

MODUL DETEKSI PENYAKIT BERBASIS WEBSITE

1. *Happy Syahrul Ramadhan*
NIM. 122450013

2. *Esteria Rohanauli Sidauruk*
NIM. 122450025

3. *Irhamna Mahdi*
NIM. 122450049

4. *Salwa Farhanatussaidah*
NIM. 122450055

5. *Ibrahim Al-kahfi*
NIM. 122450100

ABSTRAK

Penelitian ini membahas tentang pengembangan dan implementasi dari “modul deteksi penyakit berbasis website” yang menjadi bagian dari tugas besar. Modul ini dirancang untuk memfasilitasi proses deteksi penyakit secara cepat dan efisien menggunakan platform web yang mudah diakses. Dengan memanfaatkan teknologi terkini dalam bidang kecerdasan buatan, modul ini dapat mengidentifikasi penyakit tertentu berdasarkan gejala yang diinputkan oleh pengguna.

Pengguna dapat mengakses modul melalui antarmuka web, di mana mereka dapat memasukkan gejala yang mereka alami. Proses deteksi dilakukan menggunakan algoritme kecerdasan buatan yang telah terlatih dengan dataset penyakit yang tepat. Hasil deteksi akan disajikan dengan jelas kepada pengguna, ini akan mencakup informasi tambahan tentang penyakit yang ditemukan, saran untuk pengobatan, dan saran untuk tindakan selanjutnya.

Tujuan implementasi modul ini adalah untuk memberikan solusi yang mudah diakses bagi orang-orang yang mencari informasi tentang kesehatan mereka atau yang ingin melakukan pengecekan dini untuk mengidentifikasi kemungkinan penyakit. Modul ini diharapkan dapat memberikan informasi yang akurat dan relevan secara cepat dengan menggabungkan keunggulan teknologi web dan kecerdasan buatan untuk memberikan kontribusi positif terhadap pelayanan kesehatan masyarakat. Dengan menggunakan teknologi informasi yang inovatif, penelitian ini diharapkan dapat menjadi landasan untuk pengembangan lebih lanjut dalam pemeliharaan kesehatan masyarakat.

keyword : Penyakit, teknologi, algoritma.

I. Pendahuluan

A. Latar Belakang

Metode kreatif untuk memelihara kesehatan masyarakat menjadi semakin penting di era teknologi informasi yang semakin berkembang pesat. Akses yang cepat dan efektif terhadap informasi kesehatan yang akurat adalah salah satu tantangan utama yang dihadapi. Untuk meningkatkan efektivitas dan aksesibilitas informasi kesehatan, penggunaan teknologi web dan kecerdasan buatan semakin populer. Modul deteksi penyakit berbasis web menawarkan solusi yang menarik untuk membantu mendeteksi penyakit sejak dini. Menurut studi yang dilakukan oleh World Health Organization (WHO), teknologi informasi dapat meningkatkan

kualitas dan kemudahan pelayanan kesehatan (WHO, 2020).

Dalam konteks ini, pengembangan "Modul Deteksi Penyakit Berbasis Website" muncul sebagai solusi potensial. Fokus modul ini adalah untuk membuat platform yang cepat dan akurat untuk memberikan informasi kesehatan yang mudah diakses dengan memanfaatkan kekuatan teknologi web, kecerdasan buatan, dan pemrosesan citra. Kemajuan besar telah dibuat dalam penerapan kecerdasan buatan untuk mendukung diagnosis medis (Lipton et al., 2015). Diharapkan modul ini dapat mengidentifikasi penyakit berdasarkan gejala yang diberikan oleh pengguna melalui integrasi algoritma kecerdasan buatan dan dataset penyakit yang relevan.

Tujuan dari laporan tugas besar ini adalah untuk mempelajari potensi dan efek dari penggunaan Modul Deteksi Penyakit Berbasis Website dalam hal meningkatkan akses ke informasi kesehatan, menyediakan metode deteksi dini yang lebih efisien, dan memberikan dampak positif pada pelayanan kesehatan masyarakat secara keseluruhan. Diharapkan modul ini dapat membantu memahami kebutuhan masyarakat untuk pemeliharaan kesehatan dengan memahami teknologi informasi yang inovatif dalam sektor kesehatan.

B. Masalah

Berdasarkan pembahasan yang diangkat maka rumusan masalahnya sebagai berikut:

- Apa saja masalah-masalah yang masih menghambat aksesibilitas dan kecepatan deteksi penyakit, meskipun telah terjadi kemajuan dalam teknologi medis?
- Bagaimana informasi teknologi dapat memberikan kontribusi dalam mendeteksi penyakit dengan lebih akurat dan tepat waktu?
- Bagaimana modul Deteksi Penyakit Berbasis Website dapat diimplementasikan untuk meningkatkan akurasi dan keandalan proses deteksi penyakit dalam konteks pelayanan kesehatan?

C. Deskripsi Data

Deskripsi data merupakan gambaran data yang digunakan dalam suatu laporan. Pada laporan yang berjudul “modul deteksi penyakit berbasis website” menggunakan data berupa kumpulan diagnosa penyakit yang kami susun dan merupakan jenis data domain name service yang berisi domain text untuk sumber diluar domain. Pada data tersebut berisi 3 jenis gejala penyakit dan jenis penyakitnya. Berikut merupakan bentuk data yang digunakan:

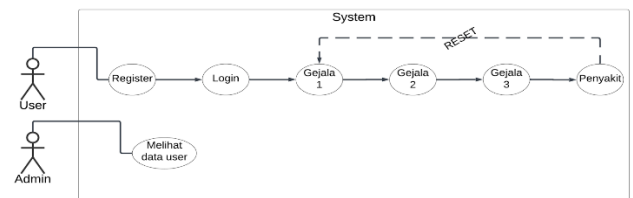
No.	Nama Penyakit	Gejala		
		1	2	3
1	Gastritis	Mual	Perut kembung	Nyeri perut
2	Kanker ovarium	Mual	Perut kembung	Nyeri perut
3	Ulkus peptikum	Mual	Perut kembung	Nyeri ulu hati
4	Tumor otak	Mual	Sakit kepala	Gangguan penglihatan
5	Galukoma	Mual	Sakit kepala	Gangguan penglihatan
6	Pankreatitis	Mual	Demam	Mata menguning
7	Apendisitis	Mual	Demam	Nyeri ulu hati
8	Kolesistitis	Mual	Demam	Nyeri perut
9	GERD	Mual	Batuk	Sakit tenggorokan
10	Gagal ginjal	Mual	Lemas	Nyeri punggung
11	Batu ginjal	Mual	Demam	Urine berbau busuk
12	Pneumonia	Sesak napas	Batuk	Demam
13	Kanker paru	Sesak napas	Batuk	Nyeri dada
14	Asma	Sesak napas	Batuk	Lemas
15	Kolesterol	Sesak napas	Nyeri dada	Kesemutan
16	Jantung koroner	Sesak napas	Nyeri dada	Mudah lelah
17	Aritmia	Sesak napas	Nyeri dada	Mudah lelah
18	Bronkitis	Sesak napas	Sakit kepala	Demam

II. Rancangan Sistem

A. Gambaran Umum

Pada laporan yang berjudul “Modul Deteksi Penyakit Berbasis Website” memiliki gambaran umum Pengguna dapat dengan mudah mendeteksi penyakit yang dialami berdasarkan gejalanya. Web akan menyediakan beberapa pilihan gejala yang dialami hingga memunculkan deteksi penyakit yang sesuai berdasarkan gejalanya. Deteksi penyakit didukung oleh algoritma kecerdasan buatan yang telah terlatih dengan dataset penyakit yang relevan, meningkatkan kemampuan sistem untuk mengenali pola-pola penyakit dengan akurat. Laporan tugas besar ini akan secara menyeluruh mengulas perancangan, implementasi, dan evaluasi Modul Deteksi Penyakit Berbasis Website. Dengan demikian, diharapkan modul ini dapat memberikan kontribusi positif dalam pemeliharaan kesehatan masyarakat melalui pemanfaatan teknologi informasi yang inovatif.

B. Use Case



User

- Membuat akun
- Login dengan akun yang telah dibuat
- Memasukkan gejala yang ingin dicari tahu
- Memasukkan gejala 2 yang berhubungan dengan gejala 1

- Memasukkan gejala 3 yang berhubungan dengan gejala 2 dan 1
- User dapat melihat penyakit berdasarkan pendekatan 3 gejala
- User dapat mereset ulang dan mulai dari memasukkan gejala 1 kembali

Admin

- Admin dapat melihat data-data user yang telah mendaftarkan akunnya

Pada sisi pengguna, aplikasi ini memungkinkan mereka untuk menciptakan pengalaman interaktif dengan beberapa langkah yang terstruktur. Pertama, pengguna dapat membuat akun untuk mengakses layanan. Setelah berhasil login, mereka dapat memasukkan gejala yang ingin dicari tahu. Proses ini tidak berhenti di situ, karena pengguna dapat melanjutkan dengan memasukkan gejala 2 yang berhubungan dengan gejala 1, dan kemudian gejala 3 yang berhubungan dengan kedua gejala sebelumnya. Hasilnya adalah kemampuan untuk melihat informasi penyakit yang mungkin berkaitan dengan kombinasi gejala yang dimasukkan. Apabila pengguna ingin mengulang atau mencoba pendekatan baru, ada fitur reset yang memungkinkan mereka kembali ke tahap awal, yaitu memasukkan gejala 1.

Dari perspektif admin, aplikasi ini memberikan kontrol dan visibilitas terhadap data pengguna. Admin dapat dengan mudah melihat informasi yang berkaitan dengan pengguna yang telah mendaftar akun. Ini memberikan kesempatan bagi admin untuk mengelola dan menganalisis data pengguna, serta memberikan dukungan lebih lanjut jika diperlukan. Ketersediaan data ini membantu admin dalam memahami tren pengguna, meningkatkan pengalaman pengguna, dan mengoptimalkan operasional aplikasi secara keseluruhan. Dengan demikian, aplikasi ini menciptakan ekosistem yang seimbang antara pengguna dan admin, memberikan pengalaman yang informatif dan terstruktur bagi pengguna, sambil memberikan kontrol yang efisien bagi admin.

C. Dependency / Library Yang Digunakan

- *Streamlit*

Streamlit adalah sebuah kerangka kerja sumber terbuka yang digunakan untuk membangun aplikasi web interaktif dengan mudah menggunakan bahasa pemrograman Python. Tujuan utama dari streamlit adalah menyederhanakan proses pengembangan aplikasi web, memungkinkan pembuatnya, termasuk mereka yang tidak berpengalaman dalam pengembangan web, untuk membuat aplikasi web dengan cepat.

- *Pandas*

Pandas adalah *library open source* pada Python yang sering digunakan untuk memproses data yang meliputi pembersihan data, manipulasi data, hingga melakukan analisis data. Ketika melakukan suatu analisis, kita tidak bisa menggunakan data mentah. Data mentah harus diproses lebih dulu hingga layak untuk dianalisis. Untuk tahap tersebut disebut juga dengan data *wrangling*. *Wrangling* merupakan proses dimana data di manajemen dan dibentuk menjadi lebih tertata. Proses ini cukup krusial karena diperlukan ketelitian dan harus bisa menjawab permasalahan yang ingin diselesaikan. *Library* Pandas dapat mengolah data berbentuk csv, txt, excel, html, dan sebagainya.

- *Hashlib*

Hashlib adalah sebuah modul dalam bahasa pemrograman python yang menyediakan fungsi untuk menghasilkan nilai hash (nilai acak yang dihasilkan dari data) dengan menggunakan berbagai algoritma hash yang umum digunakan. Algoritma hash digunakan untuk mengubah data (seperti string, file, atau objek lain) menjadi nilai hash dengan panjang tetap, yang sering digunakan untuk mengidentifikasi data atau untuk tujuan keamanan.

D. Daftar Fungsionalitas / Fitur

Fungsionalitas adalah sesuatu yang berisi tentang proses-proses apa saja yang nantinya akan dilakukan oleh sistem. Fungsionalitas juga berisikan informasi-informasi apa saja yang harus ada dan dihasilkan oleh sistem.

Berikut ini adalah daftar fungsionalitas dari “**Modul Deteksi Penyakit Berbasis Website**” yaitu sebagai berikut :

- **Registrasi User**

Berfungsi untuk melakukan registrasi data sebelum masuk kedalam aplikasi

- **Login User**

Setelah melakukan registrasi user dapat melakukan login dan masuk ke dalam aplikasi

- **Deklarasi Data**

Berfungsi untuk mendeklarasikan data yang sudah di susun sebelumnya, dalam hal ini data kami susun menjadi sebuah dictionary bercabang untuk menentukan keterkaitan antara gejala 1 dengan gejala lainnya. kemudian mendeklarasikan juga kumpulan jawaban untuk menentukan jawaban tersebut termasuk kedalam suatu penyakit tertentu.

- **Memberikan pertanyaan pertama dan memberikan rekomendasi gejala**

berfungsi untuk menampilkan pertanyaan dan memberikan pilihan gejala dari yang paling dasar.

- **Memberikan pertanyaan kedua dan memberikan rekomendasi gejala**

berfungsi untuk menampilkan pertanyaan kedua dan memberikan pilihan gejala berdasarkan dari pilihan gejala pada pertanyaan pertama

- **Memberikan pertanyaan ketiga dan memberikan rekomendasi gejala**

berfungsi untuk menampilkan pertanyaan ketiga dan memberikan pilihan gejala yang berkaitan dari gejala dari pertanyaan pertama dan kedua

- **Memberikan output**

Setelah ketiga pertanyaan dijawab berdasarkan gejala yang di rasakan maka ketika user menekan tombol submit akan muncul output berupa jenis penyakit berdasarkan ketiga gejala yang saling berkaitan dan akan menampilkan juga pengertian dari penyakit tersebut dan cara pengobatannya.

III. Hasil Program

1. mulai import data pustaka python yang diperlukan untuk membuat register dan login system

```
pages > Login.py > ...
1 import streamlit as st
2 import pandas as pd
3 import hashlib
4
```

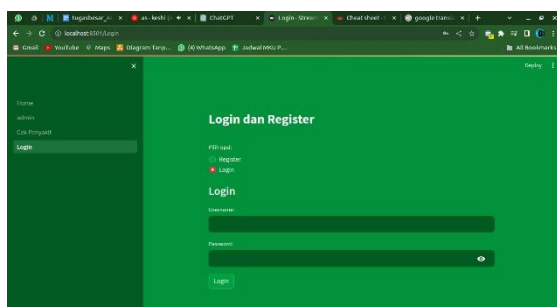
2. menginisialisasi session_state yang di perlukan

```
# inisialisasi state
if 'pagelogin' not in st.session_state:
    st.session_state.pagelogin = False
```

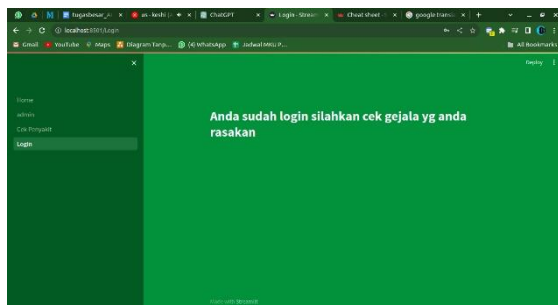
3. melakukan pengecekan apakah session state dari pagelogin true atau false jika true maka akan menampilkan output “anda sudah login silahkan cek gejala yg anda rasakan”, kemudian jika false maka akan menampilkan form login atau register

```
9
10 if st.session_state.pagelogin:
11     st.title("Anda sudah login silahkan cek gejala yg anda rasakan")
12
13 else:
14     # Fungsi untuk membaca data pengguna dari file CSV
15     def read_csv():
16         try:
17             user_data = pd.read_csv('user_data.csv')
18         except FileNotFoundError:
19             user_data = pd.DataFrame(columns=['username', 'password'])
20     return user_data
```

Output jika state dari pagelogin False :



Output jika state dari pagelogin True :



4. sebelum membuat fungsi login dan registrasi perlu membuat fungsi agar ketika user melakukan registrasi datanya masuk ke dalam file csv agar datanya mudah di akses untuk keperluan lain seperti bisa melakukan pengecekan agar username yang di input tidak boleh sama dengan yang lain

```
14 # Fungsi untuk membaca data pengguna dari file CSV
15 def read_csv():
16     try:
17         user_data = pd.read_csv('user_data.csv')
18     except FileNotFoundError:
19         user_data = pd.DataFrame(columns=['username', 'password'])
20     return user_data
21
22 # Fungsi untuk menulis data pengguna ke file CSV
23 def write_csv(data):
24     data.to_csv('user_data.csv', index=False)
25
```

5. melakukan hasing password agar ketika user melakukan pengisian password maka passwordnya akan di enkripsi untuk pencegahan hal yang tidak di inginkan

```
26 # Fungsi untuk melakukan hashing password
27 def hash_password(password):
28     return hashlib.sha256(password.encode()).hexdigest()
29
30 # Fungsi untuk mengecek password
31 def check_password(input_password, hashed_password):
32     return hash_password(input_password) == hashed_password
33
```

6. Kemudian membuat fungsi registrasi

```
# Fungsi untuk registrasi pengguna
def register(username, password):
    user_data = read_csv()

    # Memeriksa apakah username sudah ada
    if username in user_data['username'].values:
        st.warning("Username sudah digunakan. Silakan pilih username lain.")
        return

    # Menambahkan pengguna baru
    hashed_password = hash_password(password)
    new_user = pd.DataFrame({'username': [username], 'password': [hashed_password]})
    user_data = pd.concat([user_data, new_user], ignore_index=True)

    # Menyimpan data ke file CSV
    write_csv(user_data)
    st.session_state.login = True
    st.success("Register berhasil.")
```

7. Setelah membuat fungsi registrasi kemudian membuat fungsi login

```
# Fungsi untuk login pengguna
def login(username, password):
    st.session_state.username_login = username
    st.session_state.password_login = password
    user_data = read_csv()

    # Memeriksa apakah username dan password cocok
    if any((user_data['username'] == st.session_state.username_login) & (user_data['password'] == hash_password(st.session_state.password_login))):
        st.success("Login berhasil.")
        st.session_state.login = True
        st.session_state.pagelogin = True
    else:
        st.error("Login gagal. Periksa kembali username dan password.")
```

8. kemudian membuat program pada halaman utama, yang pertama menginisialisasi data gejala menggunakan tipe data dictionary bercabang yang

```
data = {
    "Mual": {
        "Perut Kembung": ["Nyeri Perut", "Nyeri ulu Hati",
            "Sakit Kepala": ["Gangguan Penglihatan"],
            "Demam": ["Mata Menguning", "Nyeri ulu Hati", "Nyeri Perut", "Urine Berbau Busuk"],
            "Batuk": ["Sakit Tenggorokan"],
            "Lemas": ["Nyeri Punggung"],
        },
        "Sesak Napas": {
            "Batuk": ["Demam", "Nyeri Dada", "Lemas"],
            "Nyeri Dada": ["Kesemutan", "Mudah Lelah"],
            "Sakit Kepala": ["Demam", "Gangguan Penglihatan"],
            "Diare": ["Kesemutan"],
        },
        "Mudah Lelah": {
            "Nyeri Perut": ["Lemas", "Mata Menguning"],
            "Batuk": ["Berkeringat di Malam Hari"],
            "Nyeri Sendi": ["Ruam Pada Kulit"],
            "Sakit Kepala": ["Kulit Pucat"],
        },
        "Diare": {
            "Sakit Kepala": ["Nyeri Otot"],
            "Penurunan Berat Badan": ["Tinja Berminyak"],
        },
    },
}
```

9. kemudian menginisialisasi data jenis penyakit yang akan keluar dari 3 gejala yang sudah dipilih menggunakan tipe data dictionary bercabang

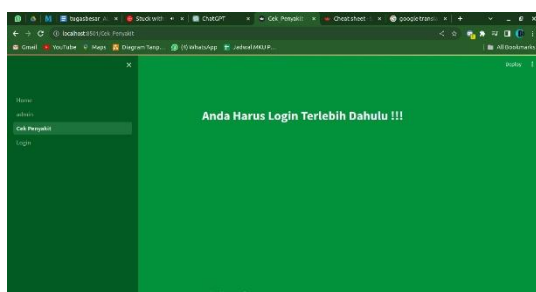
```
OutputData = {
    "Mual", "Perut Kembung", "Nyeri Perut": {
        "message": "Kemungkinan anda mengalami gejala Gastritis",
        "details": "Gastritis adalah peradangan pada dinding lambung yang dapat disebabkan oleh berba",
    },
    "Mual", "Perut Kembung", "Nyeri ulu Hati": {
        "message": "Kemungkinan anda mengalami Kanker Ovarium",
        "details": "Kanker ovarium adalah jenis kanker yang bermula dari ovarium atau indung telur, o",
    },
    "Mual", "Perut Kembung", "Nyeri ulu Hati": {
        "message": "Kemungkinan anda mengalami gejala Ulcus Peptikum",
        "details": "Ulkus peptikum, atau yang sering disebut sebagai tukak lambung atau tukak usus du",
    },
    "Mual", "Sakit Kepala", "Gangguan Penglihatan": {
        "message": "Kemungkinan anda mengalami gejala Tumor Otak",
        "details": "Tumor otak adalah pertumbuhan abnormal dari sel-sel di otak atau di sekitarnya. T",
    },
}
```

10. kemudian menginisialisasi dan mengecek apakah user sudah login atau belum, jika belum akan menampilkan output "silahkan login terlebih dahulu" kemudian jika user sudah login akan menampilkan halaman untuk memilih gejala penyakitnya

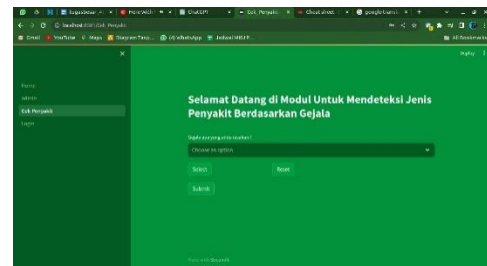
```
if 'login' not in st.session_state:
    st.session_state.login = False

if st.session_state.login:
    class HealthCheckerApp:
        def __init__(self, data, output_data):
            self.data = data
            self.output_data = output_data
            self.is_clicked = False
            self.selected_data = []
```

Output jika statenya False



Output jika statenya True :



11. Masuk ke dalam program utamanya, yang pertama kita membuat class bernama HealthCheckerApp kemudian memberikan parameter data dan output data lalu menambahkan juga variabel is_clicked = false dan variabel selected_data yang merupakan list kosong untuk menampung jawaban dari user

```
if st.session_state.login:
    class HealthCheckerApp:
        def __init__(self, data, output_data):
            self.data = data
            self.output_data = output_data
            self.is_clicked = False
            self.selected_data = []
```

12. Membuat fungsi untuk response dari user yaitu 3 pertanyaan dan list berupa jawaban yang saling berkaitan, jawaban dari setiap pertanyaan tersebut datanya didapat dari data dictionary di atas

```
def selectData(self):
    st.title('Selamat Datang di Modul Untuk Mendeteksi Jenis Penyakit Berdasarkan Gejala')

    if not self.is_clicked:
        self.selected_data = []

    if not hasattr(st.session_state, 'is_clicked'):
        st.session_state.is_clicked = False

    if not hasattr(st.session_state, 'data'):
        st.session_state.data = []

    st.session_state.q1 = st.multiselect("Gejala apa yang anda rasakan ?", list(self.data.keys()))
    if len(st.session_state.data) == 0:
        st.session_state.data.extend(st.session_state.q1)

    if st.session_state.q1:
        st.session_state.q2 = st.multiselect(f"Gejala apa yang anda rasakan setelah {st.session_state.q1}?", list(self.data.keys()))
        if len(st.session_state.data) == 1:
            st.session_state.data.extend(st.session_state.q2)

    if st.session_state.q2:
        st.session_state.q3 = st.multiselect(f"Gejala apa yang anda rasakan setelah {st.session_state.q1} {st.session_state.q2}?", list(self.data.keys()))
        if len(st.session_state.data) == 2:
            st.session_state.data.extend(st.session_state.q3)
```

Output :



13. Membuat fungsi untuk menampilkan jenis penyakit dari 3 jawaban yang di berikan oleh user

```
def display(self):
    col1, col2, col3 = st.columns(3)

    count = 0
    if col1.button("Select"):
        self.is_clicked = True

    if col2.button("Reset"):
        st.session_state.data.clear()

    if st.button("Submit"):
        selected = tuple(st.session_state.data)

        if selected in self.output_data:
            for output in self.output_data[selected]:
                count += 1
                st.warning(f"{count}. {output['message']}")
                st.warning(f"{output['details']}")
            else:
                st.warning("Tidak ada informasi terkait data yang dipilih.")

    def run(self):
        self.selectData()
        self.display()

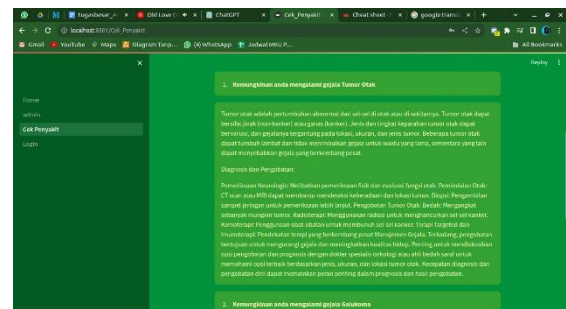
lth_checker_app = HealthCheckerApp(Data, OutputData)
lth_checker_app.run()

title("Anda Harus Login Terlebih Dahulu !!!")
```

Output:

IV. Kesimpulan

Kemajuan teknologi medis seringkali disertai dengan biaya yang sangat tinggi, yang dapat menjadi hambatan bagi sebagian masyarakat yang tidak mampu. Seperti halnya Infrastruktur Kesehatan di daerah terpencil atau berkembang, kurangnya infrastruktur kesehatan dan konektivitas dapat memperlambat aksesibilitas dan deteksi penyakit. Serta Kurangnya pemahaman dan kesadaran masyarakat terhadap pentingnya deteksi dini juga dapat menyebabkan keterlambatan dalam pencarian bantuan medis.



14. Akses Halaman Admin untuk melihat data user yang melakukan

```
import streamlit as st
import pandas as pd

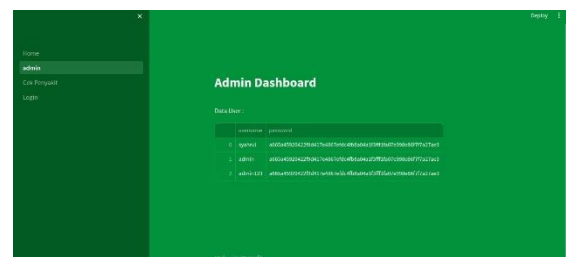
if 'login' not in st.session_state:
    st.session_state.login = False

if st.session_state.login == False:
    st.title("Silahkan Login Sebagai Admin Untuk Mengakses Halaman ini")
    st.session_state.username_login = 'gandum'
    st.session_state.password_login = '123'

if (st.session_state.username_login == 'syahrul' and st.session_state.password_login == '123'):
    st.title("Admin Dashboard")
    df = pd.read_csv("user_data.csv")
    st.write("Data User :")
    st.dataframe(df)

if st.session_state.login and st.session_state.username_login != 'syahrul' or st.session_state.password_login != '123':
    st.title("ANDA LOGIN BUKAN SEBAGAI ADMIN !!!")
```

Tampilan Halaman Admin :



Teknologi memungkinkan integrasi data dari berbagai sumber, memberikan gambaran menyeluruh tentang kondisi kesehatan pasien untuk mendukung diagnosis yang lebih akurat. Analisis big data dapat mengidentifikasi pola-pola kompleks dalam data kesehatan, memungkinkan deteksi dini dan prediksi penyakit. Selain itu teknologi mendukung telemedicine yang memungkinkan konsultasi jarak jauh dan pemantauan pasien secara real-time untuk deteksi cepat.

Modul dapat diintegrasikan dengan sistem pelayanan kesehatan yang ada,

memastikan alur kerja yang efisien dan konsistensi data. Modul dapat dirancang untuk mengakses dan memproses data secara real-time, meningkatkan kecepatan dan akurasi deteksi penyakit. Modul ini juga dapat memanfaatkan algoritma kecerdasan buatan untuk meningkatkan akurasi identifikasi gejala dan pola penyakit.

V. Referensi

- Gumilar, Adjeng Nawang. 2016. Perancangan Sistem Pakar Diagnosa Penyakit Diabetes Melitus Menggunakan Metode Forward Chaining Berbasis Web. Skripsi Program S1 Sistem Informasi Universitas Nusantara PGRI. Surabaya.
- Frajullah, Murinto. 2019. Sistem Pakar Deteksi Dini Gangguan Kecemasan (Anxiety) Menggunakan Metode Forward Chaining Berbasis Web. Jurnal Sarjana Teknik Informatika Universitas Ahmad Dahlan. Yogyakarta
- Husada, Ahmad Adi. 2016. Rancang Bangun Aplikasi Clinical Pathway Penyakit Kandungan Pada Ibu Hamil Menggunakan Algoritma Naive Bayes. <http://eprints.umm.ac.id/28092>.
- Bangun, F. (2019). SISTEM PAKAR MENDIAGNOSA PENYAKIT TBC MENGGUNAKAN METODE TEOREMA BAYES. Jurnal Teknik Dan Informatika, 6(2), 23–29.