo sekarang: Rp
{saldo_baru:,.0f}")
        return saldo_baru
    except ValueError:
        print("Input tidak valid.")
        return saldo_sekarang

def lihat_saldo(username):
    with open(datauser, mode="r") as file:
        reader = csv.DictReader(file)
        for row in reader:

```

```

        if row["username"] == username:
            print(f"Saldo Anda saat ini: Rp
{float(row['saldo']):,.0f}")
            return
        print("Akun tidak ditemukan.")

def beli_obat(username, saldo):
    lihat()
    id_obat = input("\nMasukkan ID obat yang ingin dibeli: ")
    jumlah_beli = int(input("Masukkan jumlah yang ingin dibeli: "))

    rows = []
    total_harga = 0
    found = False

    with open(dataharga_obat, mode="r") as file:
        reader = csv.reader(file)
        header = next(reader)
        for row in reader:
            if row[0] == id_obat:
                found = True
                harga = float(row[2])
                stok = int(row[3])
                if jumlah_beli > stok:
                    print("Stok tidak cukup.")
                    return saldo
                total_harga = harga * jumlah_beli
                stok -= jumlah_beli
                row[3] = str(stok)
                rows.append(row)

    if not found:
        print("ID obat tidak ditemukan.")
        return saldo

    if saldo < total_harga:
        print("Saldo tidak cukup.")
        return saldo

    saldo -= total_harga
    update_saldo(username, saldo)

    with open(dataharga_obat, mode="w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow(header)
        writer.writerows(rows)

```

```

        print(f"Pembelian berhasil! Total harga: Rp
{total_harga:,.0f}")
        print(f"Sisa saldo: Rp {saldo:,.0f}")
        return saldo

def menu_user(username, saldo):
    while True:
        clear()
        print("\n+=====+
=====+")
        print("|                               Menu
User                               |")
        print("+=====+
=====+")
        print("| [1]. Lihat Daftar
Obat                               |")
        print("| [2]. Isi
Saldo                               |")
        print("| [3]. Beli
Obat                               |")
        print("| [4]. Lihat
Saldo                               |")
        print("| [0].
Keluar                               |")
        print("+=====+
=====+")
        pilihan = input("Masukkan pilihan (0-4): ")

        if pilihan == "1":
            lihat()
        elif pilihan == "2":
            saldo = isi_saldo(username, saldo)
        elif pilihan == "3":
            saldo = beli_obat(username, saldo)
        elif pilihan == "4":
            lihat_saldo(username)
        elif pilihan == "0":
            print("Logout berhasil.")
            break
        else:
            print("Masukkan angka yang sesuai.")
            input("\nTekan Enter untuk lanjut...")

def menu_admin():
    clear()
    while True:
        clear()

```

```

        print("\n+=====+")
=====+
        print(" |                               Menu
Admin                               |")
        print("+=====+")
=====+
        print(" | [1]. Tambahkan
Obat                               |")
        print(" | [2]. Lihat Daftar
Obat                               |")
        print(" | [3]. Updet daftar
Obat                               |")
        print(" | [4]. Hapus
Obat                               |")
        print(" | [0].
Keluar                             |")
        print("+=====+")
=====+
        pilihan = input("Masukkan pilihan 0-4: ")

        if pilihan == "1":
            tambahkan()
        elif pilihan == "2":
            lihat()
        elif pilihan == "3":
            update()
        elif pilihan == "4":
            hapus()
        elif pilihan == "0":
            print("Logout berhasil.")
            break
        else:
            print("Masukkan angka yang sesuai.")
            input("\nTekan Enter untuk lanjut...")

def regis():
    clear()
    while True:
        clear()
        print("\n+=====+")
=====+
        print(" |                               Menu
login                               |")
        print("+=====+")
=====+
        print(" | [1].
Registrar                           |")

```



```

        print("| [2].
login                                                                    |")
        print("| [3].
Kelaur                                                                    |")
        print("+=====
=====+")
        masuk = input("Masukkan pilihan 1-3: ")
        if masuk == "1":
            register()
        elif masuk == "2":
            login()
        elif masuk == "3":
            print("Program selesai.")
            break
        else:
            print("Masukkan angka yang sesuai 1-3")

print(regis())

```

BAB IV

PENUTUP

4.1 Kesimpulan

Dari hasil pembuatan dan pengujian program Sistem Manajemen Apotek, dapat disimpulkan bahwa sistem ini mampu membantu dalam pengelolaan data obat, transaksi penjualan, serta pencatatan stok dengan lebih efisien dan terorganisir. Program ini dirancang untuk mempermudah kinerja apotek dalam mengatur ketersediaan obat, mencatat transaksi penjualan, serta meminimalisir kesalahan pencatatan yang sering terjadi saat dilakukan secara manual. Dengan adanya sistem ini, proses administrasi di apotek menjadi lebih cepat, akurat, dan mudah dipantau.

4.2 Saran

Agar sistem ini dapat berkembang lebih baik di masa mendatang, beberapa saran yang dapat diberikan antara lain :

1. Penambahan fitur login multiuser, agar setiap pelanggan memiliki akses dan hak yang berbeda.
2. Integrasi dengan database online atau cloud, sehingga data dapat diakses secara real-time dan aman,

DAFTAR PUSTAKA

<https://github.com/PA-DasPro-C-Kelompok-2/PA-DasPro-C-Kelompok-2>

LAMPIRAN

Lampiran 1 : Tabel Kontribusi

Nama	Kontribusi	Bagian
FITRI YANTI 23091160016	Flowchart, Pengecekan	1. Referensi konsep program 2. Readme.md 3. Pengecekan Flowchart 4. Merevisi Flowchart
MUHAMMAD RISKY ALPIANUR 2509116101	Coding, Flowchart, pengembangan	1. Logika program 2. Flowchart program 3. Coding 4. Mencari bug dan <i>error</i>
SATRIA ALFIANDI RAMADHAN AKBAR 2509116089	Konsep, Laporan, Pengecekan	1. Konsep dan penyusunan laporan 2. Menjelaskan program di Laporan 3. Pengecekan Laporan 4. Finishing Laporan

