

**PRAKTIKUM DASAR-DASAR PEMROGRAMAN**  
**PENILAIAN AKHIR**  
**“SISTEM MANAJEMEN APOTEK”**



**Disusun Oleh:**

**NAMA KELOMPOK**

**NYOMAN ARINI TRIRAHAYU (2309116002)**

**FITRI YANTI (2309116016)**

**RIFQI HADI WIJAYA (2309116042)**

**Asisten Laboratorium:**

**Awang Muhammad Novandra Arssaputra**  
**(2209116040)**

**Muhammad Fauzan Gifari Dzul Fahmi**  
**(2209116042)**

**Novianti Safitri**  
**(2209116004)**

**Nur Avivah**  
**(2209116010)**

**PROGRAM STUDI SISTEM INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS MULAWARMAN**

**2021**

## **Kata Pengantar**

Puji dan syukur atas berkat hadirat Tuhan Yang Maha Esa karena atas Rahmat dan hidayah-Nya, kami dapat menyelesaikan tugas Penilaian Akhir ini yang berjudul "SISTEM MANAJEMEN APOTEK" dengan baik. Laporan ini disusun untuk memenuhi tugas mata kuliah Dasar-Dasar Pemrograman di Universitas Mulawarman Fakultas Teknik.

Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu pembuatan laporan ini, diantaranya:

1. Para Asisten Laboratorium Sistem Informasi A 2022,
2. Seluruh teman-teman yang banyak membantu dan juga memberi semangat serta dukungan,
3. Serta pihak-pihak yang tidak kami sebutkan satu persatu yang telah membantu dalam pembuatan tugas penilaian akhir kelompok kami.

Karenanya kami dapat menyelesaikan program dan juga laporan ini dengan baik. Berkat ilmu yang kami terima dari yang di ajarkan oleh asisten laboratorium pada beberapa pertemuan yang telah dilakukan.

Kami menyadari bahwa hasil laporan ini masih banyak kekurangan dan kelemahannya, baik dalam isi maupun sistematikanya. Oleh karena itu, kami sangat mengharapkan kritik dan saran untuk menyempurnakan laporan ini. Semoga hasil penulisan laporan ini dapat memberikan manfaat, khususnya bagi penulis dan pembaca.

Samarinda, 3 November 2023

Penyusun

## Daftar Isi

<b>Kata Pengantar .....</b>	<b>ii</b>
<b>Daftar Isi.....</b>	<b>iii</b>
<b>Daftar Gambar.....</b>	<b>iv</b>
<b>Daftar Tabel .....</b>	<b>vi</b>
<b>BAB I.....</b>	<b>7</b>
1.1 Deskripsi Masalah .....	7
1.2 Rumusan Masalah.....	7
1.3 Batasan Masalah .....	9
1.4 Tujuan.....	9
1.5 Manfaat.....	9
<b>BAB II .....</b>	<b>10</b>
2.1 Analisis Program .....	10
2.2 Flowchart .....	10
<b>BAB III.....</b>	<b>14</b>
3.1 Implementasi Program.....	14
3.2 Alur Program .....	23
3.3 Tampilan Program .....	26
3.4 Source Code.....	36
<b>BAB IV.....</b>	<b>56</b>
4.1 Kesimpulan.....	56
4.2 Saran .....	56
<b>Daftar Pustaka .....</b>	<b>57</b>
<b>Lampiran .....</b>	<b>58</b>

## Daftar Gambar

<b>Gambar 2.1</b> Flowchart 1.1 .....	10
<b>Gambar 2.2</b> Flowchart 1.2 .....	11
<b>Gambar 2.3</b> Flowchart 1.3 .....	12
<b>Gambar 2.4</b> Flowchart 1.4 .....	13
<b>Gambar 3.1</b> Implementasi: Python .....	14
<b>Gambar 3.2</b> Implementasi: Dictionary akun admin.....	15
<b>Gambar 3.3</b> Implementasi: While True Loop untuk mengulang login.....	15
<b>Gambar 3.4</b> Implementasi: While dengan satu kondisi.....	15
<b>Gambar 3.5</b> Implementasi: Library PrettyTable.....	16
<b>Gambar 3.6</b> Implementasi: Table PrettyTable.....	16
<b>Gambar 3.7</b> Implementasi: Read Data Produk.....	16
<b>Gambar 3.8</b> Implementasi: Create Data Produk.....	17
<b>Gambar 3.9</b> Implementasi: Update Data Produk .....	17
<b>Gambar 3.10</b> Implementasi: Delete Produk .....	18
<b>Gambar 3.11</b> Implementasi: Data akun admin.....	18
<b>Gambar 3.12</b> Implementasi: Data akun pembeli .....	19
<b>Gambar 3.13</b> Implementasi: Pilihan menu login admin dan pembeli.....	19
<b>Gambar 3.14</b> Implementasi: Proses login admin .....	20
<b>Gambar 3.15</b> Implementasi: Proses login pembeli .....	20
<b>Gambar 3.16</b> Implementasi: pwinput .....	21
<b>Gambar 3.17</b> Implementasi: pwinput pada input login admin .....	21
<b>Gambar 3.18</b> Implementasi: pwinput pada input login pembeli .....	21
<b>Gambar 3.19</b> Implementasi: json.....	21
<b>Gambar 3.20</b> Implementasi: json pada menu.....	21
<b>Gambar 3.21</b> Implementasi: json pada akun admin.....	22
<b>Gambar 3.22</b> Implementasi: json pada akun pembeli.....	22
<b>Gambar 3.23</b> Implementasi: txt pada cetak struk.....	22
<b>Gambar 3.24</b> Alur Program: Menu utama 1.1 .....	23

<b>Gambar 3.26</b> Alur Program: Menu utama 1.2 .....	24
<b>Gambar 3.27</b> Alur Program: Menu utama 1.3 .....	24
<b>Gambar 3.28</b> Alur Program: Menu utama 1.4.....	25
<b>Gambar 3.29</b> Alur Program: Menu utama 1.5 .....	25
<b>Gambar 3.30</b> Alur Program: Menu Admin .....	26
<b>Gambar 3.31</b> Alur Program: Menu User .....	26
<b>Gambar 3.32</b> Tampilan Program: Tampilan Awal .....	26
<b>Gambar 3.33</b> Tampilan Program: Login Admin .....	26
<b>Gambar 3.34</b> Tampilan Program: Login Pembeli .....	27
<b>Gambar 3.35</b> Tampilan Program: Registrasi Admin .....	27
<b>Gambar 3.36</b> Tampilan Program: Registrasi Pembeli .....	27
<b>Gambar 3.37</b> Tampilan Program: Menu Admin .....	28
<b>Gambar 3.38</b> Tampilan Program: Lihat Produk.....	28
<b>Gambar 3.39</b> Tampilan Program: Tambah data produk .....	29
<b>Gambar 3.40</b> Tampilan Program: Ubah data produk .....	30
<b>Gambar 3.41</b> Tampilan Program: Hapus data produk .....	31
<b>Gambar 3.42</b> Tampilan Program: Kembali ke menu admin .....	31
<b>Gambar 3.43</b> Tampilan Program: Menu pembeli .....	32
<b>Gambar 3.44</b> Tampilan Program: Menu lihat dan beli obat .....	32
<b>Gambar 3.45</b> Tampilan Program: Menu isi saldo .....	33
<b>Gambar 3.46</b> Tampilan Program: menu tampilkan saldo .....	33
<b>Gambar 3.47</b> Tampilan Program: Menu search produk .....	33
<b>Gambar 3.48</b> Tampilan Program: Menu Sorting produk .....	33
<b>Gambar 3.49</b> Tampilan Program: Sorting berdasarkan nama .....	34
<b>Gambar 3.50</b> Tampilan Program: Sorting berdasarkan kategori.....	34
<b>Gambar 3.51</b> Tampilan Program: Sorting berdasarkan Harga.....	34
<b>Gambar 3.52</b> Tampilan Program: sorting kembali .....	35
<b>Gambar 3.53</b> Tampilan Program: Tampilan Akhir program.....	35

## Daftar Tabel

<b>Tabel 3.1</b> Source Code .....	55
<b>Tabel 1.1</b> Tabel Kontribusi .....	56

# BAB I

## Pendahuluan

### 1.1 Deskripsi Masalah

Di era globalisasi saat ini, perkembangan teknologi dan informasi sudah mengalami kemajuan yang sangat pesat. Hampir pada setiap aspek kegiatan manusia, baik yang dilakukan secara pribadi maupun bersama-sama selalu mempunyai hubungan dengan teknologi. Kemajuannya yang pesat ini sangatlah membantu kehidupan manusia, bahkan dengan kemajuan teknologi bisa menciptakan peluang untuk menghasilkan pundi-pundi rupiah tanpa harus meninggalkan rumah sekalipun.

Pada zaman sekarang, beberapa orang menjalankan bisnisnya tanpa memiliki toko offline. Banyak bisnis yang dijalankan hanya melalui media sosial maupun e-commerce. Menjalankan bisnis secara online merupakan salah satu cara untuk mengembangkan bisnis agar mencapai jarak yang tidak terjangkau oleh toko offline.

CRUD merupakan sebuah singkatan dari *create*, *read*, *update* dan *delete*. *Create* berfungsi untuk membuat record baru dalam sebuah database, *read* berfungsinya seperti search yang dimana dalam fungsi ini bisa melakukan pencarian dan juga mengambil data dari tabel serta membaca nilai dari data tersebut, *update* digunakan untuk memodifikasi record yang sudah dibuat sebelumnya dan tersimpan dalam database. Dan *delete* yang digunakan untuk menghapus sebuah record data yang sudah tidak dibutuhkan.

CRUD dirancang sebagai empat metode fungsi dasar untuk meningkatkan penyimpanan dengan database. Dalam pengembangan perangkat lunak modern, CRUD telah melampaui asal-usulnya sebagai fungsi dasar dari sebuah basis data dan sekarang memetakan dirinya untuk prinsip-prinsip desain untuk aplikasi dinamis.

### 1.2 Rumusan Masalah

Berdasarkan deskripsi masalah diatas, didapatkan rumusan masalah sebagai berikut:

1. Bagaimana cara membuat program python?
2. Bagaimana cara pemilik toko dalam menjalankan usahanya secara online dengan mudah?
3. Bagaimana cara mempermudah transaksi dengan menggunakan e-money?

### **1.3 Batasan Masalah**

Pembatasan suatu masalah digunakan untuk menghindari adanya penyimpangan maupun pelebaran pokok masalah agar laporan praktikum lebih terarah dan memudahkan dalam pembahasan sehingga tujuan praktikum akan tercapai. Beberapa batasan masalah dalam praktikum ini adalah sebagai berikut:

1. Program yang digunakan dalam pembuatan laporan ini dengan menggunakan python sebagai bahasa pemrograman yang digunakan, dengan menggunakan JSON sebagai database,
2. Di dalam program ini terdapat program yang dibuat untuk transaksi suatu usaha atau toko, dimana transaksi tersebut dapat membuat, menampilkan, mengedit, dan menghapus suatu item produk yang di jual.

### **1.4 Tujuan**

Adapun dari penulisan laporan ini, antara lain:

1. Untuk mengetahui cara membuat program python,
2. Untuk mempermudah pemilik toko dalam menjalankan usahanya,
3. Untuk mempermudah transaksi dengan menggunakan e-money,

### **1.5 Manfaat**

Adapun beberapa manfaat dalam penulisan laporan ini, antara lain:

1. Bagi Penulis diharapkan penulisan laporan akhir ini dapat memahami dan menambah pengetahuan tentang pembuatan program transaksi CRUD dengan menggunakan penyimpanan dinamis Json. Dan memenuhi tugas dari praktikum akhir Dasar - Dasar Pemrograman.
2. Bagi pembaca diharapkan hasil penulisan ini dapat bermanfaat untuk menambah pengetahuan dan dapat menjadi referensi atau bahan masukan dalam penelitian serupa pada waktu yang akan datang.



## BAB II

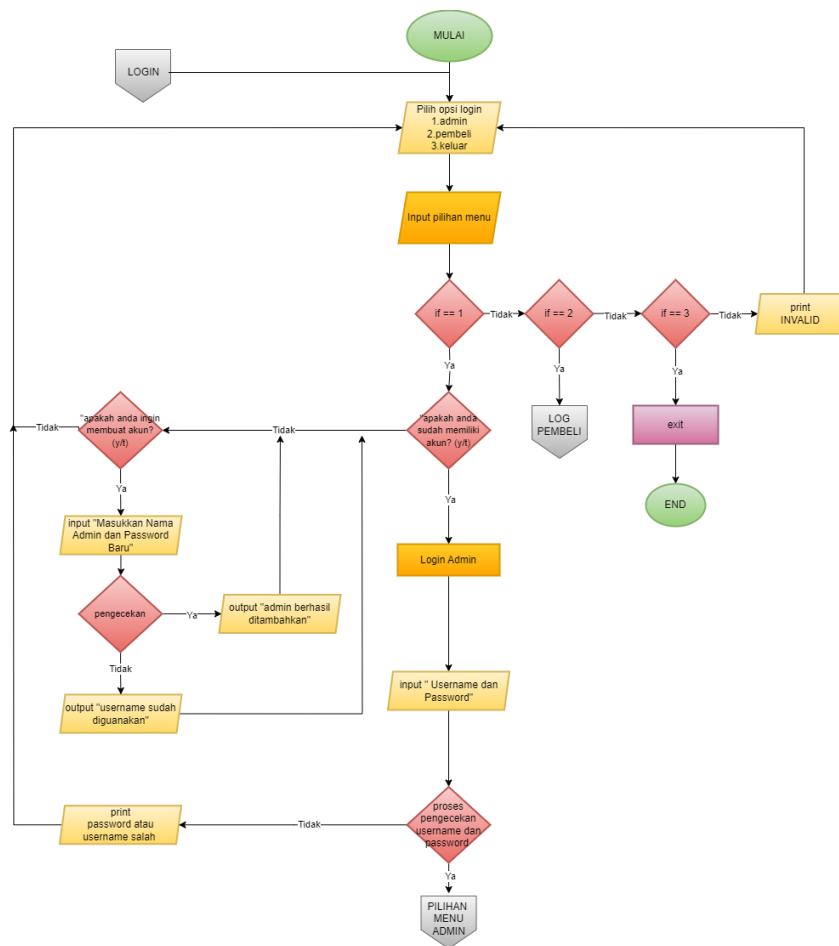
### Perancangan

#### 2.1 Analisis Program

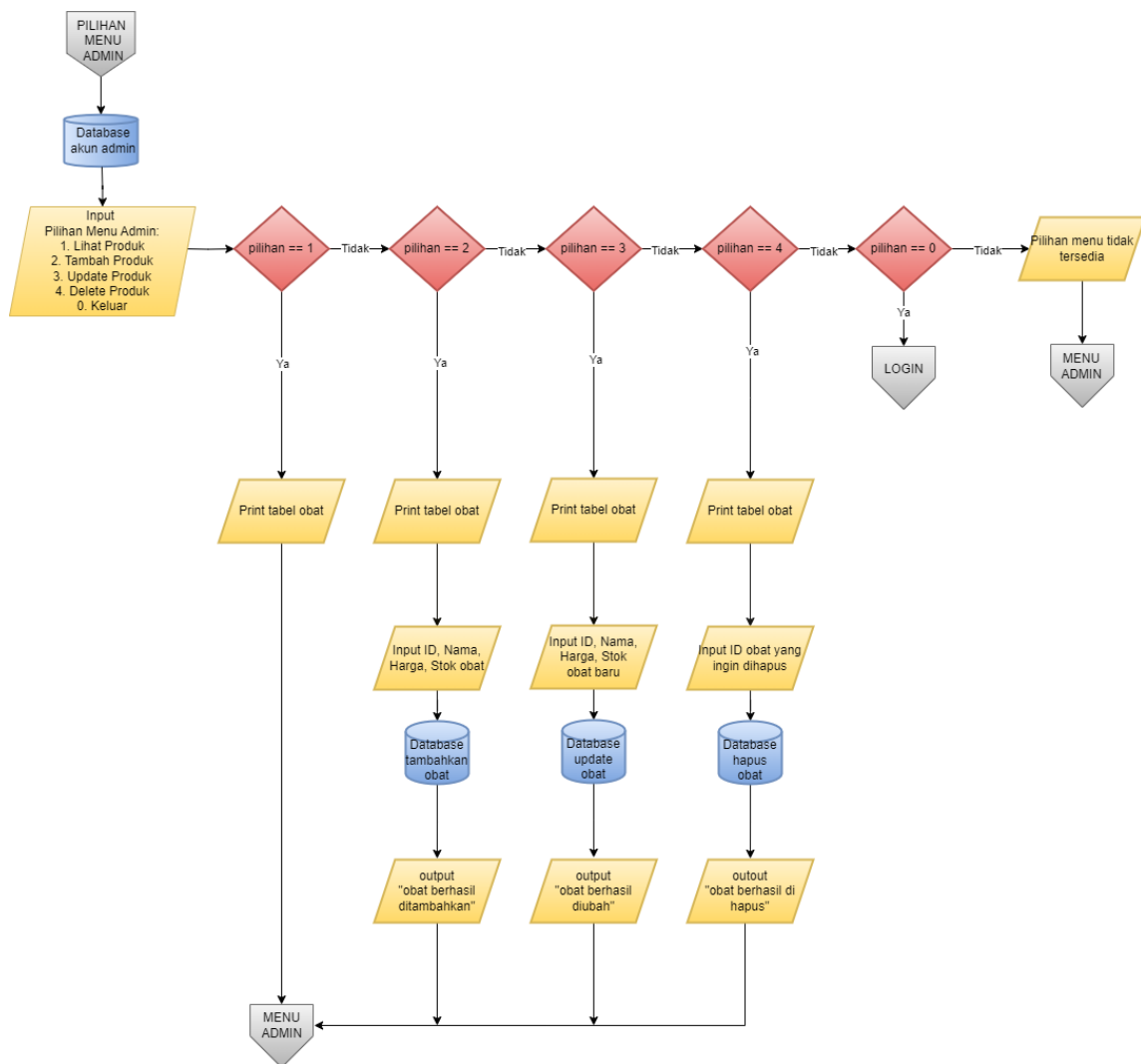
Analisis Program dibuat untuk mengetahui cara membuat program python, mempermudah pemilik toko dalam menjalankan usahanya, dan mempermudah transaksi dengan menggunakan e-money.

Program ini mempunyai 4 library yaitu, library *import os*. library *import os* berfungsi untuk berinteraksi dengan sistem operasi, library *import prettytable* form *PrettyTable*. Library *import prettytable* form *PrettyTable* berfungsi untuk mencetak atau membuat tabel rapi dan mudah dibaca dalam output, library *import pwinput* form *pwinput*. Library *import pwinput* form *pwinput* berfungsi menampilkan \*\*\* saat menginput kata sandi, library *import json*. Library *Import json* berfungsi sebagai database penyimpanan barang, akun admin dan akun pelanggan

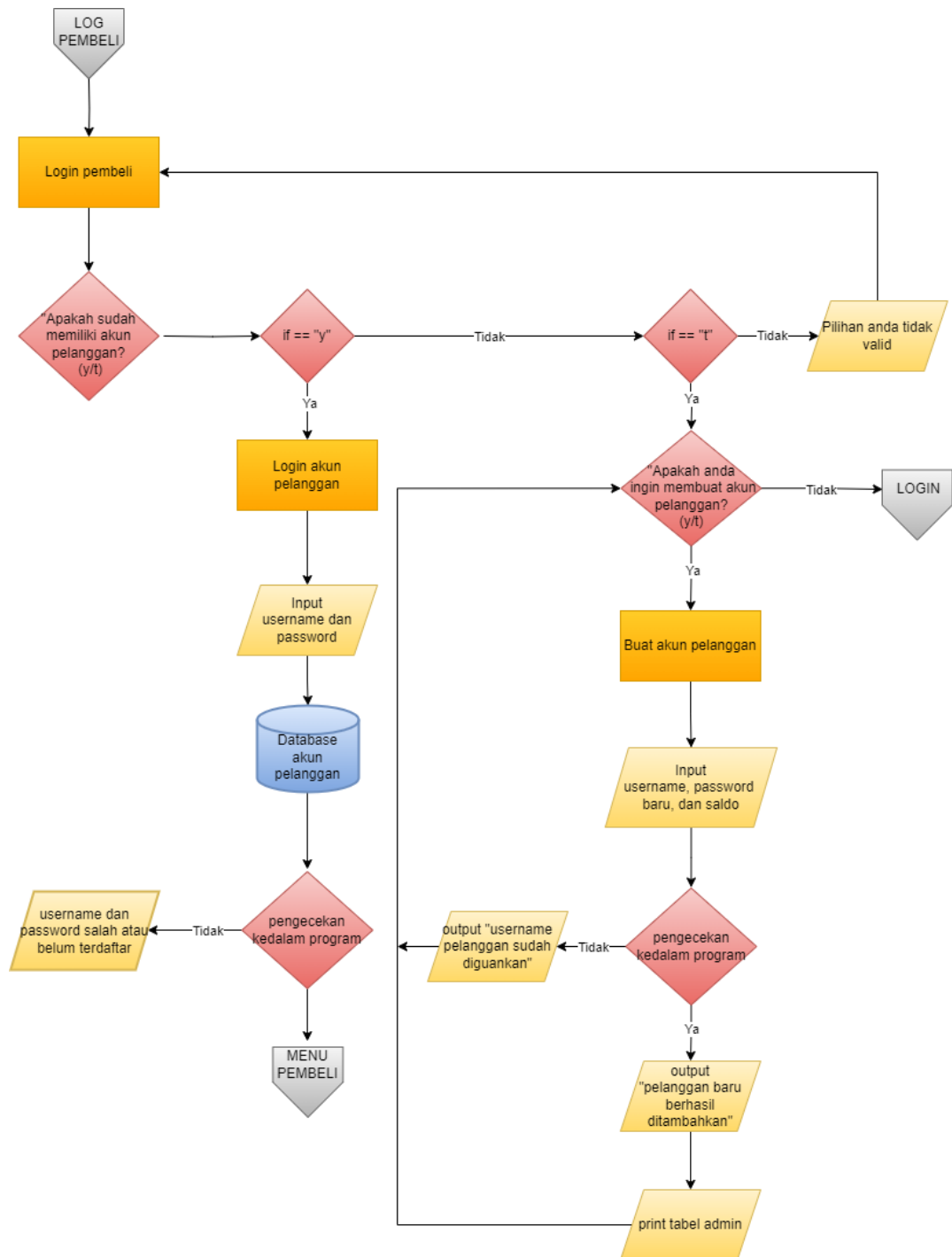
#### 2.2 Flowchart



Gambar 2.1 Flowchart 1.1



**Gambar 2.2** Flowchart 1.2



**Gambar 2.3** Flowchart 1.3



## BAB III

### Hasil dan Pembahasan

#### 3.1 Implementasi Program

##### A. Bahasa Program Python

Python adalah salah satu bahasa pemrograman yang banyak digunakan saat ini, karena python dapat digunakan dalam banyak hal, seperti membangun website aplikasi dan program perusahaan. Python dirancang oleh Guido van Rossum pada tahun 1991 dan dikembangkan oleh Python Software Foundation.

Python termasuk bahasa pemrograman yang populer karena mudah untuk dipelajari dan digunakan oleh orang awam. Dan Python tergolong bahasa pemrograman tingkat tinggi. Maksudnya, sintaks Python didesain semirip mungkin dengan bahasa manusia. Sehingga, belajar Python pemula jadi lebih mudah dan cepat.

```
# fungsi untuk menu admin
def menuAdmin():
    while True:
        print(
            f"""
            ~~~~~| SELAMAT DATANG |~~~~~
            ~~~~~| Apotek SI'A Kel 10 |~~~~~
            +-----+
            |      Menu Admin Apotek      |
            +-----+
            | [1.] Lihat Produk             |
            | [2.] Tambah Produk            |
            | [3.] Update Produk            |
            | [4.] Hapus Produk             |
            | [0.] Keluar                   |
            +-----+"""
        )
```

Gambar 3.1 Implementasi: Python

##### B. Penyimpanan Data Tipe Dictionary

Dictionary merupakan tipe data pada Python yang berfungsi untuk menyimpan kumpulan data atau nilai, yang setiap urutanya berisi key dan value. Jika biasanya kita ingin mengakses nilai pada list menggunakan indeks, di dictionary ini kita perlu kata kunci (key) untuk mengakses nilainya.

```

1  [
2      {
3          "username": "rifqi",
4          "password": "123"
5      },
6      {
7          "username": "arini",
8          "password": "123"
9      },
10     {
11         "username": "arin",
12         "password": "arin"
13     },
14     {
15         "username": "nyoman",
16         "password": "123"
17     },
18     {
19         "username": "arlen",
20         "password": "111"
21     }
22 ]

```

**Gambar 3.2** Implementasi: Dictionary akun Admin

### C. Percabangan While True Loop

While loop digunakan untuk mengeksekusi blok kode secara berulang sampai kondisi boolean yang diberikan menjadi False.

```

def regis():
    while True:
        try:
            print("Silahkan Registrasi Akun Anda^^")
            username = input("Masukkan nama admin baru: ")
            password = pwinput.pwinput("Masukkan password admin baru: ")
            if username == "":
                print("Username tidak boleh kosong")
                continue
            if password == "":
                print("password tidak boleh kosong")
                continue
            for i in data_admin:
                if username == i["username"]:
                    print("username sudah tersedia, cari username yang lain!")
                    return
            data_admin.append({"username": username, "password": password})
            tabelAdmin.add_row([username, password])
            with open(adminjson, "w") as sn:
                json.dump(data_admin, sn, indent=4)
            print(f"Admin {username} berhasil ditambahkan.")
            print(tabelAdmin)
            break
        except ValueError:
            print("Masukkan input yang benar")
        except KeyboardInterrupt:
            print("\ninput yang anda masukkan tidak valid")
        except Exception as e:
            print("Gagal mendaftarkan admin: ", (e))

```

**Gambar 3.3** Implementasi: While True Loop untuk mengulang login

```

while True:
    buatadmin = input("Apakah Ingin Membuat Akun Admin? y/t: ")
    if buatadmin == "y":
        regis()
    elif buatadmin == "t":
        break
    else:
        print("Pilihan Anda Tidak Valid")
else:
    print("silahkan input pilihan y/t")
except FileNotFoundError:
    print("Terjadi Kesalahan Saat Membaca File.")
except ValueError:
    print("Silahkan Coba Lagi Nanti.")
except KeyboardInterrupt:
    print("\ninput yang anda masukkan tidak valid")

```

**Gambar 3.4** Implementasi: While dengan satu kondisi

#### D. Tabel data dengan menggunakan PrettyTable

Di program ini, PrettyTable digunakan untuk membuat tabel menu agar pengguna lebih mudah dan nyaman dalam melihat menu. Tetapi sebelum menggunakannya kita harus menginstal dulu, dengan cara menyalin pip install PrettyTable dan menempelkannya pada command prompt (CMD) dan setelah itu jalankan.

```
from prettytable import PrettyTable
```

Gambar 3.5 Implementasi: Library PrettyTable

```
*****
                                APOTEK SI'23 KEL 10
*****
```

ID	Nama Obat	Kategori Obat	Harga Obat	Stok Obat
1	Paracetamol	Pilek	5000	20
2	Brufen	Pilek	7000	15
3	Dafalgan	Batuk	8000	20
4	Promag	Mag	5000	25
5	Maxtro	Lambung	20000	12

Gambar 3.6 Implementasi: Table PrettyTable

#### E. CRUD (Create, Read, Update, Delete)

- Lihat Produk

```
# fungsi read produk
def read_produk():
    global tabel_produk
    tabel_produk = PrettyTable()
    tabel_produk.field_names = [
        "ID",
        "Nama Obat",
        "Kategori Obat",
        "Harga Obat",
        "Stok Obat",
    ]
    try:
        for x in data_produk:
            tabel_produk.add_row(
                [x["id"], x["nama"], x["kategori"], x["harga"], x["stok"]]
            )
        print(tabel_produk)
    except FileNotFoundError:
        print("File tidak ditemukan.")
    except Exception as e:
        print(f"Terjadi kesalahan: {e}")
    except KeyboardInterrupt:
        print("\nMasukkan inputan dengan benar")
```

Gambar 3.7 Implementasi: Read Data Produk

- Tambah Produk

```
# fungsi untuk create produk
def create_produk():
    while True:
        try:
            tambahId = int(input("tambahkan id baru: "))
            tambahObat = input("isi nama obatnya: ")
            tambahKategori = input("isi kategori obat: ")
            tambahHarga = int(input("isi harganya: "))
            tambahStok = int(input("tambahkan stok: "))
            if tambahObat == "":
                print("nama obat tidak boleh kosong")
                continue
            if tambahKategori == "":
                print("kategori obat tidak boleh kosong")
                continue
            for x in data_produk:
                if tambahId == x["id"]:
                    print("id sudah tersedia, buat id yang lain!")
                    return
                continue
            data_produk.append(
                {
                    "id": tambahId,
                    "nama": tambahObat,
                    "kategori": tambahKategori,
                    "harga": tambahHarga,
                    "stok": tambahStok,
                }
            )
            tabel_produk.add_row(
                [tambahId, tambahObat, tambahKategori, tambahHarga, tambahStok]
            )
            print("obat berhasil ditambahkan")
            savejsonProduk()
            lagi = input("apakah anda ingin menambah obat lagi? (y/n)")
            if lagi == "y":
                create_produk()
            elif lagi == "n":
                break
            else:
                print("input y/n")
        except ValueError:
            print("Masukkan input yang benar")
        except Exception as e:
            print("Error:", str(e))
        except KeyboardInterrupt:
            print("\nMasukkan inputan dengan benar")
```

**Gambar 3.8** Implementasi: Crete Data Produk

- Update Produk

```
def update_produk(id, nama=None, kategori=None, harga=None, stok=None):
    read_produk()
    try:
        print("Silahkan Update Daftar Obat")
        id = int(input("Masukkan ID Obat yang akan diperbarui: "))
        nama = str(input("Masukkan Nama Obat baru: "))
        kategori = str(input("Masukkan Kategori Obat Baru: "))
        harga = int(input("Masukkan Harga Obat baru: "))
        stok = int(input("Masukkan Stok Obat baru: "))
        for item in data_produk:
            if item["id"] == id:
                if nama:
                    item["nama"] = nama
                if kategori:
                    item["kategori"] = kategori
                if harga:
                    item["harga"] = harga
                if stok:
                    item["stok"] = stok
                savejsonProduk()
                print("Produk berhasil di Update")
                return
        print("Produk tidak ditemukan")
    except ValueError:
        print("\nMasukkan input yang benar")
    except Exception as e:
        print("\nError:", str(e))
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")
```

**Gambar 3.9** Implementasi: Update Data Produk



- Delete Produk

```
# fungsi delete produk
def delete_produk(data_produk):
    try:
        hapusId = int(input("masukkan id obat yang akan dihapus: "))
        index = -1
        for i in range(0, len(data_produk)):
            id = data_produk[i]
            if hapusId == id["id"]:
                index = i
                break
        if index == -1:
            print("id obat tidak ditemukan")
        else:
            del data_produk[index]
            print("berhasil menghapus obat")
            savejsonProduk()
    except ValueError:
        print("Masukkan input yang benar")
    except Exception as e:
        print("Error:", str(e))
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")
```

**Gambar 3.10** Implementasi: Delete Data Produk

## F. Akses Admin dan User

Username dan password admin terletak di json.

```
1  [
2      {
3          "username": "rifqi",
4          "password": "123"
5      },
6      {
7          "username": "arini",
8          "password": "123"
9      },
10     {
11         "username": "arin",
12         "password": "arin"
13     },
14     {
15         "username": "nyoman",
16         "password": "123"
17     },
18     {
19         "username": "arien",
20         "password": "111"
21     }
22 ]
```

**Gambar 3.11** Implementasi: Data akun admin

Username dan password pembeli terletak di json.

```
1  [
2      {
3          "username": "rifqi",
4          "password": "123",
5          "saldo": 313000
6      },
7      {
8          "username": "trirahayu",
9          "password": "123",
10         "saldo": 40000
11     },
12     {
13         "username": "fitriyanti",
14         "password": "789",
15         "saldo": 30000
16     },
17     {
18         "username": "arini",
19         "password": "arini",
20         "saldo": 37000
21     }
22 ]
```

**Gambar 3.12** Implementasi: Data akun pembeli

```
# fungsi program utama
def main():
    while True:
        print(
            f"""
            ~~~~~| SELAMAT DATANG |~~~~~
            ~~~~~| Apotek SI'A Kel 10 |~~~~~
            +-----+
            |          MASUK SEBAGAI          |
            +-----+
            |          [1.] Admin              |
            |          [2.] Pembeli            |
            |          [0.] Keluar              |
            +-----+"""
        )
        try:
            pilihan = int(input("Masuk Sebagai: "))
```

**Gambar 3.13** Implementasi: Pilihan menu login Admin dan Pembeli

## G. Login

```
def login_admin():
    try:
        admin_regis = input("Apakah Sudah Memiliki Akun? y/t: ")
        if admin_regis == "y":
            print("~~~~~Silahkan Login~~~~~")
            print("Masukkan Username dan Password dengan Benar!")
            username = input("Masukkan username: ")
            pw = pwinput.pwinput("Masukkan password: ")
            os.system("cls")
            for i in data_admin:
                if i["username"] == username:
                    if i["password"] == pw:
                        print("Login Berhasil.")
                        print(f"Hallo Admin: {username}")
                        menuAdmin()
                        return
                    else:
                        print("password atau username anda masukkan salah")
                        return
            else:
                print("password atau username anda tidak terdaftar")
                return
        elif admin_regis == "t":
            while True:
                buatadmin = input("Apakah Ingin Membuat Akun Admin? y/t: ")
                if buatadmin == "y":
                    regis()
                elif buatadmin == "t":
                    break
            else:
                print("Pilihan Anda Tidak Valid")
        else:
            print("silahkan input pilihan y/t")
    except FileNotFoundError:
        print("Terjadi Kesalahan Saat Membaca File.")
    except ValueError:
        print("Silahkan Coba Lagi Nanti.")
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")
```

Gambar 3.14 Implementasi: Proses login admin

```
def loginpelanggan():
    global saldo_pelanggan
    global inUsername
    print("Selamat Datang Di Apotek SI'A Kel 10")
    try:
        pilihMember = input("apakah sudah memiliki akun pelanggan? y/t: ")
        if pilihMember == "y":
            inUsername = input("masukkan username anda: ")
            inpass = pwinput.pwinput("masukkan password anda: ")
            for x in data_pelanggan:
                if x["username"] == inUsername:
                    for y in data_pelanggan:
                        if y["password"] == inpass:
                            saldo_pelanggan = x["saldo"]
                            os.system("cls")
                            print("Login Berhasil.")
                            print(f"Hallo Pelanggan: {inUsername}")
                            menu_pelanggan()
                            return True
            print("username atau password salah atau tidak terdaftar")
        elif pilihMember == "t":
            while True:
                pilihMember = input("Apakah Ingin Membuat Akun Pelanggan? y/t: ")
                if pilihMember == "y":
                    tambah_pelanggan()
                elif pilihMember == "t":
                    break
            else:
                print("Pilihan Anda Tidak Valid")
    except ValueError:
        print("\nMasukkan input yang benar")
    except Exception as e:
        print(f"Terjadi kesalahan: {e}")
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")
```

Gambar 3.15 Implementasi: Proses login pembeli

## H. Modul pwininput

Di program ini, pwininput digunakan supaya password asli yang diketik tidak terlihat, dan isi password diganti dengan simbol bintang '\*\*\*\*\*'.

```
import pwininput
```

**Gambar 3.16** Implementasi: Pwininput

```
pw = pwininput.pwininput("Masukkan password: ")
```

**Gambar 3.17** Implementasi: Pwininput pada login admin

```
inpass = pwininput.pwininput("masukkan password anda: ")
```

**Gambar 3.18** Implementasi: Pwininput pada login pembeli

## I. Penyimpanan Dinamis (JSON)

JSON adalah penyimpanan dinamis yang berbasis teks dan berfungsi untuk menyimpan dan mentransfer data.

```
import json
```

**Gambar 3.19** Implementasi: JSON

```
produkjson = f"{ROOD_DIR}/projectAkhir/crudpa.json"
try:
    with open(produkjson, "r") as jsonproduk:
        data_produk = json.loads(jsonproduk.read())
except FileNotFoundError:
    print("File tidak ditemukan.")
    data_produk = []
except Exception as e:
    print(f"Terjadi kesalahan: {e}")
except KeyboardInterrupt:
    print("\nMasukkan inputan dengan benar")
```

**Gambar 3.20** Implementasi: JSON pada menu

```

tabelAdmin = PrettyTable()
tabelAdmin.field_names = ["nama admin", "password admin"]
adminjson = f"{ROOD_DIR}/projectAkhir/paluginadmin.json"
try:
    with open(adminjson, "r") as jsonadmin:
        data_admin = json.loads(jsonadmin.read())

        for i in data_admin:
            tabelAdmin.add_row([i["username"], i["password"]])
except FileNotFoundError:
    print(f"File not found: {adminjson}")
except KeyError:
    print("Invalid key in the JSON file.")
except json.JSONDecodeError:
    print("Invalid JSON file.")
except Exception as e:
    print(f"Unexpected error: {e}")
except KeyboardInterrupt:
    print("\nMasukkan inputan dengan benar")

```

**Gambar 3.21** Implementasi: JSON pada akun admin

```

tabelPelanggan = PrettyTable()
tabelPelanggan.field_names = ["username", "password", "saldo"]
pelangganjson = f"{ROOD_DIR}/projectAkhir/paluginpelanggan.json"
try:
    with open(pelangganjson, "r") as jsonpelanggan:
        data_pelanggan = json.loads(jsonpelanggan.read())

        for i in data_pelanggan:
            tabelPelanggan.add_row([i["username"], i["password"], i["saldo"]])
except FileNotFoundError:
    print(f"File not found: {pelangganjson}")
except KeyError:
    print("\nInvalid key in the JSON file.")
except json.JSONDecodeError:
    print("\nInvalid JSON file.")
except Exception as e:
    print(f"\nUnexpected error: {e}")
except KeyboardInterrupt:
    print("\nMasukkan inputan dengan benar")

```

**Gambar 3.22** Implementasi: JAON pada login pembeli

## J. Penyimpanan Dinamis (TXT)

TXT adalah penyimpanan dinamis yang berbasis teks dan berfungsi untuk menyimpan dan mentransfer data.

```

≡ struk.txt
1
2 *****
3 |      |      |      |      |      |      |      |
4 |      |      |      |      |      |      |      |
5 Tanggal Transaksi: 2023-11-03 13:58:08
6 Nama Pelanggan: arini
7 +-----+-----+-----+-----+-----+-----+
8 | ID Obat | Nama Obat | Kategori | Kuantitas | Harga Satuan | Total Harga |
9 +-----+-----+-----+-----+-----+-----+
10 |    1    | Paracetamol | Pilek    |    3      |    5000      |    15000    |
11 +-----+-----+-----+-----+-----+-----+
12

```

**Gambar 3.23** Implementasi: TXT pada cetak struk

### 3.2 Alur Program

Berikut ini adalah alur program bagian input dengan deskripsi sebagai berikut:

#### 1. Menu Utama



```
# fungsi program utama
def main():
    while True:
        print(
            f"""
            ~~~~~| SELAMAT DATANG |~~~~~
            ~~~~~| Apotek SI'A Kel 10 |~~~~~
            +-----+
            |      MASUK SEBAGAI      |
            +-----+
            | [1.] Admin              |
            | [2.] Pembeli            |
            | [0.] Keluar              |
            +-----+
            """
        )
        try:
            pilihan = int(input("Masuk Sebagai: "))
        except ValueError:
            print("\nMasukkan angka pilihan (0/1/2) ")
            continue
        except KeyboardInterrupt:
            print("\nInput yang anda masukkan tidak valid")
            continue
        if pilihan == 1:
            os.system("cls")
            login_admin()
        elif pilihan == 2:
            os.system("cls")
            loginpelanggan()
        elif pilihan == 0:
            os.system("cls")
            print("Terimakasih Telah Menggunakan Program ini ^^ ! ")
            print(
                f"""
                =====
                SEMANGAT BEKERJA DAN JAGA KESEHATAN!
                =====
                """
            )
            break
        else:
            print("pilihan anda tidak valid")

main()
```

**Gambar 3.24** Alur Program: Menu Utama 1.1

```

# fungsi buat login admin
def login_admin():
    try:
        admin_regis = input("Apakah Sudah Memiliki Akun? y/t: ")
        if admin_regis == "y":
            print("~~~~~Silahkan Login~~~~~")
            print("Masukkan Username dan Password dengan Benar!")
            username = input("Masukkan username: ")
            pw = pwinput.pwinput("Masukkan password: ")
            os.system("cls")
            for i in data_admin:
                if i["username"] == username:
                    if i["password"] == pw:
                        print("Login Berhasil.")
                        print(f"Hallo Admin: {username}")
                        menuAdmin()
                        return
                    else:
                        print("password atau username anda masukkan salah")
                        return
            else:
                print("password atau username anda tidak terdaftar")
                return
        elif admin_regis == "t":
            while True:
                buatadmin = input("Apakah Ingin Membuat Akun Admin? y/t: ")
                if buatadmin == "y":
                    regis()
                elif buatadmin == "t":
                    break
                else:
                    print("Pilihan Anda Tidak Valid")
            else:
                print("silahkan input pilihan y/t")
        except FileNotFoundError:
            print("Terjadi Kesalahan Saat Membaca File.")
        except ValueError:
            print("Silahkan Coba Lagi Nanti.")
        except KeyboardInterrupt:
            print("\ninput yang anda masukkan tidak valid")

```

**Gambar 3.25** Alur Program: Menu Utama 1.2

```

# fungsi untuk menambah admin/registrasi
def regis():
    while True:
        try:
            print("Silahkan Registrasi Akun Anda^^")
            username = input("Masukkan nama admin baru: ")
            password = pwinput.pwinput("Masukkan password admin baru: ")
            if username == "":
                print("username tidak boleh kosong")
                continue
            if password == "":
                print("password tidak boleh kosong")
                continue
            for i in data_admin:
                if i["username"] == username:
                    print("username sudah tersedia, cari username yang lain!")
                    return
            continue
            data_admin.append({"username": username, "password": password})
            tabelAdmin.add_row([username, password])
            with open(adminjson, "w") as sn:
                json.dump(data_admin, sn, indent=4)
            print(f"Admin {username} berhasil ditambahkan.")
            print(tabelAdmin)
            break
        except ValueError:
            print("Masukkan input yang benar")
        except KeyboardInterrupt:
            print("\ninput yang anda masukkan tidak valid")
        except Exception as e:
            print("Gagal mendaftarkan admin: ", (e))

```

**Gambar 3.26** Alur Program: Menu Utama 1.3

```

# fungsi login pelanggan
def loginpelanggan():
    global saldo_pelanggan
    global inUsername
    print("Selamat Datang Di Apotek SI'A Kel 10")
    try:
        pilihMember = input("apakah sudah memiliki akun pelanggan? y/t: ")
        if pilihMember == "y":
            inUsername = input("masukkan username anda: ")
            inpass = pwinput.pwinput("masukkan password anda: ")
            for x in data_pelanggan:
                if x["username"] == inUsername:
                    for y in data_pelanggan:
                        if y["password"] == inpass:
                            saldo_pelanggan = x["saldo"]
                            os.system("cls")
                            print("Login Berhasil.")
                            print(f"Hallo Pelanggan: {inUsername}")
                            menu_pelanggan()
                            return True
            print("username atau password salah atau tidak terdaftar")
        elif pilihMember == "t":
            while True:
                pilihMember = input("Apakah Ingin Membuat Akun Pelanggan? y/t: ")
                if pilihMember == "y":
                    tambah_pelanggan()
                elif pilihMember == "t":
                    break
                else:
                    print("Pilihan Anda Tidak Valid")
    except ValueError:
        print("\nMasukkan input yang benar")
    except Exception as e:
        print(f"Terjadi kesalahan: {e}")
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")

```

**Gambar 3.27** Alur Program: Menu Utama 1.4

```

# fungsi untuk menambah pelanggan
def tambah_pelanggan():
    global data_pelanggan
    global tabel_produk
    while True:
        try:
            username = input("Masukkan nama pelanggan baru: ")
            password = pwinput.pwinput("Masukkan password pelanggan baru: ")
            saldo = int(input("berapa saldo anda: "))
            if username == "":
                print("username tidak boleh kosong")
                continue
            if password == "":
                print("password tidak boleh kosong")
                continue
            for i in data_pelanggan:
                if username == i["username"]:
                    print("username sudah tersedia, cari username yang lain!")
                    return
            continue
            data_pelanggan.append(
                {"username": username, "password": password, "saldo": saldo}
            )
            tabelPelanggan.add_row([username, password, saldo])
            with open(pelangganjson, "w") as sn:
                json.dump(data_pelanggan, sn, indent=4)
            print(f"Pelanggan Baru {username} berhasil ditambahkan.")
            print(tabelPelanggan)
            break
        except ValueError:
            print("\nMasukkan input yang benar")
        except Exception as e:
            print("\nGagal mendaftarkan admin: ", (e))
        except KeyboardInterrupt:
            print("\ninput yang anda masukkan tidak valid")

```

**Gambar 3.28** Alur Program: Menu Utama 1.5



## 2) Menu Admin

Dalam program ini, ada 5 pilihan yang bisa digunakan yaitu, “Lihat Produk”, “Tambah Produk”, “Update Produk”, “Hapus Produk”, dan “Back”. Jika admin ingin melihat tabel obat, maka admin perlu mengetik angka “1” dan tabel obat akan muncul. Jika admin ingin menambahkan obat baru, maka admin perlu mengetik angka “2” dan pengguna bisa menambahkan obat baru. Jika admin ingin mengedit obat, maka admin perlu mengetik angka “3” dan admin bisa mengedit obat. Jika admin ingin menghapus obat, maka admin perlu mengetik angka “4” dan admin bisa menghapus obat. Dan jika admin mengetik angka “0” admin bisa Kembali.

```
def menuAdmin():
    while True:
        print(
            f"""
            ~~~~~| SELAMAT DATANG |~~~~~
            ~~~~~| Apotek SI'A Kel 10 |~~~~~
            +-----+
            |      Menu Admin Apotek      |
            +-----+
            | [1.] Lihat Produk           |
            | [2.] Tambah Produk          |
            | [3.] Update Produk          |
            | [4.] Hapus Produk           |
            | [0.] Keluar                 |
            +-----+"""
        )
```

**Gambar 3.29** Alur Program: Menu Admin

## 3) Menu user

Pada menu pembeli, terdapat 6 menu yang bisa diakses oleh pembeli yaitu menu “Lihat dan Beli Obat”, “Isi Saldo”, “Tampilkan Saldo”, “Search Produk”, “Sorting Produk”, dan “Kembali”. Jika pembeli ingin melihat dan melakukan pemesanan obat maka pembeli tinggal mengetik angka 1. Jika pembeli ingin melakukan isi saldo maka pembeli hanya tinggal mengetik angka 2. Jika pembeli ingin melakukan cek saldo maka pembeli hanya tinggal mengetik angka 3. Jika pembeli ingin melakukan search produk maka pembeli hanya tinggal mengetik angka 4. Jika pembeli ingin melakukan sorting produk maka pembeli hanya tinggal mengetik angka 5. Dan jika mengetik angka 0 maka akan kembali ke menu utama.

```
def menu_pelanggan():
    global saldo_pelanggan
    while True:
        print(
            f"""
            ~~~~~| SELAMAT DATANG |~~~~~
            ~~~~~| Apotek SI'A Kel 10 |~~~~~
            +-----+
            |           Menu Pembeli Apotek           |
            +-----+
            | [1.] Lihat dan beli Produk |
            | [2.] Isi Saldo |
            | [3.] Tampilkan Saldo |
            | [4.] Search Produk |
            | [5.] Sorting Produk |
            | [0.] Keluar |
            +-----+"""
        )
    )
```

**Gambar 3.30** Alur Program: Menu Pembel

### 3.3 Tampilan Program

```
~~~~~| SELAMAT DATANG |~~~~~
~~~~~| Apotek SI'A Kel 10 |~~~~~
+-----+
|           MASUK SEBAGAI           |
+-----+
| [1.] Admin |
| [2.] Pembeli |
| [0.] Keluar |
+-----+
Masuk Sebagai:
```

**Gambar 3.32** Tampilan Program: Tampilan Awal

```
Apakah Sudah Memiliki Akun? y/t: y
~~~~~Silahkan Login~~~~~
Masukkan Username dan Password dengan Benar!
Masukkan username: rifqi
Masukkan password: ***
```

**Gambar 3.33** Tampilan Program: Login Admin

```
Selamat Datang Di Apotek SI'A Kel 10
apakah sudah memiliki akun pelanggan? y/t: y
masukkan username anda: arini
masukkan password anda: *****
```

**Gambar 3.34** Tampilan Program: Login Pembeli

```
Apakah Sudah Memiliki Akun? y/t: t
Apakah Ingin Membuat Akun Admin? y/t: y
Silahkan Registrasi Akun Anda^^
Masukkan nama admin baru: arien
Masukkan password admin baru: ***
Admin arien berhasil ditambahkan.
+-----+-----+
| nama admin | password admin |
+-----+-----+
| rifqi      | 123            |
| hadi       | 456            |
| wijaya     | 456            |
| arini      | 123            |
| arin       | arin           |
| nyoman     | 123            |
| ppp        | ppp            |
| aduh       | 123            |
| arien      | 111            |
+-----+-----+
Apakah Ingin Membuat Akun Admin? y/t: 
```

**Gambar 3.35** Tampilan Program: Registrasi Admin

```
Selamat Datang Di Apotek SI'A Kel 10
apakah sudah memiliki akun pelanggan? y/t: t
Apakah Ingin Membuat Akun Pelanggan? y/t: y
Masukkan nama pelanggan baru: arin
Masukkan password pelanggan baru: ***
berapa saldo anda: 2000
Pelanggan baru arin berhasil ditambahkan.
+-----+-----+-----+
| username | password | saldo |
+-----+-----+-----+
| rifqi    | 123      | 313000 |
| trirahayu | 123      | 40000  |
| fitriyanti | 789      | 30000  |
| arini     | arini    | 37000  |
| arin      | qqq      | 2000   |
+-----+-----+-----+
Apakah Ingin Membuat Akun Pelanggan? y/t: 
```

**Gambar 3.36** Tampilan Program: Registrasi Pembeli

```
Login Berhasil.
Hallo Admin: rifqi

~~~~~| SELAMAT DATANG |~~~~~
~~~~~| Apotek SI'A Kel 10 |~~~~~
+-----+
|      Menu Admin Apotek      |
+-----+
| [1.] Lihat Produk           |
| [2.] Tambah Produk          |
| [3.] Update Produk          |
| [4.] Hapus Produk           |
| [0.] Keluar                 |
+-----+
Masukkan pilihan Admin: █
```

**Gambar 3.7** Tampilan Program: Menu Admin

```
*****
                        APOTEK SI'23 KEL 10
*****

+---+---+---+---+---+---+
| ID | Nama Obat | Kategori Obat | Harga Obat | Stok Obat |
+---+---+---+---+---+---+
| 1  | Paracetamol | Pilek         | 5000       | 20        |
| 2  | Brufen      | Pilek         | 7000       | 15        |
| 3  | Dafalgan    | Batuk         | 8000       | 20        |
| 4  | Promag      | Mag           | 5000       | 25        |
| 5  | Maxtro      | Lambung       | 20000      | 12        |
+---+---+---+---+---+---+
```

**Gambar 3.38** Tampilan Program: Menu Lihat Produk

```
*****
APOTEK SI'23 KEL 10
*****

+---+-----+-----+-----+-----+
| ID | Nama Obat | Kategori Obat | Harga Obat | Stok Obat |
+---+-----+-----+-----+-----+
| 1 | Paracetamol | Pilek | 5000 | 20 |
| 2 | Brufen | Pilek | 7000 | 15 |
| 3 | Dafalgan | Batuk | 8000 | 20 |
| 4 | Promag | Mag | 5000 | 25 |
| 5 | Maxtro | Lambung | 20000 | 12 |
+---+-----+-----+-----+-----+
tambahkan id baru: 6
isi nama obatnya: Panadol
isi kategori obat: Sakit Kepala
isi harganya: 5000
tambahkan stok: 10

*****
APOTEK SI'23 KEL 10
*****

+---+-----+-----+-----+-----+
| ID | Nama Obat | Kategori Obat | Harga Obat | Stok Obat |
+---+-----+-----+-----+-----+
| 1 | Paracetamol | Pilek | 5000 | 20 |
| 2 | Brufen | Pilek | 7000 | 15 |
| 3 | Dafalgan | Batuk | 8000 | 20 |
| 4 | Promag | Mag | 5000 | 25 |
| 5 | Maxtro | Lambung | 20000 | 12 |
| 6 | Panadol | Sakit Kepala | 5000 | 10 |
+---+-----+-----+-----+-----+
```

**Gambar 3.39** Tampilan Program: Menu Tambah Produk

*****					
APOTEK SI'23 KEL 10					
*****					
+---+---+---+---+---+---+					
ID	Nama Obat	Kategori Obat	Harga Obat	Stok Obat	
+---+---+---+---+---+---+					
1	Paracetamol	Pilek	5000	20	
2	Brufen	Pilek	7000	15	
3	Dafalgan	Batuk	8000	20	
4	Promag	Mag	5000	25	
5	Maxtro	Lambung	20000	12	
6	Panadol	Sakit Kepala	5000	10	
+---+---+---+---+---+---+					
Silahkan Update Daftar Obat					
Masukkan ID Obat yang akan diperbarui: 6					
Masukkan Nama Obat baru: Bodrex					
Masukkan Kategori Obat Baru: Sakit Kepala Sebelah					
Masukkan Harga Obat baru: 2000					
Masukkan Stok Obat baru: 10					
Produk berhasil di Update					
*****					
APOTEK SI'23 KEL 10					
*****					
+---+---+---+---+---+---+					
ID	Nama Obat	Kategori Obat	Harga Obat	Stok Obat	
+---+---+---+---+---+---+					
1	Paracetamol	Pilek	5000	20	
2	Brufen	Pilek	7000	15	
3	Dafalgan	Batuk	8000	20	
4	Promag	Mag	5000	25	
5	Maxtro	Lambung	20000	12	
6	Bodrex	Sakit Kepala Sebelah	2000	10	
+---+---+---+---+---+---+					

**Gambar 3.40** Tampilan Program: Menu Update

```

*****
APOTEK SI'23 KEL 10
*****

```

ID	Nama Obat	Kategori Obat	Harga Obat	Stok Obat
1	Paracetamol	Pilek	5000	20
2	Brufen	Pilek	7000	15
3	Dafalgan	Batuk	8000	20
4	Promag	Mag	5000	25
5	Maxtro	Lambung	20000	12

```

*****
APOTEK SI'23 KEL 10
*****

```

ID	Nama Obat	Kategori Obat	Harga Obat	Stok Obat
1	Paracetamol	Pilek	5000	20
2	Brufen	Pilek	7000	15
3	Dafalgan	Batuk	8000	20
4	Promag	Mag	5000	25
5	Maxtro	Lambung	20000	12
6	Bodrex	Sakit Kepala Sebelah	2000	10

masukkan id obat yang akan dihapus: 6  
berhasil menghapus obat

**Gambar 3.41** Tampilan Program: Menu Hapus Produk

```

~~~~~| SELAMAT DATANG |~~~~~
~~~~~| Apotek SI'A Kel 10 |~~~~~

```

MASUK SEBAGAI
[1.] Admin
[2.] Pembeli
[0.] Keluar

Masuk Sebagai:

**Gambar 3.42** Tampilan Program: Kembali kemenu admin

```

Login Berhasil.
Hallo Pelanggan: arini

~~~~~| SELAMAT DATANG |~~~~~
~~~~~| Apotek SI'A Kel 10 |~~~~~
+-----+
|          Menu Pembeli Apotek          |
+-----+
|      [1.] Lihat dan beli Produk      |
|      [2.] Isi Saldo                   |
|      [3.] Tampilkan Saldo             |
|      [4.] Search Produk               |
|      [5.] Sorting Produk              |
|      [0.] Keluar                      |
+-----+
pilih menu: █

```

**Gambar 3.43** Tampilan Program: Menu Pembeli

```

~~~~~
                        SELAMAT BERBELANJA!
~~~~~
+---+---+---+---+---+---+
| ID | Nama Obat | Kategori Obat | Harga Obat | Stok Obat |
+---+---+---+---+---+---+
| 1  | Paracetamol | Pilek         | 5000       | 20        |
| 2  | Brufen      | Pilek         | 7000       | 15        |
| 3  | Dafalgan    | Batuk         | 8000       | 20        |
| 4  | Promag      | Mag           | 5000       | 25        |
| 5  | Maxtro      | Lambung       | 20000      | 12        |
+---+---+---+---+---+---+

Masukkan id obat: 1
Produk yang anda pilih adalah Paracetamol
Masukkan kuantitas barang: 3
Barang berhasil di beli
Cetak Struk Belanja Anda? y/t: y█

Tanggal Transaksi: 2023-11-03 13:58:08
Nama Pelanggan: arini
+---+---+---+---+---+---+
| ID Obat | Nama Obat | Kategori | Kuantitas | Harga Satuan | Total Harga |
+---+---+---+---+---+---+
| 1       | Paracetamol | Pilek    | 3         | 5000         | 15000       |
+---+---+---+---+---+---+

Saldo Anda Sebelum Transaksi: 47000
Saldo Anda Sesudah Transaksi: 32000
Terimakasih sudah berbelanja di Apotek SI'A Kel 10
Semoga Lekas Sembuh

```

**Gambar 3.45** Tampilan Program: Menu Lihat dan Beli Produk



```

~~~~~
SELAMAT DATANG
ISI SALDO ANDA
~~~~~
Masukkan berapa saldo yang ingin ditambahkan: 20000
Saldo Anda Berhasil Di Tambahkan!

```

**Gambar 3.45** Tampilan Program: Menu Isi Saldo

```
Login Berhasil.  
Hallo Admin: arini  
Saldo anda sekarang: 37000
```

**Gambar 3.46** Tampilan Program: Menu Tampilkan Saldo

```
Masukkan Nama Produk yang akan dicari: Paracetamol
pencarian anda: Paracetamol ada di data
+-----+-----+-----+-----+
|      Nama      | Kategori | Harga | Stok |
+-----+-----+-----+-----+
| Paracetamol    | Pilek    | 5000  | 20   |
+-----+-----+-----+-----+
```

**Gambar 3.47** Tampilan Program: Menu search produk

```
+-----+
|      Sorting barang      |
+-----+
|  1. Sorting Nama        |
|  2. Sorting Kategori   |
|  3. Sorting Harga      |
|  4. Sorting ID         |
|  0. Keluar             |
+-----+
```

Pilih Sorting berdasarkan (0/1/2/3/4): 1

**Gambar 3.48** Tampilan Program: Menu sorting produk

```
===|SORTING OBAT SESUAI URUTAN NAMA|===
```

ID	Nama Obat	Kategori Obat	Harga Obat	Stok Obat
2	Brufen	Pilek	7000	15
3	Dafalgan	Batuk	8000	20
5	Maxtro	Lambung	20000	12
1	Paracetamol	Pilek	5000	20
4	Promag	Mag	5000	25

**Gambar 3.49** Tampilan Program: Sorting berdasarkan nama

```
===|SORTING OBAT SESUAI URUTAN KATEGORI|===
```

ID	Nama Obat	Kategori Obat	Harga Obat	Stok Obat
3	Dafalgan	Batuk	8000	20
5	Maxtro	Lambung	20000	12
4	Promag	Mag	5000	25
1	Paracetamol	Pilek	5000	20
2	Brufen	Pilek	7000	15

**Gambar 3.50** Tampilan Program: Sorting berdasarkan kategori

```
===|SORTING OBAT SESUAI URUTAN HARGA TERRENDAH|===
```

ID	Nama Obat	Kategori Obat	Harga Obat	Stok Obat
1	Paracetamol	Pilek	5000	20
4	Promag	Mag	5000	25
2	Brufen	Pilek	7000	15
3	Dafalgan	Batuk	8000	20
5	Maxtro	Lambung	20000	12

**Gambar 3.51** Tampilan Program: Sorting berdasarkan harga terendah

```
~~~~~| SELAMAT DATANG |~~~~~
~~~~~| Apotek SI'A Kel 10 |~~~~~
+-----+
|          MASUK SEBAGAI          |
+-----+
|          [1.] Admin             |
|          [2.] Pembeli           |
|          [0.] Keluar            |
+-----+
Masuk Sebagai: █
```

**Gambar 3.52** Tampilan Program: Sorting Kembali

```
Terimakasih Telah Menggunakan Program ini ^^ !

=====
SEMANGAT BEKERJA DAN JAGA KESEHATAN!
=====
PS D:\projectAkhir re> █
```

**Gambar 3.53** Tampilan Program: Tampilan Akhir

### 3.4 Source Code

```
# Modul
import os # untuk mengakses sistem operasi yang mendasari aplikasi
Python kita
from prettytable import (
    PrettyTable,
) # Modul fitur untuk membuat tabel yang terlihat indah dan
teratur dalam teks.
import pwinput # Modul ini digunakan untuk menyembunyikan input
pengguna saat mengetik password.
import json # Modul ini digunakan untuk mengkonversi data antara
format JSON dan data bawaan Python.
import datetime # Modul ini menyediakan fungsi dan kelas untuk
mengolah tanggal dan waktu dalam Python.

# Modul Membersihkan Terminal
os.system("cls")
# merupakan perintah dalam bahasa pemrograman Python yang digunakan
untuk mengatur direktori root aplikasi.
ROOD_DIR = os.path.abspath(os.curdir)

# ini adalah prettytable untuk read produk obat
produkjson = f"{ROOD_DIR}/projectAkhir/crudpa.json"
try:
    with open(produkjson, "r") as jsonproduk:
        data_produk = json.loads(jsonproduk.read())
except FileNotFoundError:
    print("File tidak ditemukan.")
    data_produk = []
except Exception as e:
    print(f"Terjadi kesalahan: {e}")
except KeyboardInterrupt:
    print("\nMasukkan inputan dengan benar")

# ini adalah prettytable untuk read admin
tabelAdmin = PrettyTable()
tabelAdmin.field_names = ["nama admin", "password admin"]
adminjson = f"{ROOD_DIR}/projectAkhir/paloginadmin.json"
try:
    with open(adminjson, "r") as jsonadmin:
        data_admin = json.loads(jsonadmin.read())

    for i in data_admin:
        tabelAdmin.add_row([i["username"], i["password"]])
```

```

except FileNotFoundError:
    print(f"File not found: {adminjson}")
except KeyError:
    print("Invalid key in the JSON file.")
except json.JSONDecodeError:
    print("Invalid JSON file.")
except Exception as e:
    print(f"Unexpected error: {e}")
except KeyboardInterrupt:
    print("\nMasukkan inputan dengan benar")

# ini adalah prettytabel untuk read pelanggan apotik
tabelPelanggan = PrettyTable()
tabelPelanggan.field_names = ["username", "password", "saldo"]
pelangganjson = f"{ROOD_DIR}/projectAkhir/paloginpelanggan.json"
try:
    with open(pelangganjson, "r") as jsonpelanggan:
        data_pelanggan = json.loads(jsonpelanggan.read())

        for i in data_pelanggan:
            tabelPelanggan.add_row([i["username"], i["password"],
i["saldo"]])
except FileNotFoundError:
    print(f"File not found: {pelangganjson}")
except KeyError:
    print("\nInvalid key in the JSON file.")
except json.JSONDecodeError:
    print("\nInvalid JSON file.")
except Exception as e:
    print(f"\nUnexpected error: {e}")
except KeyboardInterrupt:
    print("\nMasukkan inputan dengan benar")

# variabel saldo pelanggan
saldo_pelanggan = 0

# fungsi untuk menyimpan file json bagian data pelanggan
def savedatapelanggan():
    try:
        with open(pelangganjson, "w") as sn:
            json.dump(data_pelanggan, sn, indent=4)
    except FileNotFoundError:
        print(f"File not found: {pelangganjson}")
    except Exception as e:
        print(f"Unexpected error: {e}")
    except KeyboardInterrupt:
        print("\nMasukkan inputan dengan benar")

```

```

# fungsi untuk menyimpan file json bagian produk
def savejsonProduk():
    try:
        with open(produkjson, "w") as sn:
            json.dump(data_produk, sn, indent=4)

    except FileNotFoundError:
        print(f"File not found: {produkjson}")
    except Exception as e:
        print(f"Unexpected error: {e}")
    except KeyboardInterrupt:
        print("\nMasukkan inputan dengan benar")

# fungsi untuk create produk
def create_produk():
    while True:
        try:
            tambahId = int(input("tambahkan id baru: "))
            tambahObat = input("isi nama obatnya: ")
            tambahKategori = input("isi kategori obat: ")
            tambahHarga = int(input("isi harganya: "))
            tambahStok = int(input("tambahkan stok: "))
            if tambahObat == "":
                print("nama obat tidak boleh kosong")
                continue
            if tambahKategori == "":
                print("kategori obat tidak boleh kosong")
                continue
            for x in data_produk:
                if tambahId == x["id"]:
                    print("id sudah tersedia, buat id yang lain!")
                    return
                continue
            data_produk.append(
                {
                    "id": tambahId,
                    "nama": tambahObat,
                    "kategori": tambahKategori,
                    "harga": tambahHarga,
                    "stok": tambahStok,
                }
            )
            tabel_produk.add_row(
                [tambahId, tambahObat, tambahKategori, tambahHarga,
                tambahStok]
            )

```

```

        print("obat berhasil ditambahkan")
        savejsonProduk()
        lagi = input("apakah anda ingin menambah obat lagi?
(y/n)")

        if lagi == "y":
            create_produk()
        elif lagi == "n":
            break
        else:
            print("input y/n")
    except ValueError:
        print("Masukkan input yang benar")
    except Exception as e:
        print("Error:", str(e))
    except KeyboardInterrupt:
        print("\nMasukkan inputan dengan benar")

# fungsi read produk
def read_produk():
    global tabel_produk
    tabel_produk = PrettyTable()
    tabel_produk.field_names = [
        "ID",
        "Nama Obat",
        "Kategori Obat",
        "Harga Obat",
        "Stok Obat",
    ]
    try:
        for x in data_produk:
            tabel_produk.add_row(
                [x["id"], x["nama"], x["kategori"], x["harga"],
x["stok"]]
            )
        print(tabel_produk)
    except FileNotFoundError:
        print("File tidak ditemukan.")
    except Exception as e:
        print(f"Terjadi kesalahan: {e}")
    except KeyboardInterrupt:
        print("\nMasukkan inputan dengan benar")

# fungsi update produk
def update_produk(id, nama=None, kategori=None, harga=None,
stok=None):
    read_produk()
    try:

```

```

        print("Silahkan Update Daftar Obat")
        id = int(input("Masukkan ID Obat yang akan diperbarui: "))
        nama = str(input("Masukkan Nama Obat baru: "))
        kategori = str(input("Masukkan Kategori Obat Baru: "))
        harga = int(input("Masukkan Harga Obat baru: "))
        stok = int(input("Masukkan Stok Obat baru: "))
        for item in data_produk:
            if item["id"] == id:
                if nama:
                    item["nama"] = nama
                if kategori:
                    item["kategori"] = kategori
                if harga:
                    item["harga"] = harga
                if stok:
                    item["stok"] = stok
                savejsonProduk()
                print("Produk berhasil di Update")
                return
        print("Produk tidak ditemukan")
        return
    except ValueError:
        print("\nMasukkan input yang benar")
    except Exception as e:
        print("\nError:", str(e))
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")

# fungsi delete produk
def delete_produk(data_produk):
    try:
        hapusId = int(input("masukkan id obat yang akan dihapus:
"))
        index = -1
        for i in range(0, len(data_produk)):
            id = data_produk[i]
            if hapusId == id["id"]:
                index = i
                break
        if index == -1:
            print("id obat tidak ditemukan")
        else:
            del data_produk[index]
            print("berhasil menghapus obat")
            savejsonProduk()
    except ValueError:
        print("Masukkan input yang benar")

```



```

except Exception as e:
    print("Error:", str(e))
except KeyboardInterrupt:
    print("\ninput yang anda masukkan tidak valid")

# Fungsi search produk
def search():
    try:
        cari_produk = str(input("Masukkan Nama Produk yang akan
dicari: "))
        produk_dict = {produk["nama"]: produk for produk in
data_produk}

        if cari_produk in produk_dict:
            print(f"pencarian anda: {cari_produk} ada di data ")

            if "nama" not in produk_dict[cari_produk]:
                print("Error: Data produk tidak lengkap, nama tidak
ada")

            else:
                global tabel_produk
                tabel_produk = PrettyTable()
                tabel_produk.field_names = ["Nama", "Kategori",
"Harga", "Stok"]
                tabel_produk.add_row(
                    [
                        produk_dict[cari_produk]["nama"],
                        produk_dict[cari_produk]["kategori"],
                        produk_dict[cari_produk]["harga"],
                        produk_dict[cari_produk]["stok"],
                    ]
                )
                print(tabel_produk)

        else:
            print(f"pencarian anda: {cari_produk} tidak ada di data
")

    except ValueError:
        print("Masukkan input yang benar")
    except Exception as e:
        print("Error:", str(e))
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")

# Fungsi Sorting produk nama
def sort_nama():
    global tabel_produk
    tabel_produk = PrettyTable()

```

```

tabel_produk.field_names = [
    "ID",
    "Nama Obat",
    "Kategori Obat",
    "Harga Obat",
    "Stok Obat",
]
try:
    for x in data_produk:
        tabel_produk.add_row(
            [x["id"], x["nama"], x["kategori"], x["harga"],
x["stok"]]
        )
        tabel_produk.sortby = "Nama Obat"
        print(tabel_produk)
except FileNotFoundError:
    print("File tidak ditemukan.")
except Exception as e:
    print(f"Terjadi kesalahan: {e}")
except KeyboardInterrupt:
    print("\ninput yang anda masukkan tidak valid")

# Fungsi Sorting produk kategori
def sort_kategori():
    global tabel_produk
    tabel_produk = PrettyTable()
    tabel_produk.field_names = [
        "ID",
        "Nama Obat",
        "Kategori Obat",
        "Harga Obat",
        "Stok Obat",
    ]
    try:
        for x in data_produk:
            tabel_produk.add_row(
                [x["id"], x["nama"], x["kategori"], x["harga"],
x["stok"]]
            )
            tabel_produk.sortby = "Kategori Obat"
            print(tabel_produk)
        except FileNotFoundError:
            print("File tidak ditemukan.")
        except Exception as e:
            print(f"Terjadi kesalahan: {e}")
        except KeyboardInterrupt:
            print("\ninput yang anda masukkan tidak valid")

```

```

# Fungsi Sorting produk harga
def sort_harga():
    global tabel_produk
    tabel_produk = PrettyTable()
    tabel_produk.field_names = [
        "ID",
        "Nama Obat",
        "Kategori Obat",
        "Harga Obat",
        "Stok Obat",
    ]
    try:
        for x in data_produk:
            tabel_produk.add_row(
                [x["id"], x["nama"], x["kategori"], x["harga"],
x["stok"]]
            )
        tabel_produk.sortby = "Harga Obat"
        print(tabel_produk)
    except FileNotFoundError:
        print("File tidak ditemukan.")
    except Exception as e:
        print(f"Terjadi kesalahan: {e}")
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")

# Fungsi Sorting produk ID
def sort_id():
    global tabel_produk
    tabel_produk = PrettyTable()
    tabel_produk.field_names = [
        "ID",
        "Nama Obat",
        "Kategori Obat",
        "Harga Obat",
        "Stok Obat",
    ]
    try:
        for x in data_produk:
            tabel_produk.add_row(
                [x["id"], x["nama"], x["kategori"], x["harga"],
x["stok"]]
            )
        tabel_produk.sortby = "ID"
        print(tabel_produk)
    except FileNotFoundError:

```

```

        print("File tidak ditemukan.")
    except Exception as e:
        print(f"Terjadi kesalahan: {e}")
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")

# fungsi untuk menu admin
def menuAdmin():
    while True:
        print(
            f"""
~~~~~| SELAMAT DATANG |~~~~~
~~~~~| Apotek SI'A Kel 10 |~~~~~
+-----+
|      Menu Admin Apotek      |
+-----+
| [1.] Lihat Produk           |
| [2.] Tambah Produk          |
| [3.] Update Produk          |
| [4.] Hapus Produk           |
| [0.] Keluar                 |
+-----+"""
        )
        try:
            pilihan = int(input("Masukkan pilihan Admin: "))
        except ValueError:
            print("\nmasukkan angka pilihan angka 0-4")
            continue
        except KeyboardInterrupt:
            print("\ninput yang anda masukkan tidak valid")
            continue
        if pilihan == 1:
            os.system("cls")
            print("*" * 60 + "\n")
            print("                                APOTEK SI'23 KEL
10                                ")
            print("*" * 60 + "\n")
            read_produk()
        elif pilihan == 2:
            os.system("cls")
            print("*" * 60 + "\n")
            print("                                APOTEK SI'23 KEL
10                                ")
            print("*" * 60 + "\n")
            read_produk()
            create_produk()
        elif pilihan == 3:

```

```

        os.system("cls")
        print("'" * 60 + "\n")
        print("
10         ")
        print("'" * 60 + "\n")
        update_produk(id, nama=None, harga=None, stok=None)
    elif pilihan == 4:
        os.system("cls")
        print("'" * 60 + "\n")
        print("
10         ")
        print("'" * 60 + "\n")
        read_produk()
        delete_produk(data_produk)
    elif pilihan == 0:
        os.system("cls")
        break
    else:
        print("masukkan angka dengan benar")

# fungsi melihat list admin
def liatAdmin():
    try:
        print(tabelAdmin)
    except FileNotFoundError:
        print("File tidak ditemukan.")
    except Exception as e:
        print(f"Terjadi kesalahan: {e}")
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")

# fungsi untuk menambah admin/registrasi
def regis():
    while True:
        try:
            print("Silahkan Registrasi Akun Anda^^")
            username = input("Masukkan nama admin baru: ")
            password = pwinput.pwinput("Masukkan password admin
baru: ")

            if username == "":
                print("username tidak boleh kosong")
                continue
            if password == "":
                print("password tidak boleh kosong")
                continue
            for i in data_admin:
                if username == i["username"]:

```

```

        print("username sudah tersedia, cari username
yang lain!")
        return
        continue
    data_admin.append({"username": username, "password":
password})
    tabelAdmin.add_row([username, password])
    with open(adminjson, "w") as sn:
        json.dump(data_admin, sn, indent=4)
    print(f"Admin {username} berhasil ditambahkan.")
    print(tabelAdmin)
    break
except ValueError:
    print("Masukkan input yang benar")
except KeyboardInterrupt:
    print("\ninput yang anda masukkan tidak valid")
except Exception as e:
    print("Gagal mendaftarkan admin: ", (e))

# fungsi buat login admin
def login_admin():
    try:
        admin_regis = input("Apakah Sudah Memiliki Akun? y/t: ")
        if admin_regis == "y":
            print("~~~~~Silahkan Login~~~~~")
            print("Masukkan Username dan Password dengan Benar!")
            username = input("Masukkan username: ")
            pw = pwinput.pwinput("Masukkan password: ")
            os.system("cls")
            for i in data_admin:
                if i["username"] == username:
                    if i["password"] == pw:
                        print("Login Berhasil.")
                        print(f"Hallo Admin: {username}")
                        menuAdmin()
                        return
                    else:
                        print("password atau username anda masukkan
salah")
                        return
                else:
                    print("password atau username anda tidak
terdaftar")
                    return
            elif admin_regis == "t":
                while True:

```

```

        buatadmin = input("Apakah Ingin Membuat Akun Admin?
y/t: ")

        if buatadmin == "y":
            regis()
        elif buatadmin == "t":
            break
        else:
            print("Pilihan Anda Tidak Valid")
    else:
        print("silahkan input pilihan y/t")
except FileNotFoundError:
    print("Terjadi Kesalahan Saat Membaca File.")
except ValueError:
    print("Silahkan Coba Lagi Nanti.")
except KeyboardInterrupt:
    print("\ninput yang anda masukkan tidak valid")

# fungsi login pelanggan
def loginpelanggan():
    global saldo_pelanggan
    global inUsername
    print("Selamat Datang Di Apotek SI'A Kel 10")
    try:
        pilihMember = input("apakah sudah memiliki akun pelanggan?
y/t: ")
        if pilihMember == "y":
            inUsername = input("masukkan username anda: ")
            inpass = pwininput.pwininput("masukkan password anda: ")
            for x in data_pelanggan:
                if x["username"] == inUsername:
                    for y in data_pelanggan:
                        if y["password"] == inpass:
                            saldo_pelanggan = x["saldo"]
                            os.system("cls")
                            print("Login Berhasil.")
                            print(f"Hallo Pelanggan: {inUsername}")
                            menu_pelanggan()
                            return True
                    print("username atau password salah atau tidak
terdaftar")
                elif pilihMember == "t":
                    while True:
                        pilihMember = input("Apakah Ingin Membuat Akun
Pelanggan? y/t: ")
                        if pilihMember == "y":
                            tambah_pelanggan()
                        elif pilihMember == "t":

```

```

        break
    else:
        print("Pilihan Anda Tidak Valid")
except ValueError:
    print("\nMasukkan input yang benar")
except Exception as e:
    print(f"Terjadi kesalahan: {e}")
except KeyboardInterrupt:
    print("\ninput yang anda masukkan tidak valid")

# fungsi untuk menambah pelanggan
def tambah_pelanggan():
    global data_pelanggan
    global tabel_produk
    while True:
        try:
            username = input("Masukkan nama pelanggan baru: ")
            password = pwininput("Masukkan password pelanggan
baru: ")

            saldo = int(input("berapa saldo anda: "))
            if username == "":
                print("username tidak boleh kosong")
                continue
            if password == "":
                print("password tidak boleh kosong")
                continue
            for i in data_pelanggan:
                if username == i["username"]:
                    print("username sudah tersedia, cari username
yang lain!")

                    return
                continue
            data_pelanggan.append(
                {"username": username, "password": password,
"saldo": saldo}
            )
            tabelPelanggan.add_row([username, password, saldo])
            with open('pelanggan.json', 'w') as sn:
                json.dump(data_pelanggan, sn, indent=4)
            print(f"Pelanggan baru {username} berhasil
ditambahkan.")
            print(tabelPelanggan)
            break
        except ValueError:
            print("\nMasukkan input yang benar")
        except Exception as e:
            print("\nGagal mendaftarkan admin: ", (e))

```



```

        except KeyboardInterrupt:
            print("\ninput yang anda masukkan tidak valid")

# fungsi untuk pembeli membeli obat dan dapat langsung mencetak
# struk
def belanja():
    global saldo_pelanggan
    produk_dict = {produk["id"]: produk for produk in data_produk}
    try:
        read_produk()
        pilih = int(input("Masukkan id obat: "))
    except FileNotFoundError:
        print("\nFile data_produk tidak ditemukan")
        return
    except ValueError:
        print("\nMasukkan inputan angka")
        return
    except KeyboardInterrupt:
        print("\ninput yang anda masukkan tidak valid")
        return
    if pilih in produk_dict:
        produk_terpilih = produk_dict[pilih]
        print(f"Produk yang anda pilih adalah
{produk_terpilih['nama']}")
        kuantitas = int(input("Masukkan kuantitas barang: "))
        if kuantitas <= produk_terpilih["stok"]:
            harga_total = produk_terpilih["harga"] * kuantitas
            produk_terpilih["stok"] -= kuantitas
            if saldo_pelanggan >= harga_total:
                sisa_saldo = saldo_pelanggan - harga_total
                saldo_pelanggan = sisa_saldo
                for pelanggan in data_pelanggan:
                    if pelanggan["username"] == inUsername:
                        pelanggan["saldo"] = saldo_pelanggan
                        savedatapelanggan()
                print("Barang berhasil di beli")
                cetakstruk = input("Cetak Struk Belanja Anda? y/t: ")
            else:
                print("Saldo tidak cukup")
        else:
            print("Stok barang tidak cukup")
    else:
        print("Produk tidak ditemukan")

    if cetakstruk == "y":
        os.system("cls")
        with open("struk.txt", "a") as f:
            f.write("\n")
            f.write(" " * 80 + "\n")
            f.write(
                "
                APOTEK SI'A KEL
            " + "\n"
            )

```

```

        f.write("*" * 80 + "\n")

        for pelanggan in data_pelanggan:
            if pelanggan["username"] == inUsername:
                tanggal_transaksi =
datetime.datetime.now().strftime(
                    "%Y-%m-%d %H:%M:%S"
                )
                print(f"Tanggal Transaksi:
{tanggal_transaksi}")
                f.write(f"Tanggal Transaksi:
{tanggal_transaksi}\n")

                print(f>Nama Pelanggan:
{inUsername}")
                f.write(f>Nama Pelanggan:
{inUsername}\n")

                tabelstruk = PrettyTable(
                    [
                        "ID Obat",
                        "Nama Obat",
                        "Kategori",
                        "Kuantitas",
                        "Harga Satuan",
                        "Total Harga",
                    ]
                )
                tabelstruk.add_row(
                    [
                        pilih,
                        produk_terpilih["nama"],
                        produk_terpilih["kategori"],

                        kuantitas,
                        produk_terpilih["harga"],
                        harga_total,
                    ]
                )
                print(tabelstruk)
                f.write(str(tabelstruk) + "\n")

                print(
                    f"Saldo Anda Sebelum Transaksi:
{saldo_pelanggan + harga_total}"
                )
                print(

```

```

f"Saldo Anda Sesudah Transaksi:
{saldo_pelanggan}"
    )
    print(
        "Terimakasih sudah berbelanja
di Apotek SI'A Kel 10"
    )
    print("Semoga Lekas Sembuh")
    elif cetakstruk == "t":
        print("Terimakasih sudah berbelanja di Apotek
SI'A Kel 10")
        print("Semoga Lekas Sembuh")
    else:
        print("Pilihan anda tidak valid")
    else:
        print("Saldo tidak cukup")
    else:
        print("Kuantitas yang anda pesan melebihi stok")
else:
    print("Barang tidak ada")

# fungsi menu pelanggan
def menu_pelanggan():
    global saldo_pelanggan
    while True:
        print(
            f"""
~~~~~| SELAMAT DATANG |~~~~~
~~~~~| Apotek SI'A Kel 10 |~~~~~
+-----+
|          Menu Pembeli Apotek          |
+-----+
| [1.] Lihat dan beli Produk             |
| [2.] Isi Saldo                         |
| [3.] Tampilkan Saldo                   |
| [4.] Search Produk                     |
| [5.] Sorting Produk                     |
| [0.] Keluar                             |
+-----+"""
        )
        try:
            pilih = int(input("pilih menu: "))
        except ValueError:
            print("\nmasukkan angka pilihan angka (0/1/2/3/4/5): ")
            continue
        except KeyboardInterrupt:
            print("\ninput yang anda masukkan tidak valid")

```

```

        continue
    if pilih == 1:
        os.system("cls")
        print("~" * 50)
        print("
BERBELANJA!                                SELAMAT
                                ")
        print("~" * 50)
        belanja()
    elif pilih == 2:
        try:
            os.system("cls")
            print("~" * 50)
            print("
DATANG                                SELAMAT
                                ")
            print("
ANDAS                                ISI SALDO
                                ")
            print("~" * 50)
            tambah_saldo = int(
                input("Masukkan berapa saldo yang ingin
ditambahkan: ")
            )
        except ValueError:
            print("\nmasukkan inputan yang benar")
            continue
        except KeyboardInterrupt:
            print("\nInput yang anda masukkan tidak valid")
            continue
        saldo_baru = saldo_pelanggan + tambah_saldo
        saldo_pelanggan = saldo_baru
        for pelanggan in data_pelanggan:
            if pelanggan["username"] == inUsername:
                pelanggan["saldo"] = saldo_pelanggan
                savedatapelanggan()
                print("Saldo Anda Berhasil Di Tambahkan!")
    elif pilih == 3:
        os.system("cls")
        print("Login Berhasil.")
        print(f"Hallo Admin: {inUsername}")
        print(f"Saldo anda sekarang: {saldo_pelanggan}")
    elif pilih == 4:
        os.system("cls")
        search()
    elif pilih == 5:
        os.system("cls")
        print(
            f"""
+-----+
|      Sorting barang      |
            """
        )

```

```

+-----+
| 1. Sorting Nama      |
| 2. Sorting Kategori |
| 3. Sorting Harga     |
| 4. Sorting ID        |
| 0. Keluar            |
+-----+
)
try:
    pilih = input("\nPilih Sorting berdasarkan
(0/1/2/3/4): ")
    if pilih == "1":
        os.system("cls")
        print("\n===|SORTING OBAT SESUAI URUTAN
NAMA|=== ")
        sort_nama()
    elif pilih == "2":
        os.system("cls")
        print("\n===|SORTING OBAT SESUAI URUTAN
KATEGORI|=== ")
        sort_kategori()
    elif pilih == "3":
        os.system("cls")
        print("\n===|SORTING OBAT SESUAI URUTAN HARGA
TERRENDAH|=== ")
        sort_harga()
    elif pilih == "4":
        os.system("cls")
        print("\n===|SORTING OBAT SESUAI URUTAN ID|===
")
        sort_id()
    elif pilih == "0":
        os.system("cls")
        break
    else:
        print("Pilihan anda tidak valid")
except ValueError:
    print("\nmasukkan inputan yang benar")
    continue
except KeyboardInterrupt:
    print("\nInput yang anda masukkan tidak valid")
    continue
elif pilih == 0:
    os.system("cls")
    break
else:
    print("Pilihan anda tidak valid")

```

```

# fungsi program utama
def main():
    while True:
        print(
            f"""
~~~~~| SELAMAT DATANG |~~~~~
~~~~~| Apotek SI'A Kel 10 |~~~~~
+-----+
|      MASUK SEBAGAI      |
+-----+
|      [1.] Admin        |
|      [2.] Pembeli      |
|      [0.] Keluar       |
+-----+"""
        )
        try:
            pilihan = int(input("Masuk Sebagai: "))
        except ValueError:
            print("\nMasukkan angka pilihan (0/1/2) ")
            continue
        except KeyboardInterrupt:
            print("\nInput yang anda masukkan tidak valid")
            continue
        if pilihan == 1:
            os.system("cls")
            login_admin()
        elif pilihan == 2:
            os.system("cls")
            loginpelanggan()
        elif pilihan == 0:
            os.system("cls")
            print("Terimakasih Telah Menggunakan Program ini ^^ !")
    print(
        f"""
=====
SEMANGAT BEKERJA DAN JAGA KESEHATAN!
===== """
    )
    break
else:
    print("pilihan anda tidak valid")

main()

```

--

**Tabel 3.1** Source Code



## **BAB IV**

### **Penutup**

#### **4.1 Kesimpulan**

Berdasarkan hasil diskusi kami, kesimpulan yang dapat kami ambil yaitu dengan diadakannya penilaian akhir ini diharapkan dapat membantu pembaca dalam memahami dan menambah wawasan tentang CRUD. Disini kami belajar tentang CRUD yang dimana, itu dibutuhkan dalam proses pengembangan bisnis untuk mempermudah proses transaksi pembelian barang. CRUD merupakan sebuah singkatan dari create, read, update dan delete. Create berfungsi untuk membuat record baru dalam sebuah database, read berfungsinya seperti search yang dimana dalam fungsi ini bisa melakukan pencarian dan juga mengambil data dari tabel dan membaca nilai dari data tersebut, update digunakan untuk memodifikasi record yang sudah dibuat sebelumnya dan tersimpan dalam database. Dan delete yang digunakan untuk menghapus sebuah record data yang sudah tidak dibutuhkan.

#### **4.2 Saran**

Dalam program yang kami buat ini, masih banyak kekurangan. Sehingga, kami mengharapkan masukan seperti kritik dan saran dari pembaca agar program yang kami buat menjadi lebih baik kedepannya.



## Daftar Pustaka

[https://www.w3school.com/phyton/phyton\\_datetime.asp](https://www.w3school.com/phyton/phyton_datetime.asp)

<https://zetcode.com/phyton/prettytable/>

<https://pypi.org/project/pwinput/>

<https://www.pythontutorial.net/python-basics/python-read-text-file/>

<https://jagongoding.com/phyton/dasar/perulangan-while/>

## Lampiran

Tabel Kontribusi

<b>Nama</b>	<b>Kontribusi</b>	<b>Bagian</b>
Nyoman Arini Trirahayu (2309116002)	Konsep, Coding, Pengecekan , Flowchart	1. Referensi konsep program 2. Coding 3. Membuat Flowchart 4. Mencari bug dan <i>error</i>
Fitri Yanti (2309116016)	Laporan	1. <i>Finishing</i> Laporan 2. Penyusun Laporan
Rifqi Hadi Wijaya (2309116042)	Konsep, Laporan, Pengembangan	1. Konsep program 2. Konsep dan penyusunan Laporan 3. Pengembangan program 4. Mengecek alur program

**Tabel 1.1** Tabel Kontribusi