

**LAPORAN PROYEK AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMROGRAMAN
DASAR**



MANAJEMEN DATA PASIEN RUMAH SAKIT

Oleh:

Kelompok 1

ANDI FAIRUL AMRI	2409106065
MUH. BRITAMA PUTRA JAYA	2409106047
DEVON FALEN PASAE	2409106055

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
SAMARINDA 2024**

KATA PENGANTAR

Puji syukur kita panjatkan kepada Tuhan Yang Maha Esa, atas limpahan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan penyusunan laporan ini dengan baik. Penulisan laporan ini bertujuan untuk menjelaskan penerapan program *Python* dalam merancang pengelolaan penitipan kendaraan, yang diharapkan dapat menjadi sumber pembelajaran dalam penerapannya.

Program ini dibuat untuk mengelola data pasien rumah sakit. Program ini berisi fitur utama tentang pengelolaan data seperti menambah, melihat, mengubah, dan menghapus data pasien yang ada di rumah sakit. Selain itu program ini juga dilengkapi fitur-fitur tambahan guna menunjang fitur utama.

Selama proses pengembangan program ini, kami menghadapi berbagai tantangan teknis dan non-teknis yang memerlukan penyesuaian agar program dapat berjalan dengan baik, seperti kesalahan dalam kode dan pengelolaan data, maupun kendala lainnya yang berkaitan dengan waktu dan pengujian sistem ini.

Kami ingin mengucapkan terima kasih kepada asisten kami Bang Ifnu Umar yang telah memberikan bantuan dan dukungan selama proses penyusunan laporan ini. Bimbingan, saran, dan kerja sama yang telah diberikan sangat membantu dalam pembuatan program yang direncanakan, sehingga laporan ini dapat diselesaikan dengan baik. Akhir kata, kami berharap laporan ini dapat diterima dengan baik dan dapat memberikan manfaat yang sebesar-besarnya bagi semua pihak yang terlibat.

Samarinda, 22 November 2024

Kelompok 1

TAKARIR

Database	Basis Data
Manajemen	Mengatur
Input	Memasukkan data dari pengguna ke dalam sistem.
Output	Menampilkan informasi
Login	Proses masuk ke dalam sistem
Register	Proses pendaftaran pengguna baru dalam sistem.
Admin	Pengguna dengan hak akses penuh untuk mengelola sistem.
Guest	Pengguna dengan hak akses terbatas, biasanya hanya bisa melihat data tertentu.
Flowchart	Diagram alir yang menunjukkan urutan langkah atau alur proses.
Validasi	Proses pengecekan untuk memastikan input pengguna sesuai dengan kriteria yang ditetapkan.
Autentikasi	Proses verifikasi identitas pengguna melalui login.
Percabangan	Proses pemilihan kondisi dalam alur program menggunakan perintah if, elif, atau else.
Perulangan	Proses pengulangan dalam program yang menggunakan struktur while.
List	Tipe data yang digunakan untuk menyimpan beberapa item dalam satu variabel.
Dictionary	Struktur data yang menyimpan pasangan kunci dan nilai.

Modul Eksternal	Kumpulan fungsi eksternal yang digunakan dalam program, seperti os untuk membersihkan layar dan tabulate untuk menampilkan tabel.
Menu	Antarmuka pilihan yang digunakan untuk navigasi dalam program.
Password	Kunci keamanan yang digunakan pengguna untuk mengakses akun.
Username	Nama pengguna

DAFTAR ISI

KATA PENGANTAR	ii
TAKARIR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Kebutuhan Fungsional.....	2
1.3 Rumusan Masalah	3
1.4 Batasan Masalah.....	3
1.5 Tujuan.....	3
BAB II PERANCANGAN.....	5
2.1 Analisis Program	5
2.2 Flowchart.....	6
2.3 Konsep/Materi Praktikum yang dipakai.....	11
BAB III HASIL DAN PEMBAHASAN.....	14
3.1 Tampilan Program.....	14
1. Tampilan Menu Autentikasi.....	14
2. Menu login	14
3. Menu Admin	15
4. Menu Dokter	16
5. Menu Guest	16
3.2 Source Code	17
BAB IV PENUTUP	26

4.1 Kesimpulan.....	26
4.2 Saran	26
DAFTAR PUSTAKA	27
LAMPIRAN.....	28

DAFTAR GAMBAR

Gambar 2. 1 Flowchart utama	6
Gambar 2. 2 Flowchart menu login.....	7
Gambar 2. 3 Flowchart menu register	8
Gambar 2. 4 Flowchart menu admin	9
Gambar 2. 5 Flowchart menu dokter	10
Gambar 2. 6 Flowchart menu guest.....	11
Gambar 3. 1 Menu autentikasi	14
Gambar 3. 2 Menu login	14
Gambar 3. 3 Menu login 2	14
Gambar 3. 4 Menu login 3	15
Gambar 3. 5 Menu admin.....	15
Gambar 3. 6 Menu admin 2.....	15
Gambar 3. 7 Menu dokter	16
Gambar 3. 8 Menu dokter 2	16
Gambar 3. 9 Menu guest	17
Gambar 3. 10 Menu guest 2	17
Gambar 3. 11 Source code 1	17
Gambar 3. 12 Source code 2	18
Gambar 3. 13 Source code 3	19
Gambar 3. 14 Source code 4	20
<i>Gambar 3. 15 Source code 5</i>	<i>21</i>
<i>Gambar 3. 16 Source code 6.....</i>	<i>22</i>
<i>Gambar 3. 17 Source code 7.....</i>	<i>23</i>
<i>Gambar 3. 18 Source code 8.....</i>	<i>24</i>
<i>Gambar 3. 19 Source code 9.....</i>	<i>24</i>
Lampiran 1 Lembar Konsul (Cover)	28
Lampiran 2 Lembar Konsul (Isi).....	28

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah memberikan dampak yang signifikan dalam berbagai bidang, termasuk kesehatan. Seiring dengan meningkatnya kebutuhan layanan kesehatan, rumah sakit sebagai institusi kesehatan dituntut untuk mengelola data pasien dengan lebih efisien dan akurat. Pengelolaan data yang baik tidak hanya mempermudah tenaga medis dalam memberikan layanan kesehatan, tetapi juga memastikan keamanan dan kerahasiaan data pasien.

Data pasien merupakan komponen vital dalam proses pelayanan kesehatan, mencakup informasi medis, riwayat penyakit, hasil pemeriksaan, dan data pribadi lainnya. Pengelolaan data yang tidak efisien dapat menyebabkan berbagai masalah, seperti kesalahan diagnosa, perawatan yang tidak tepat, dan keterlambatan dalam proses pelayanan. Oleh karena itu, dibutuhkan sistem manajemen data yang efektif untuk memastikan bahwa semua informasi dapat diakses dengan cepat, akurat, dan aman.

Dalam konteks ini, sistem manajemen data pasien berbasis Python diusulkan sebagai solusi untuk mengatasi tantangan tersebut. Python dipilih karena merupakan bahasa pemrograman yang populer, memiliki sintaks yang mudah dipahami, dan didukung oleh berbagai library yang kuat. Sistem ini dirancang untuk memenuhi kebutuhan rumah sakit dalam hal pendaftaran pasien baru, pengelolaan data pasien, pencarian data, dan pembuatan laporan kesehatan. Selain itu, sistem ini juga dilengkapi dengan fitur keamanan untuk memastikan kerahasiaan data pasien.

Secara lebih luas, penerapan teknologi dalam pengelolaan data pasien juga sejalan dengan upaya pemerintah dan organisasi kesehatan global untuk meningkatkan kualitas layanan kesehatan melalui digitalisasi. Digitalisasi sistem kesehatan tidak hanya meningkatkan efisiensi operasional, tetapi juga

memungkinkan analisis data kesehatan yang lebih mendalam untuk pengambilan keputusan yang lebih baik. Dalam jangka panjang, pengembangan sistem manajemen data pasien yang efektif dapat berkontribusi pada peningkatan kualitas layanan kesehatan dan kepuasan pasien.

Dengan latar belakang ini, program manajemen data pasien berbasis Python diharapkan dapat memberikan solusi praktis dan efisien bagi rumah sakit dalam mengelola data pasien, serta mendukung upaya peningkatan kualitas layanan kesehatan secara keseluruhan. Inisiatif ini juga membuka peluang untuk pengembangan lebih lanjut dan inovasi di bidang teknologi kesehatan, yang pada akhirnya akan memberikan manfaat besar bagi masyarakat.

1.2 Kebutuhan Fungsional

Program manajemen data pasien yang dikembangkan memiliki beberapa kebutuhan fungsional utama, antara lain:

- **Manajemen Akun Pengguna:**
 - Admin dapat melihat, menambah, mengedit, dan menghapus akun dokter dan guest.
 - Dokter dapat melihat data pasien, menambah, mengedit, dan menghapus data pasien yang menjadi tanggung jawabnya.
 - Guest dapat melihat data pasien yang tersedia tanpa dapat melakukan perubahan.
 - Setiap pengguna memiliki hak akses yang dibatasi berdasarkan peran mereka (admin, dokter, guest).
- **Login dan Register:**
 - Pengguna (admin, dokter, guest) dapat melakukan login dengan menggunakan username dan password yang valid.
 - Guest dapat mendaftar untuk membuat akun baru, sementara admin dan dokter tidak dapat mendaftar sebagai guest.
- **Manajemen Data Pasien:**
 - Dokter dapat mengelola data pasien yang menjadi tanggung jawabnya, termasuk menambah, mengedit, atau menghapus data pasien.
 - Admin dapat mengelola data pasien dan dokter secara keseluruhan.

- Data pasien mencakup informasi seperti ID pasien, nama, jenis kelamin, golongan darah, berat badan, tinggi badan, diagnosa, dokter yang menangani, dan ruangan.
- Validasi Input:
 - Sistem memvalidasi input dari pengguna untuk memastikan bahwa data yang dimasukkan valid (misalnya, password minimal 8 karakter).
- Pengelolaan Password:
 - Pengguna dapat mengganti password mereka, dengan validasi bahwa password baru harus terdiri dari minimal 8 karakter.
- Tampilan Data dalam Bentuk Tabel:
 - Data ditampilkan dalam format tabel yang mudah dibaca dan dikelola menggunakan tabulasi yang jelas.

1.3 Rumusan Masalah

Adapun rumusan masalah pada program ini yakni

1. Bagaimana cara mengelola data akun pengguna (admin, dokter, dan guest) dengan aman dan sesuai hak aksesnya?
2. Bagaimana cara memastikan dokter hanya bisa mengelola data pasien yang menjadi tanggung jawabnya?
3. Bagaimana cara membuat sistem yang mudah digunakan untuk mengelola data pasien dan akun pengguna?

1.4 Batasan Masalah

Program ini memiliki beberapa batasan, yaitu:

1. Program ini hanya akan dikembangkan menggunakan bahasa pemrograman Python
2. Data yang dikelola terbatas pada data pasien yang mencakup informasi dasar
3. Sistem tidak terintegrasi dengan system informasi rumah sakit lainnya

1.5 Tujuan

Tujuan dari pengembangan program adalah

1. Untuk membuat program pengelolaan data pasien rumah sakit dengan menggunakan Bahasa python
2. Memudahkan admin dalam mengelola akun dokter dan guest dengan fitur tambah, edit, dan hapus akun.
3. Membantu dokter dan admin dalam mengelola data pasien dengan mudah, baik untuk menambah, mengedit, maupun menghapus data pasien.
4. Membantu dokter dan admin dalam mengelola data pasien dengan mudah, baik untuk menambah, mengedit, maupun menghapus data pasien.

BAB II

PERANCANGAN

2.1 Analisis Program

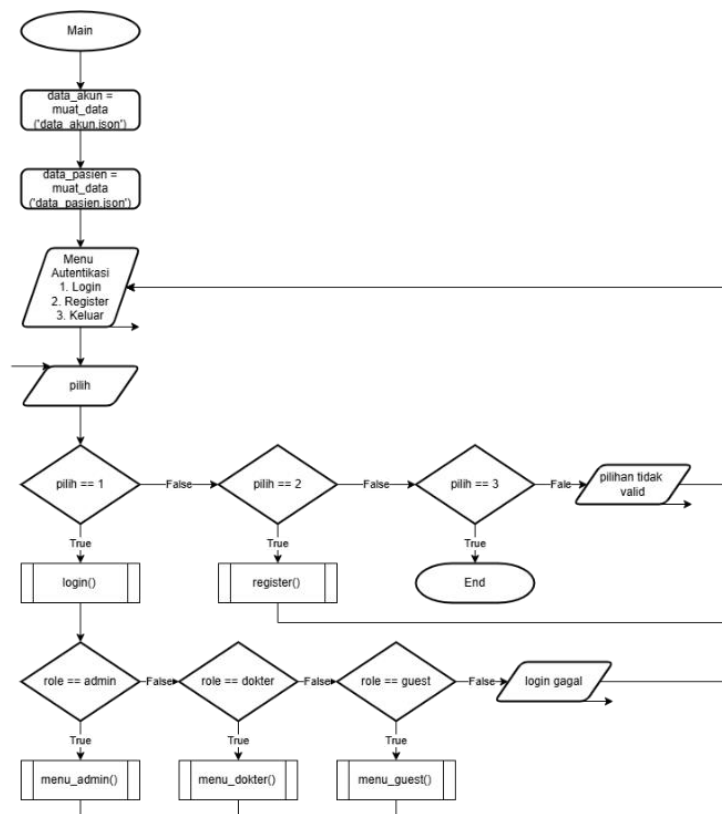
Program ini merupakan media untuk mengelola data pasien di rumah sakit. Ada tiga jenis peran dalam program ini yaitu admin, dokter, dan guest. Masing-masing peran memiliki hak akses yang berbeda. Berikut penjelasan mengenai alur dari program:

- Login dan registrasi
 - Pengguna diminta untuk memasukkan username dan password yang telah terdaftar
 - Pilihan registrasi dikhususkan untuk guest. Lalu setelah memiliki akun, dapat login sesuai dengan akun yang telah terdaftar.
 - Sistem akan memverifikasi kecocokan data yang dimasukkan dengan data yang ada di file data_akun.json.
 - Jika berhasil, sistem akan mengarahkan pengguna ke menu sesuai dengan perannya: admin, dokter, atau guest.
- Pembagian menu
 - Admin dapat melihat, menambah, mengedit, atau menghapus data akun dokter dan guest. Admin juga dapat melihat daftar data pasien.
 - Dokter dapat melihat daftar pasien yang ditangani, menambah, mengedit, atau menghapus data pasien. Dokter juga bisa mengganti password.
 - Guest hanya dapat melihat data pasien dan mengganti password mereka sendiri.
- Logout:
 - Pengguna dapat keluar dari sistem dengan memilih opsi logout. Sistem akan kembali ke layar login. Dan akan ada pilihan keluar di menu login yang akan keluar program saat dipilih.
- Validasi dan keamanan
 - Password harus memenuhi kriteria minimal panjang karakter (misalnya 8 karakter) untuk meningkatkan keamanan.

- Input yang dimasukkan pengguna harus divalidasi agar tidak ada data yang kosong atau tidak sesuai format.
- Format penyimpanan data
 - Data akun pengguna dan data pasien disimpan dalam file dengan format JSON. Hal ini dilakukan untuk memudahkan penyimpanan dan pembacaan data secara sederhana.
 - Data akun pengguna disimpan dengan nama data_akun.json dan data pasien disimpan dalam data_pasien.json

2.2 Flowchart

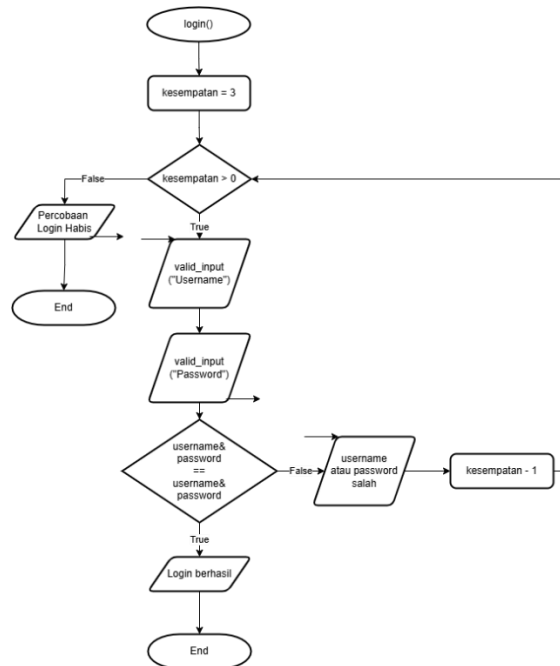
Berikut flowchart yang telah dibuat



Gambar 2. 1 Flowchart utama

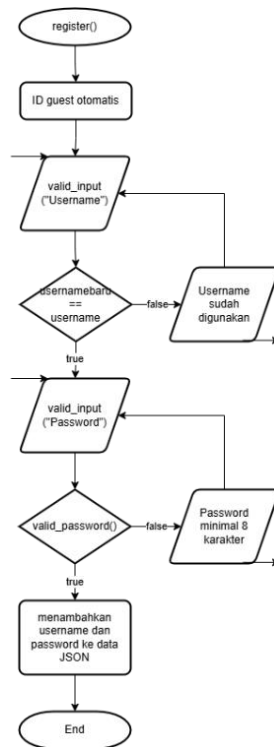
Di awal program akan membaca file eksternal dengan format json. Setelah itu akan muncul output menu autentikasi yang akan mengarahkan pengguna untuk login agar dapat memasuki menu utama. Pilihan menu 1 akan melanjutkan program ke proses login yang akan meminta memasukkan username dan password agar bisa masuk ke dalam menu utama. Menu utama akan disesuaikan

dengan peran pengguna masing-masing. Menu 2 dikhususkan untuk guest atau tamu yang harus mendaftar terlebih dahulu dan memiliki akun agar yang akan digunakan pada proses login. Menu 3 untuk memberhentikan program.



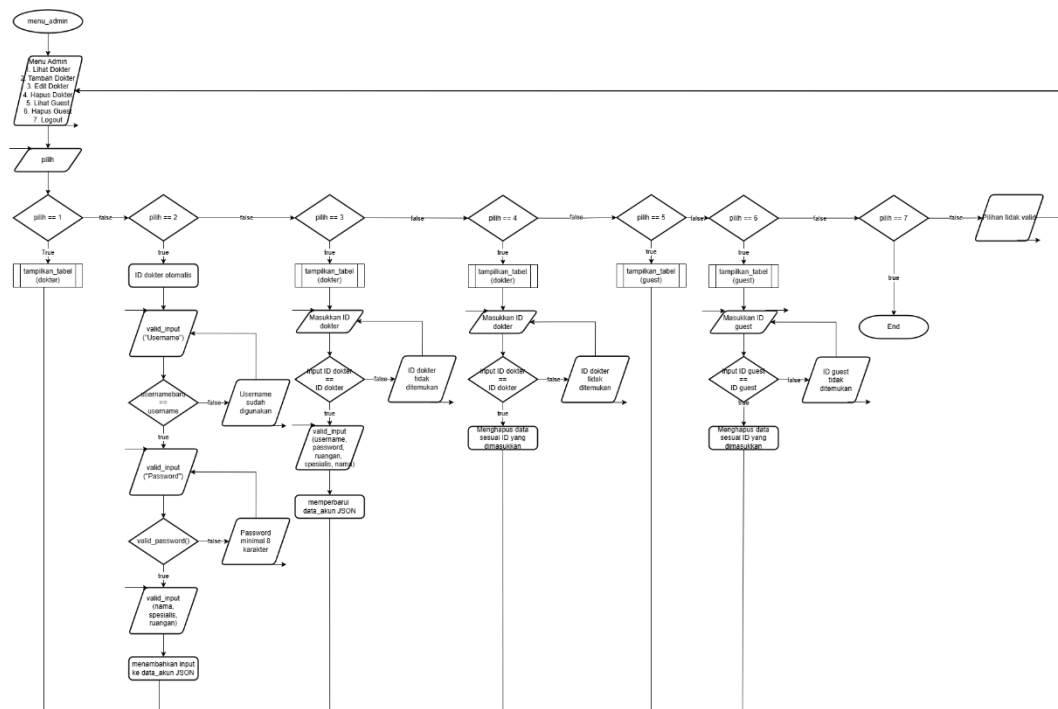
Gambar 2. 2 Flowchart menu login

Pada menu login, diperlukan masukan username dan password untuk melanjutkan menu sesuai dengan peran masing-masing. Kesempatan memasukkan username dan password dengan benar adalah sebanyak 3 kali. Jika kesempatan habis, maka program akan berhenti.



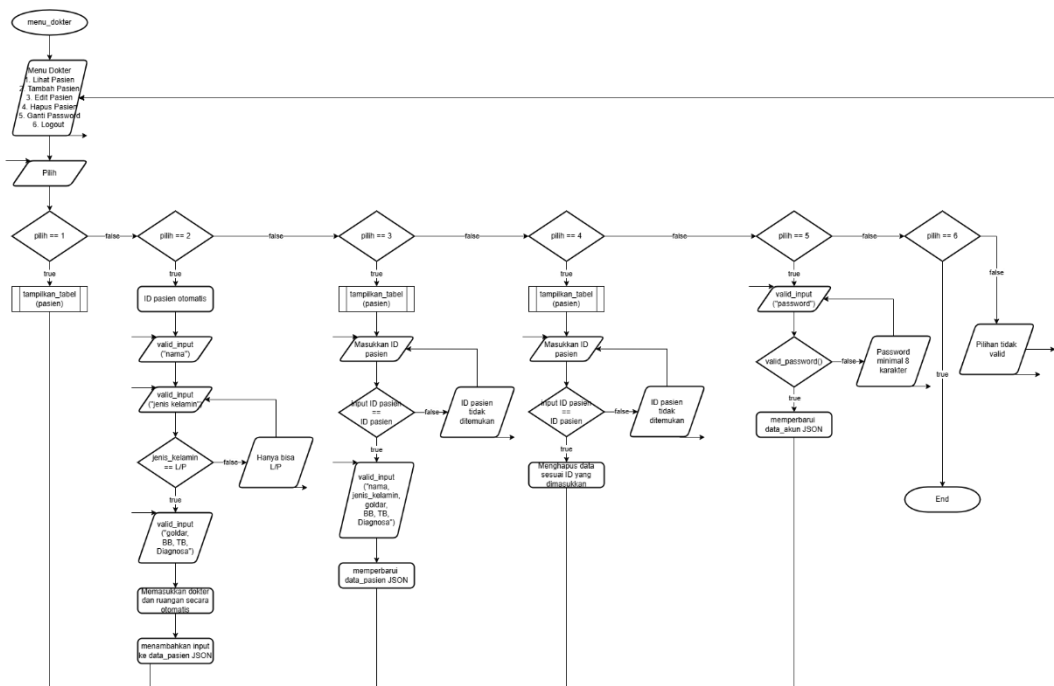
Gambar 2. 3 Flowchart menu register

Pada menu register, guest akan diminta untuk memasukkan username dan password yang nantinya akan digunakan pada saat login dengan beberapa ketentuan seperti username tidak boleh sama dengan data sebelumnya, serta password minimal 8 karakter.



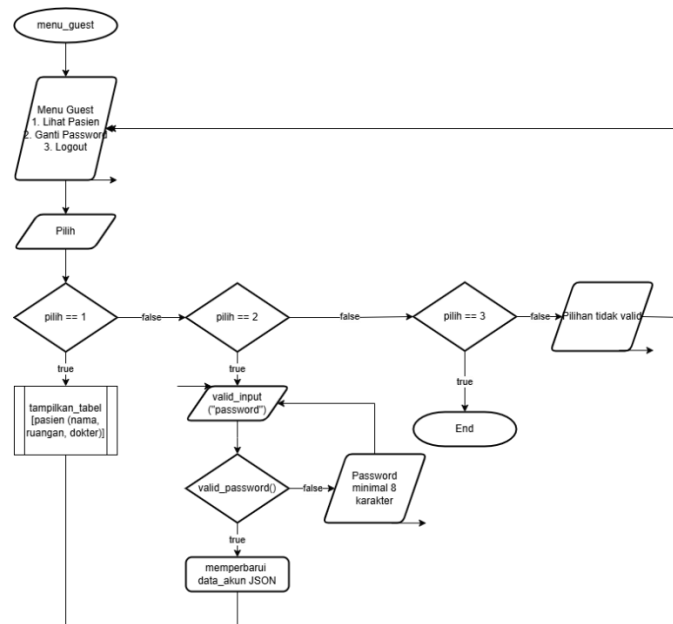
Gambar 2. 4 Flowchart menu admin

Pada menu admin, terdapat banyak pilihan menu yang tersedia diantaranya pengelolaan data akun dokter dan guest. Jika memilih menu 1, akan menampilkan data akun dokter. Jika memilih menu 2, program akan berlanjut dengan mendaftarkan akun dokter. Jika memilih menu 3, maka akan menampilkan data lalu akan meminta input untuk mengubah data dokter. Jika memilih menu 4, maka akan menampilkan data akun dokter lalu akan meminta input untuk menghapus dokter. Jika memilih menu 5, maka akan menampilkan akun guest. Jika memilih menu 6, maka akan menampilkan data akun guest lalu akan meminta input untuk menghapus data akun guest. Jika memilih menu 7, maka akan keluar dari menu admin dan kembali ke menu autentikasi. Jika memilih selain itu, maka akan muncul pesan bahwa pilihan tidak tersedia dan menu akan terulang. Menu akan berulang secara terus menerus hingga pengguna memilih menu keluar.



Gambar 2. 5 Flowchart menu dokter

Pada menu dokter, akan ada banyak menu yang tersedia terkait pengelolaan data pasien di rumah sakit. Jika memilih menu 1, maka akan menampilkan data pasien. Jika memilih menu 2, maka akan program meminta input data pasien. Jika memilih menu 3, maka akan menampilkan data pasien yang selanjutnya meminta input untuk mengubah data pasien. Jika memilih menu 4, maka akan menampilkan data pasien lalu meminta input untuk menghapus data pasien. Jika memilih menu 5, maka akan meminta password baru untuk memperbarui password. Jika memilih menu 6, maka akan kembali ke menu autentikasi. Jika memilih selain itu, maka akan muncul pesan bahwa pilihan tidak tersedia dan menu akan terulang. Menu akan berulang secara terus menerus hingga pengguna memilih menu keluar.



Gambar 2. 6 Flowchart menu guest

Pada menu guest, ada beberapa menu yang tersedia. Menu pada guest hanya untuk menampilkan data pasien serta mengganti password. Jika memilih menu 1, maka akan menampilkan data pasien yang ada. Jika memilih menu 2, maka akan meminta input password baru untuk memperbarui password guest pada data akun. Jika memilih menu 3, maka akan kembali ke menu autentikasi. Jika memilih selain itu, maka akan muncul pesan bahwa pilihan tidak tersedia dan menu akan terulang. Menu akan berulang secara terus menerus hingga pengguna memilih menu keluar.

2.3 Konsep/Materi Praktikum yang dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan persyaratan yang terdapat pada modul diantaranya:

1. Fungsi dasar.

Fungsi dasar yang digunakan antara lain:

- Input, digunakan untuk mengambil data dari pengguna. Contohnya saat login, menambah data dokter, atau menambah data pasien, program

meminta input dari pengguna seperti username, password, atau data spesifik pasien.

- Output, dengan menggunakan fungsi `print()` untuk menampilkan informasi kepada pengguna, seperti menampilkan menu, pesan kesalahan, atau informasi berhasil.

2. Percabangan

Menggunakan `if`, `elif`, dan `else` untuk mengecek kondisi tertentu dan menentukan alur program berdasarkan kondisi tersebut. Seperti dalam menentukan role pengguna saat login untuk pembagian menu.

3. Perulangan

Menggunakan `while` loop untuk mengurangi proses tertentu hingga kondisi berhenti terpenuhi. Digunakan pada kesempatan login sebanyak 3 kali, mengulangi input hingga data valid diberikan, serta menampilkan menu secara terus menerus hingga pengguna memilih opsi keluar.

4. List

Program memanfaatkan `list` untuk menyimpan dan mengelola data akun dan data pasien yang diambil dari file JSON. Contohnya, `data_akun` dan `data_pasien` adalah `list` yang menyimpan data dalam bentuk `dictionary`.

5. Dictionary

Digunakan untuk menyimpan data dengan struktur yang lebih kompleks. Contoh data pasien atau akun yang masing-masing memiliki atribut.

6. Fungsi dan prosedur

Fungsi menggunakan potongan kode yang mengembalikan suatu nilai atau hasil. Fungsi digunakan agar dapat dipanggil berulang. Digunakan saat memuat data dan memvalidasi masukan.

Prosedur merupakan Potongan kode yang melakukan suatu aksi tetapi tidak mengembalikan nilai. Dalam program ini, fungsi-fungsi seperti

menu_admin() dan menu_dokter() berperan sebagai prosedur karena menjalankan logika menu tanpa mengembalikan nilai.

7. Error handling

Try-Except digunakan dalam program untuk menangani kesalahan. Seperti pada saat membaca file json. Jika file tidak ditemukan, maka program akan mengembalikan list kosong daripada menyebabkan error. Validasi input juga digunakan untuk melakukan validasi input yang salah atau kosong dengan memberikan pesan kesalahan dan meminta pengguna untuk mengisi ulang input yang benar.

8. File eksternal

Dalam program ini file eksternal yang digunakan yaitu JSON (JavaScript Object Notation). Hal ini digunakan agar data yang tersimpan dapat dibuka sewaktu-waktu walau program telah dihentikan.

9. Modul eksternal

Beberapa modul juga digunakan seperti modul os untuk membersihkan layer, modul tabulate untuk menampilkan data dalam bentuk tabel.

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Menu Autentikasi

```
=== Selamat Datang di Sistem Rumah Sakit ===  
1. Login  
2. Register (guest)  
3. Keluar  
Pilih opsi:
```

Gambar 3. 1 Menu autentikasi

Menu ini terdiri dari login, register yang dikhususkan untuk tamu, serta menu keluar.

2. Menu login

```
== Menu Login ==  
Username: admin  
Password: admin123S
```

Gambar 3. 2 Menu login

Pada menu ini, pengguna akan diminta untuk memasukkan username dan password. Jika benar, maka akan lanjut ke menu selanjutnya. Jika salah, maka akan diberikan kesempatan 2 kali lagi untuk memasukkan inputan yang benar.

```
== Menu Login ==  
Username: admin  
Password: admin1234  
Username atau password salah. Sisa percobaan : 2  
Tekan enter untuk melanjutkan...
```

Gambar 3. 3 Menu login 2

Jika kesempatan habis, maka program akan terhenti.

```

● == Menu Login ==
Username: afa
Password: afa
Username atau password salah. Sisa percobaan : 0
Tekan enter untuk melanjutkan...
Terlalu banyak percobaan login yang gagal. Program akan berhenti.

```

Gambar 3. 4 Menu login 3

3. Menu Admin

```

D === Menu Admin ===
1. Lihat Data Akun Dokter
2. Tambah Akun Dokter
3. Edit Akun Dokter
4. Hapus Akun Dokter
5. Lihat Data Guest
6. Hapus Akun Guest
7. Logout
Pilih opsi: █

```

Gambar 3. 5 Menu admin

Menu ini dikhususkan untuk pengguna dengan peran sebagai admin. Pada menu ini tersedia pengelolaan data akun dokter dan guest. Admin memiliki akses untuk melihat data akun dokter dan guest. Selain itu, admin juga dapat mengubah dan menghapus akun dokter serta menghapus akun guest.

```

D === Menu Admin ===
1. Lihat Data Akun Dokter
2. Tambah Akun Dokter
3. Edit Akun Dokter
4. Hapus Akun Dokter
5. Lihat Data Guest
6. Hapus Akun Guest
7. Logout
Pilih opsi: 1

```

ID Dokter	Username	Nama	Spesialis	Ruangan
DKT001	devon	Dr. Devon	Umum	A101
DKT002	andi	Dr. Andi	Anak	B101
DKT006	fahrul	Dr. Fahrul	Gigi	C101
DKT007	ahmad	dr. Ahmad	Jantung	D303

```

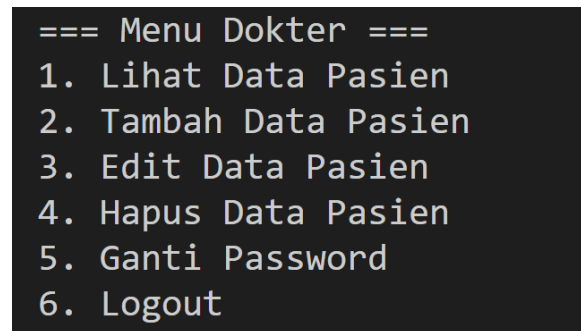
Tekan enter untuk kembali...█

```

Gambar 3. 6 Menu admin 2

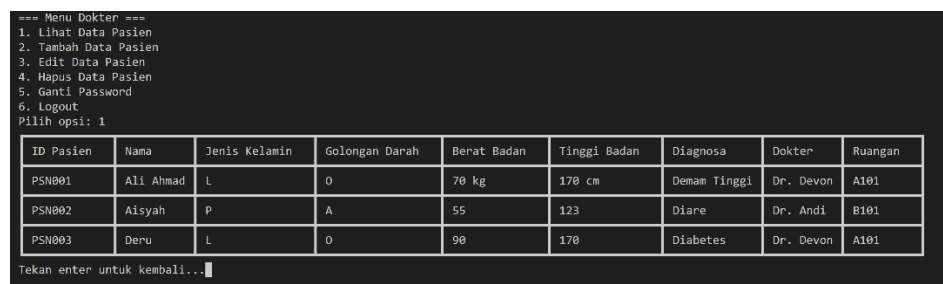
Gambar tersebut merupakan tampilan dari salah satu menu yaitu lihat data akun dokter. Data tersebut tersaji di dalam tabel agar membuat data lebih mudah untuk dibaca.

4. Menu Dokter



Gambar 3. 7 Menu dokter

Menu ini dikhususkan untuk pengguna dengan peran sebagai dokter. Pada menu ini tersedia pengelolaan data pasien, mulai dari melihat, menambah, mengedit, serta menghapus data pasien di rumah sakit. Pengguna juga dapat mengganti password dengan memilih ganti password pada menu. Serta memiliki menu logout yang akan mengembalikan program ke menu autentikasi.



Gambar 3. 8 Menu dokter 2

Gambar tersebut merupakan tampilan dari salah satu menu yaitu lihat data pasien. Data tersebut tersaji di dalam tabel agar membuat data lebih mudah untuk dibaca.

5. Menu Guest

```

=== Menu Guest ===
1. Lihat Data Pasien
2. Ganti Password
3. Logout
Pilih opsi: █

```

Gambar 3. 9 Menu guest

Menu ini dikhususkan untuk pengguna dengan peran sebagai guest. Pada menu ini tersedia menu untuk melihat data pasien, namun terbatas pada beberapa data saja. Pilihan menu yang selanjutnya memungkinkan pengguna dapat mengganti password. Serta memiliki menu logout yang akan mengembalikan program ke menu autentikasi.

```

=== Menu Guest ===
1. Lihat Data Pasien
2. Ganti Password
3. Logout
Pilih opsi: 1

```

ID Pasien	Nama	Dokter	Ruangan
PSN001	Ali Ahmad	Dr. Devon	A101
PSN002	Aisyah	Dr. Andi	B101
PSN003	Deru	Dr. Devon	A101

```

Tekan enter untuk kembali...█

```

Gambar 3. 10 Menu guest 2

Gambar tersebut merupakan tampilan dari salah satu menu yaitu lihat data pasien. Data tersebut tersaji di dalam tabel agar membuat data lebih mudah untuk dibaca.

3.2 Source Code

```

Sistem Manajemen Data Pasien Rumah Sakit

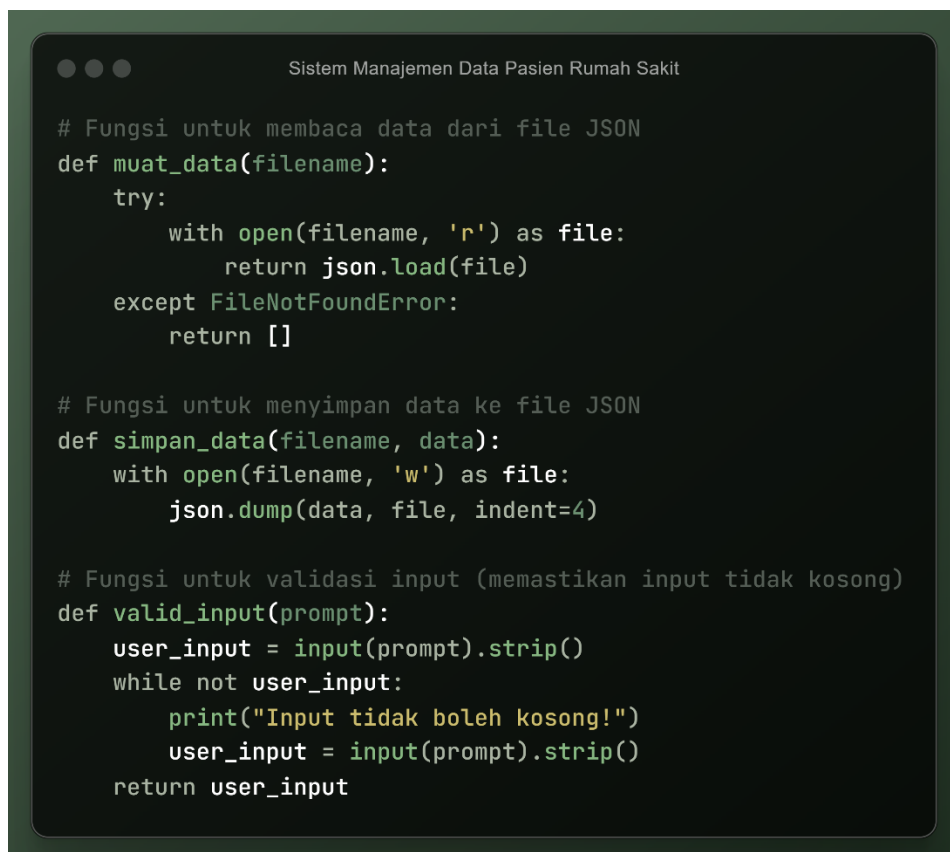
import json
import os
from tabulate import tabulate

data_akun = "data_akun.json"
data_pasien = "data_pasien.json"

```

Gambar 3. 11 Source code 1

- json digunakan untuk mengelola data dalam format JSON, seperti membaca dan menulis data ke dalam file JSON.
- os digunakan untuk membersihkan layar menggunakan perintah sistem, tergantung pada sistem operasi yang digunakan.
- tabulate digunakan untuk menampilkan data dalam bentuk tabel yang rapi.



```

Sistem Manajemen Data Pasien Rumah Sakit

# Fungsi untuk membaca data dari file JSON
def muat_data(filename):
    try:
        with open(filename, 'r') as file:
            return json.load(file)
    except FileNotFoundError:
        return []

# Fungsi untuk menyimpan data ke file JSON
def simpan_data(filename, data):
    with open(filename, 'w') as file:
        json.dump(data, file, indent=4)

# Fungsi untuk validasi input (memastikan input tidak kosong)
def valid_input(prompt):
    user_input = input(prompt).strip()
    while not user_input:
        print("Input tidak boleh kosong!")
        user_input = input(prompt).strip()
    return user_input

```

Gambar 3. 12 Source code 2

- Fungsi `muat_data` membaca data dari file JSON. Jika file tidak ditemukan, fungsi ini akan mengembalikan list kosong.
- Fungsi `simpan_data` menyimpan data dalam format JSON ke dalam file yang ditentukan.
- `valid_input` memastikan bahwa input pengguna tidak kosong. Jika kosong, pengguna diminta untuk memasukkan input lagi.

- `valid_password` memastikan bahwa password memiliki panjang minimal 8 karakter.
- `clear_screen` membersihkan layar terminal atau command prompt.
- `tampilkan_tabel` menampilkan data dalam bentuk tabel menggunakan `tabulate`.



```

Sistem Manajemen Data Pasien Rumah Sakit

# Fungsi untuk menu admin
def menu_admin(data_akun, data_pasien):
    while True:
        clear_screen()
        print("=== Menu Admin ===")
        print("1. Lihat Data Akun Dokter")
        print("2. Tambah Akun Dokter")
        print("3. Edit Akun Dokter")
        print("4. Hapus Akun Dokter")
        print("5. Lihat Data Guest")
        print("6. Hapus Akun Guest")
        print("7. Logout")
        pilih = input("Pilih opsi: ")

        if pilih == '1':
            # Tampilkan tabel akun dokter
            dokter_data = [akun for akun in data_akun if akun['role'] == 'dokter']
            dokter_tabel = [[akun['id'], akun['username'], akun['nama'], akun['spesialis'],
akun['ruangan']] for akun in dokter_data]
            tampilkan_tabel(dokter_tabel, ["ID Dokter", "Username", "Nama", "Spesialis",
"Ruangan"])
            input("Tekan enter untuk kembali...")
        elif pilih == '2':
            # Tambah akun dokter
            id_baru = f"DKT{len(data_akun) + 1:03d}"
            while True:
                username = valid_input("Username: ")
                # Cek apakah username sudah ada
                if any(akun['username'] == username for akun in data_akun):
                    print("Username telah terpakai. Silakan pilih username lain.")
                else:
                    break # Username valid, keluar dari loop
            password = valid_input("Password: ")
            while not valid_password(password):
                print("Password harus minimal 8 karakter!")
                password = valid_input("Password: ")
            nama = valid_input("Nama Dokter: ")
            spesialis = valid_input("Spesialis: ")
            ruangan = valid_input("Ruangan: ")

            dokter_baru = {
                "id": id_baru,
                "username": username,
                "password": password,
                "role": "dokter",
                "nama": nama,
                "spesialis": spesialis,
                "ruangan": ruangan
            }
            data_akun.append(dokter_baru)
            simpan_data("data_akun.json", data_akun)
            print("Akun dokter berhasil ditambahkan!")
            input("Tekan enter untuk kembali...")

```

Gambar 3. 13 Source code 3

```

elif pilih == '3':
    # Edit akun dokter
    dokter_data = [akun for akun in data_akun if akun['role'] == 'dokter']
    dokter_tabel = [[akun['id'], akun['username'], akun['nama'], akun['spesialis'],
    akun['ruangan']] for akun in dokter_data]
    tampilkan_tabel(dokter_tabel, ["ID Dokter", "Username", "Nama", "Spesialis",
    "Ruangan"])
    id_dokter = valid_input("Masukkan ID Dokter yang ingin diedit (DKTxxx): ").upper()
    dokter = next((akun for akun in data_akun if akun['id'] == id_dokter), None)

    if dokter:
        dokter['username'] = valid_input(f"Username ({dokter['username']}): ") or
        dokter['username']
        dokter['password'] = valid_input(f"Password ({dokter['password']}): ") or
        dokter['password']
        dokter['nama'] = valid_input(f>Nama Dokter ({dokter['nama']}): ") or
        dokter['nama']
        dokter['spesialis'] = valid_input(f"Spesialis ({dokter['spesialis']}): ") or
        dokter['spesialis']
        dokter['ruangan'] = valid_input(f"Ruangan ({dokter['ruangan']}): ") or
        dokter['ruangan']
        simpan_data("data_akun.json", data_akun)
        print("Akun dokter berhasil diedit!")
    else:
        print("Dokter tidak ditemukan!")
        input("Tekan enter untuk kembali...")
elif pilih == '4':
    # Hapus akun dokter
    dokter_data = [akun for akun in data_akun if akun['role'] == 'dokter']
    dokter_tabel = [[akun['id'], akun['username'], akun['nama'], akun['spesialis'],
    akun['ruangan']] for akun in dokter_data]
    tampilkan_tabel(dokter_tabel, ["ID Dokter", "Username", "Nama", "Spesialis",
    "Ruangan"])
    id_dokter = valid_input("Masukkan ID Dokter yang ingin dihapus (DKTxxx):
    ").upper()
    data_akun = [akun for akun in data_akun if akun['id'] != id_dokter]
    simpan_data("data_akun.json", data_akun)
    print("Akun dokter berhasil dihapus!")
    input("Tekan enter untuk kembali...")
elif pilih == '5':
    # Lihat data guest
    guest_data = [akun for akun in data_akun if akun['role'] == 'guest']
    guest_tabel = [[akun['id'], akun['username'], akun['password']] for akun in
    guest_data]
    tampilkan_tabel(guest_tabel, ["ID Guest", "Username", "Password"])
    input("Tekan enter untuk kembali...")
elif pilih == '6':
    # Hapus akun guest
    guest_data = [akun for akun in data_akun if akun['role'] == 'guest']
    guest_tabel = [[akun['id'], akun['username']] for akun in guest_data]
    tampilkan_tabel(guest_tabel, ["ID Guest", "Username"])
    id_guest = valid_input("Masukkan ID Guest yang ingin dihapus (GSTxxx): ").upper()
    data_akun = [akun for akun in data_akun if akun['id'] != id_guest]
    simpan_data("data_akun.json", data_akun)
    print("Akun guest berhasil dihapus!")
    input("Tekan enter untuk kembali...")
elif pilih == '7':
    break
else:
    print("Pilihan tidak tersedia.")

```

Gambar 3. 14 Source code 4

Menu ini digunakan oleh admin untuk mengelola akun dokter dan akun guest. Admin dapat melihat, menambah, mengedit, dan menghapus akun.

Pilihan:

- Lihat Data Akun Dokter: Menampilkan tabel akun-akun dokter.
- Tambah Akun Dokter: Admin dapat menambahkan akun dokter baru dengan memasukkan informasi seperti username, password, nama, spesialis, dan ruangan.
- Edit Akun Dokter: Admin dapat mengedit data akun dokter berdasarkan ID.
- Hapus Akun Dokter: Admin dapat menghapus akun dokter berdasarkan ID.
- Lihat Data Guest: Admin dapat melihat daftar akun guest.
- Hapus Akun Guest: Admin dapat menghapus akun guest berdasarkan ID.

```

# Fungsi untuk menu dokter
def menu_dokter(data_pasien, data_akun, current_user):
    while True:
        clear_screen()
        print("=== Menu Dokter ===")
        print("1. Lihat Data Pasien")
        print("2. Tambah Data Pasien")
        print("3. Edit Data Pasien")
        print("4. Hapus Data Pasien")
        print("5. Ganti Password")
        print("6. Logout")
        pilih = input("Pilih opsi: ")

        if pilih == '1':
            # Tampilkan tabel data pasien lengkap
            pasien_tabel = [
                [pasien['id'], pasien['nama'], pasien['jenis_kelamin'],
                 pasien['golongan_darah'], pasien['berat_badan'],
                 pasien['tinggi_badan'], pasien['diagnosa'], pasien['dokter'],
                 pasien['ruangan']] for pasien in data_pasien
            ]
            tampilkan_tabel(pasien_tabel, ["ID Pasien", "Nama", "Jenis Kelamin", "Golongan
            Darah", "Berat Badan", "Tinggi Badan", "Diagnosa", "Dokter", "Ruangan"])
            input("Tekan enter untuk kembali...")

        elif pilih == '2':
            # Tambah pasien
            id_baru = f"PSN{len(data_pasien) + 1:03d}"
            nama = valid_input("Nama Pasien: ")
            jenis_kelamin = valid_input("Jenis Kelamin (L/P): ").upper()
            while jenis_kelamin not in ['L', 'P']:
                print("Input tidak valid! Masukkan hanya 'L' untuk Laki-laki atau 'P' untuk
                Perempuan.")
            jenis_kelamin = valid_input("Jenis Kelamin (L/P): ").upper()
            golongan_darah = valid_input("Golongan Darah: ").upper()
            berat_badan = valid_input("Berat Badan: ")
            tinggi_badan = valid_input("Tinggi Badan: ")
            diagnosa = valid_input("Diagnosa: ")
            dokter = current_user['nama']
            ruangan = current_user['ruangan']

            pasien_baru = {
                "id": id_baru,
                "nama": nama,
                "jenis_kelamin": jenis_kelamin,
                "golongan_darah": golongan_darah,
                "berat_badan": berat_badan,
                "tinggi_badan": tinggi_badan,
                "diagnosa": diagnosa,
                "dokter": dokter,
                "ruangan": ruangan,
            }
            data_pasien.append(pasien_baru)
            simpan_data("data_pasien.json", data_pasien)
            print("Pasien berhasil ditambahkan!")
            input("Tekan enter untuk kembali...")

```

Gambar 3. 15 Source code 5

```

elif pilih == '3':
    # Edit pasien
    pasien_tabel = [
        [pasien['id'], pasien['nama'], pasien['jenis_kelamin'],
        pasien['golongan_darah'], pasien['berat_badan'],
        pasien['tinggi_badan'], pasien['diagnosa'], pasien['dokter'],
        pasien['ruangan']] for pasien in data_pasien
    ]
    tampilkan_tabel(pasien_tabel, ["ID Pasien", "Nama", "Jenis Kelamin", "Golongan
    Darah", "Berat Badan", "Tinggi Badan", "Diagnosa", "Dokter", "Ruangan"])
    id_pasien = valid_input("Masukkan ID Pasien yang ingin diedit (PSNxxx): ").upper()
    pasien = next((p for p in data_pasien if p['id'] == id_pasien), None)

    if pasien:
        # Pastikan pasien yang akan diedit ditangani oleh dokter yang sedang login
        if pasien['dokter'] == current_user['nama']:
            pasien['nama'] = valid_input(f>Nama Pasien ({pasien['nama']}): ") or
            pasien['nama']
            pasien['jenis_kelamin'] = valid_input(f>Jenis Kelamin
            ({pasien['jenis_kelamin']}): ").upper() or pasien['jenis_kelamin']
            while pasien['jenis_kelamin'] not in ['L', 'P']:
                print("Input tidak valid! Masukkan hanya 'L' untuk Laki-laki atau 'P'
                untuk Perempuan.")
            pasien['jenis_kelamin'] = valid_input("Jenis Kelamin (L/P): ").upper()
            pasien['golongan_darah'] = valid_input(f>Golongan Darah
            ({pasien['golongan_darah']}): ").upper() or pasien['golongan_darah']
            pasien['berat_badan'] = valid_input(f>Berat Badan
            ({pasien['berat_badan']}): ") or pasien['berat_badan']
            pasien['tinggi_badan'] = valid_input(f>Tinggi Badan
            ({pasien['tinggi_badan']}): ") or pasien['tinggi_badan']
            pasien['diagnosa'] = valid_input(f>Diagnosa ({pasien['diagnosa']}): ") or
            pasien['diagnosa']
            simpan_data("data_pasien.json", data_pasien)
            print("Data pasien berhasil diedit!")
        else:
            print("Anda tidak memiliki izin untuk mengedit data pasien ini.")
    else:
        print("Pasien tidak ditemukan!")
        input("Tekan enter untuk kembali...")
elif pilih == '4':
    # Hapus pasien
    pasien_tabel = [
        [pasien['id'], pasien['nama'], pasien['jenis_kelamin'],
        pasien['golongan_darah'], pasien['berat_badan'],
        pasien['tinggi_badan'], pasien['diagnosa'], pasien['dokter'],
        pasien['ruangan']] for pasien in data_pasien
    ]
    tampilkan_tabel(pasien_tabel, ["ID Pasien", "Nama", "Jenis Kelamin", "Golongan
    Darah", "Berat Badan", "Tinggi Badan", "Diagnosa", "Dokter", "Ruangan"])
    id_pasien = valid_input("Masukkan ID Pasien yang ingin dihapus (PSNxxx):
    ").upper()
    pasien = next((p for p in data_pasien if p['id'] == id_pasien), None)

    if pasien:
        # Pastikan pasien yang akan dihapus ditangani oleh dokter yang sedang login
        if pasien['dokter'] == current_user['nama']:
            data_pasien = [p for p in data_pasien if p['id'] != id_pasien]
            simpan_data("data_pasien.json", data_pasien)
            print("Data pasien berhasil dihapus!")
        else:
            print("Anda tidak memiliki izin untuk menghapus data pasien ini.")
    else:
        print("Pasien tidak ditemukan!")
        input("Tekan enter untuk kembali...")
elif pilih == '5':
    # Ganti password
    password_baru = valid_input("Password baru: ")
    while not valid_password(password_baru):
        print("Password harus minimal 8 karakter!")
        password_baru = valid_input("Password baru: ")
    current_user['password'] = password_baru
    simpan_data("data_akun.json", data_akun)
    print("Password berhasil diganti!")
    input("Tekan enter untuk kembali...")
elif pilih == '6':
    break
else:
    print("Pilihan tidak tersedia.")

```

Gambar 3. 16 Source code 6

Menu ini digunakan oleh dokter untuk mengelola data pasien.

Pilihan:

- Lihat Data Pasien: Menampilkan data pasien yang tercatat.
- Tambah Data Pasien: Dokter dapat menambah pasien baru dengan memasukkan data seperti nama, jenis kelamin, golongan darah, berat badan, tinggi badan, diagnosa, dan ruangan.
- Edit Data Pasien: Dokter dapat mengedit data pasien yang mereka tangani berdasarkan ID pasien.
- Hapus Data Pasien: Dokter dapat menghapus data pasien yang mereka tangani berdasarkan ID pasien.
- Ganti Password: Dokter dapat mengganti password mereka.

```
Sistem Manajemen Data Pasien Rumah Sakit

# Fungsi untuk menu guest
def menu_guest(data_pasien, current_user):
    while True:
        clear_screen()
        print("=== Menu Guest ===")
        print("1. Lihat Data Pasien")
        print("2. Ganti Password")
        print("3. Logout")
        pilih = input("Pilih opsi: ")

        if pilih == '1':
            # Tampilkan tabel data pasien
            pasien_tabel = [
                [pasien['id'], pasien['nama'], pasien['dokter'], pasien['ruangan']] for pasien
                in data_pasien
            ]
            tampilkan_tabel(pasien_tabel, ["ID Pasien", "Nama", "Dokter", "Ruangan"])
            input("Tekan enter untuk kembali...")
        elif pilih == '2':
            # Ganti password
            password_baru = valid_input("Password baru: ")
            while not valid_password(password_baru):
                print("Password harus minimal 8 karakter!")
                password_baru = valid_input("Password baru: ")
            current_user['password'] = password_baru
            simpan_data("data_akun.json", data_akun)
            print("Password berhasil diganti!")
            input("Tekan enter untuk kembali...")
        elif pilih == '3':
            break
        else:
            print("Pilihan tidak tersedia.")
```

Gambar 3. 17 Source code 7

Menu ini digunakan oleh guest untuk melihat data pasien dan mengganti password.

Pilihan:

- Lihat Data Pasien: Guest dapat melihat informasi pasien dasar seperti ID, nama, dokter, dan ruangan.
- Ganti Password: Guest dapat mengganti password mereka.

```
Sistem Manajemen Data Pasien Rumah Sakit

# Fungsi login
def login(data_akun):
    kesempatan = 3 # Inisialisasi jumlah percobaan login
    while kesempatan > 0:
        clear_screen()
        print ("== Menu Login ==")
        username = valid_input("Username: ")
        password = valid_input("Password: ")

        user = next((akun for akun in data_akun if akun['username'] == username and
akun['password'] == password), None)
        if user:
            return user
        else:
            kesempatan -= 1 # Menambah jumlah percobaan login
            print(f"Username atau password salah. Sisa percobaan : {kesempatan}")
            input("Tekan enter untuk melanjutkan...")
            if kesempatan == 0:
                print("Terlalu banyak percobaan login yang gagal. Program akan berhenti.")
                exit() # Menghentikan program setelah 3 percobaan gagal
```

Gambar 3. 18 Source code 8

Login memungkinkan pengguna untuk masuk menggunakan username dan password. Pengguna memiliki 3 kesempatan untuk mencoba login. Jika gagal 3 kali, program akan berhenti.

```
Sistem Manajemen Data Pasien Rumah Sakit

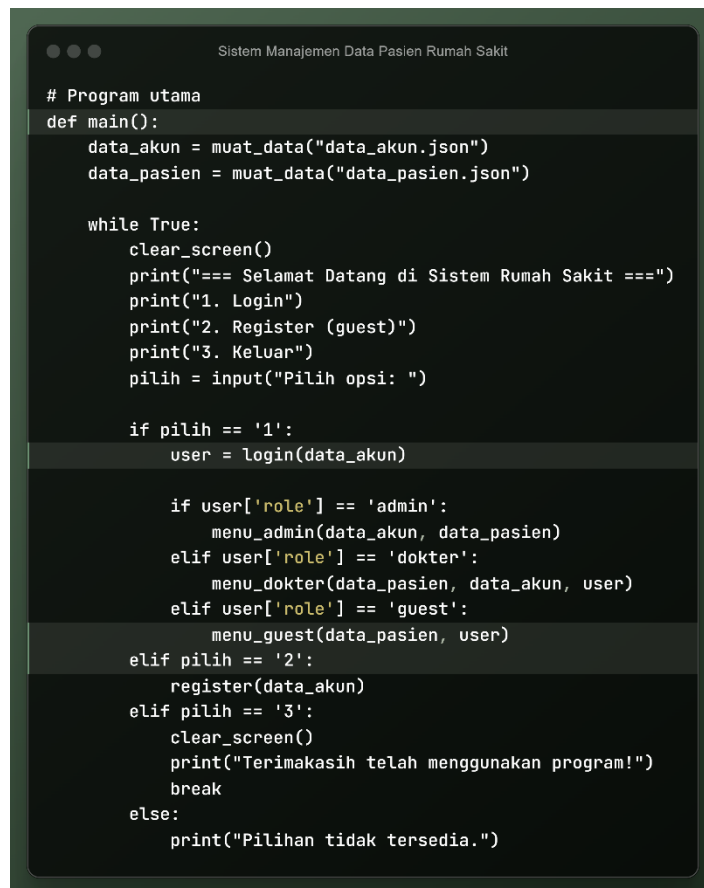
# Fungsi untuk register
def register(data_akun):
    clear_screen()
    print("=== Register ===")
    id_baru = f"6ST{len(data_akun) + 1:03d}"
    while True:
        username = valid_input("Username: ")
        # Cek apakah username sudah ada
        if any(akun['username'] == username for akun in data_akun):
            print("Username telah terpakai. Silakan pilih username lain.")
        else:
            break # Username valid, keluar dari loop

    password = valid_input("Password: ")
    while not valid_password(password):
        print("Password harus minimal 8 karakter!")
        password = valid_input("Password: ")

    guest_baru = {
        "id": id_baru,
        "username": username,
        "password": password,
        "role": "guest",
    }
    data_akun.append(guest_baru)
    simpan_data("data_akun.json", data_akun)
    print("Akun guest berhasil dibuat!")
    input("Tekan enter untuk kembali...")
```

Gambar 3. 19 Source code 9

Register memungkinkan pengguna untuk membuat akun baru dengan memilih username dan password, serta memastikan username yang dimasukkan belum terdaftar. Untuk password harus memenuhi syarat yaitu minimal 8 karakter. Untuk register dikhususkan untuk guest.



```
# Program utama
def main():
    data_akun = muat_data("data_akun.json")
    data_pasien = muat_data("data_pasien.json")

    while True:
        clear_screen()
        print("=== Selamat Datang di Sistem Rumah Sakit ===")
        print("1. Login")
        print("2. Register (guest)")
        print("3. Keluar")
        pilih = input("Pilih opsi: ")

        if pilih == '1':
            user = login(data_akun)

            if user['role'] == 'admin':
                menu_admin(data_akun, data_pasien)
            elif user['role'] == 'dokter':
                menu_dokter(data_pasien, data_akun, user)
            elif user['role'] == 'guest':
                menu_guest(data_pasien, user)
        elif pilih == '2':
            register(data_akun)
        elif pilih == '3':
            clear_screen()
            print("Terimakasih telah menggunakan program!")
            break
        else:
            print("Pilihan tidak tersedia.")
```

Main adalah fungsi utama yang mengatur alur program. Pengguna dapat memilih untuk login, mendaftar, atau keluar dari program. Dan akan dilanjutkan sesuai dengan pilihan.

BAB IV

PENUTUP

4.1 Kesimpulan

Adapun kesimpulan yang diperoleh adalah sebagai berikut.

1. Sistem ini mampu mengelola data akun pengguna dengan baik, memisahkan peran setiap pengguna (admin, dokter, dan guest) berdasarkan hak akses yang telah ditentukan. Admin memiliki hak penuh untuk mengelola data pengguna dan pasien, sementara dokter hanya dapat mengelola data pasien yang menjadi tanggung jawabnya. Guest hanya dapat melihat informasi terbatas, yang memastikan keamanan dan privasi data di dalam sistem.
2. Dokter hanya dapat mengakses dan mengelola data pasien yang tercatat di bawah tanggung jawabnya. Hal ini diimplementasikan dengan memanfaatkan identifikasi ID dokter yang terkait langsung dengan data pasien. Pembatasan ini menjamin bahwa dokter hanya bisa mengubah atau melihat informasi yang relevan dengan pasien yang sedang mereka tangani.
3. Sistem ini dikembangkan dengan antarmuka berbasis terminal yang sederhana namun fungsional. Pengguna dapat dengan mudah mengakses berbagai fitur seperti penambahan data pasien, pengelolaan akun pengguna, dan melihat informasi penting lainnya. Meskipun antarmuka berbasis teks, sistem ini cukup efisien dan mudah digunakan oleh pengguna yang memiliki akses sesuai dengan perannya.


4.2 Saran

1. Untuk mempermudah interaksi dengan sistem, disarankan untuk mengembangkan antarmuka grafis (GUI) yang lebih interaktif dan user-friendly.
2. Penambahan fitur pelengkap seperti fitur pencarian pasien berdasarkan nama, ID pasien, atau diagnosis untuk memudahkan dokter dalam mengakses data pasien.













DAFTAR PUSTAKA

- Hermans, Felienne, and Tijs van der Storm. "TrueGrid: Code the table, tabulate the data." Federation of International Conferences on Software Technologies: Applications and Foundations. Cham: Springer International Publishing, 2016.
- Liu, Zhen Hua, Beda Hammerschmidt, and Doug McMahon. "JSON data management: supporting schema-less development in RDBMS." Proceedings of the 2014 ACM SIGMOD international conference on Management of data. 2014.

LAMPIRAN

<p>Aturan Konsultasi :</p> <p>A. Kartu Konsul wajib dibawa saat dilakukan konsultasi</p> <p>B. Ketua Kelompok dan Anggota diwajibkan untuk hadir tiap konsultasi</p> <p>C. Konsul dilaksanakan minimal 2 kali dengan ketentuan sebagai berikut :</p> <ul style="list-style-type: none"> - Konsul 1 : Konsep Program - Konsul 2 : Penyelesaian Program <p>N.B : Batas waktu konsultasi ialah H+7 untuk konsul 1 dan H+14 untuk konsul 2 (Dimulai sejak pengumuman dan pembentukan kelompok PA)</p>	<p style="text-align: center;">Manajemen Data Pasien Rawat Inap Rumah Sakit</p> <p style="text-align: center;">Algoritma dan Pemrograman Dasar</p> <p style="text-align: center;">KELAS IF B1 24:</p> <p style="text-align: center;">KELOMPOK 1:</p> <p style="text-align: center;">Andi Fairul Amri (65) (Ketua)</p> <p style="text-align: center;">Muhammad Britama Putra Jaya (47)</p> <p style="text-align: center;">Devon Falen Pasae (55)</p> <div style="text-align: center;">  </div> <p style="text-align: center;">LABORATORIUM FAKULTAS TEKNIK</p> <p style="text-align: center;">UNIVERSITAS MULAWARMAN</p> <p style="text-align: center;">2024</p>
--	--

Lampiran 1 Lembar Konsul (Cover)

<p>Tanggal Konsultasi : 3 NOVEMBER 2024</p> <p>Uraian / Pembahasan :</p> <p>- Flowchart</p> <p>- Program</p> <ul style="list-style-type: none"> • Buat multi user (dokter, admin, tamu) • Tamu hanya dapat melihat • Dokter hanya dapat mengedit pasien yang ia buat. <table border="1" style="width: 100%; margin-top: 20px;"> <tr> <td style="width: 50%; text-align: center;"> <p>Asisten Lab</p>  <p>Nama: FANU</p> </td> <td style="width: 50%; text-align: center;"> <p>Ketua Kelompok</p>  <p>Nama: ANDI FAIRUL AMRI</p> </td> </tr> </table>	<p>Asisten Lab</p>  <p>Nama: FANU</p>	<p>Ketua Kelompok</p>  <p>Nama: ANDI FAIRUL AMRI</p>	<p>Tanggal Konsultasi : 14 NOVEMBER 2024</p> <p>Uraian / Pembahasan :</p> <ul style="list-style-type: none"> - styling border tabel - admin hanya menambah dokter, untuk pasien seluruhnya hak akses dokter. - Fitur edit password untuk dokter - Clear screen - Fitur adminapus data dokter. <table border="1" style="width: 100%; margin-top: 20px;"> <tr> <td style="width: 50%; text-align: center;"> <p>Asisten Lab</p>  <p>Nama: FANU</p> </td> <td style="width: 50%; text-align: center;"> <p>Ketua Kelompok</p>  <p>Nama: ANDI FAIRUL AMRI</p> </td> </tr> </table>	<p>Asisten Lab</p>  <p>Nama: FANU</p>	<p>Ketua Kelompok</p>  <p>Nama: ANDI FAIRUL AMRI</p>
<p>Asisten Lab</p>  <p>Nama: FANU</p>	<p>Ketua Kelompok</p>  <p>Nama: ANDI FAIRUL AMRI</p>				
<p>Asisten Lab</p>  <p>Nama: FANU</p>	<p>Ketua Kelompok</p>  <p>Nama: ANDI FAIRUL AMRI</p>				

Lampiran 2 Lembar Konsul (Isi)