| Universitas BSI on X: "Logo resmi Universitas Bina Sarana ... |  | |
| --- | --- | --- |
|  |  |

**DPS (Diagnosa Penyakit Sistem)**

**1Fauzan Akbar Wijaya, 2Tubagus Bayu, 3Muhammad Fajar Sodiq, 4Raditya Dwi Arkananta**

Teknik & Informatika, Informatika, Univesitas Bina Sarana Informatika, Jakarta, Indonesia

Gmail: 1fauzanakbarwijaya2303@gmail.com, 2,\*tubagusbayu91@gmail.com 3,\*fajarzman@gmail.com 4,\*raditya9arkananta@gmail.com

(\* : coressponding author)

**Abstrak**

Diagnosa Penyakit adalah sebuah aplikasi yang dirancang untuk membantu dokter dalam mendiagnosa penyakit berdasarkan gejala yang dilaporkan oleh pasien. Program ini menyediakan pemeriksaan poli umum. Dokter dapat masuk ke poli yang sesuai, memasukkan gejala yang dilaporkan pasien dan mendapatkan hasil diagnosa yang kemungkinan besar relevan. Untuk pengarsipan, sistem terintegrasi dengan penyimpanan data konsultasi secara otomatis dalam file Excel, yang mempermudah pencatatan riwayat konsultasi pasien. Dengan fokus pada efisiensi dan kemudahan akses, Diagnosa Penyakit diharapkan dapat meningkatkan kualitas layanan kesehatan melalui teknologi yang inovatif dan praktis.

**Kata Kunci:** Diagnosa Penyakit, Konsultasi, Data Penyakit, Python

**Abstract**

Disease Diagnosis is an application designed to assist doctors in diagnosing diseases based on symptoms reported by patients. The program provides general poly examinations. The doctor can log in to the appropriate clinic, enter the symptoms reported by the patient and get the most likely relevant diagnosis results. For archiving, the system is integrated with automatic storage of consultation data in Excel files, which makes it easy to record patient consultation history. With a focus on efficiency and ease of access, Diagnosa Penyakit is expected to improve the quality of healthcare through innovative and practical technology.

**Keywords**: Disease Diagnosis,Consultation, Data

1. **PENDAHULUAN**

Perkembangan teknologi digital telah memberikan dampak signifikan di berbagai sektor, termasuk bidang kesehatan. Dalam upaya meningkatkan kualitas layanan kesehatan, khususnya dalam proses diagnosa penyakit, teknologi dapat digunakan untuk mempermudah dan mempercepat proses pengambilan keputusan medis. Salah satu tantangan utama dalam dunia medis adalah memastikan diagnosa yang cepat, akurat, dan terdokumentasi dengan baik tanpa mengurangi efisiensi waktu dokter.

Untuk menjawab tantangan tersebut, dikembangkan aplikasi **DPS**, yang dirancang untuk membantu dokter dalam mendiagnosa penyakit berdasarkan gejala yang dilaporkan oleh pasien. Aplikasi ini menyediakan layanan poli, yaitu **Poli Umum**, yang masing-masing memiliki daftar gejala khusus untuk membantu dokter menentukan diagnosa yang kemungkinan besar relevan. Selain itu, sistem ini telah terintegrasi dengan fitur penyimpanan data konsultasi ke dalam file Excel untuk mempermudah pengarsipan riwayat konsultasi pasien secara otomatis.

Dengan memanfaatkan teknologi ini, diharapkan dokter dapat meningkatkan efisiensi kerja sekaligus memberikan pelayanan yang lebih baik kepada pasien. **DPS** menawarkan solusi inovatif yang praktis, mudah diakses, dan relevan untuk kebutuhan layanan kesehatan modern.

1. **METODOLOGI PENELITIAN**

**2.1 Tahapan Penilitian**

Penelitian ini menggunakan metode pengembangan sistem yang terdiri dari beberapa tahapan : Desain, Implementasi, dan Pengujian. Setiap tahap dilakukan secara berurutan untuk memastikan hasil penelitian sesuai dengan tujuan yang telah ditetapkan. Tahapan penelitian dirangkum dan dijelaskan sebagai berikut :

**2.1.1 Desain Sistem**

Pada tahap ini, Desain antarmuka pengguna(UI/UX) dilakukan dalam menggunakan **Figma**. Proses ini meliputi perancangan tampilan halaman login, diagnosa, lihat gejala, dan riwayat konsultasi. Desain difokuskan untuk memberikan pengalaman pengguna yang intuitif dan mudah diakses.

**2.1.2 Implementasi**

Aplikasi dikembangkan menggunakan Python dan framework Streamlit. Tahap implementasi mencakup :

* Pembuatan fitur login dan diagnosa gejala
* Integrasi sistem pencatatan data ke file Excel
* Implementasi antarmuka interaktif sesuai desain pada tahap sebelumnya

**2.1.3 Pengujian(Testing)**

Pengujian dilakukan untuk memastikan sistem berjalan sesuai harapan. Jenis pengujian yang akan meliputi :

* Functional Testing : Memastikan fungsi-fungsi utama bekerja sesuai dengan kebutuhan.
* Usability Testing : Menguji kenyamanan pengguna saat menggunakan aplikasi.
* Error Testing : Mengidentifikasi dan memperbaiki bug pada sistem

1. **HASIL DAN PEMBAHASAN**

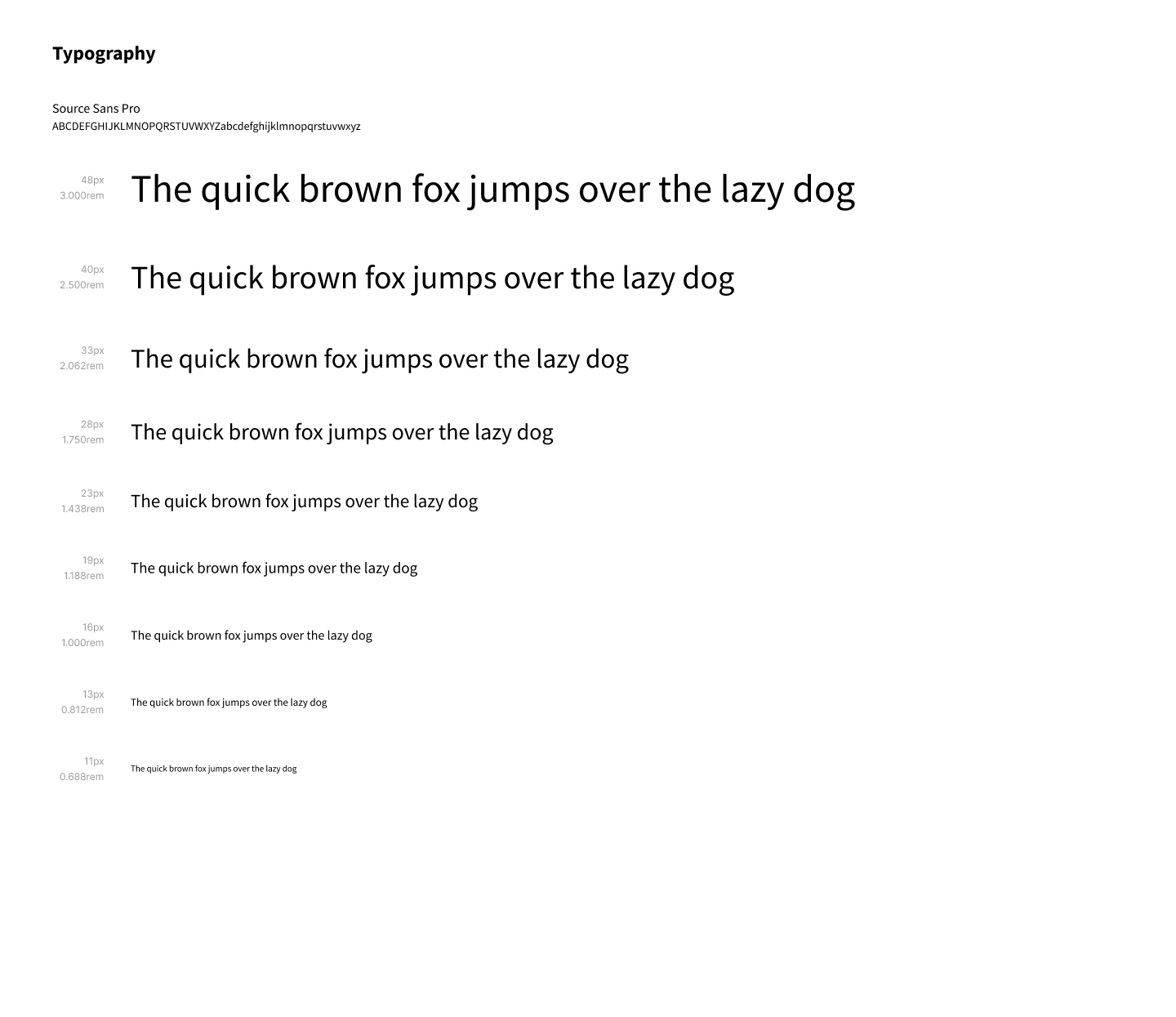
**3.1 Hasil dan pembahasan tahapan**

Setelah melalui tahapan desain, implementasi, dan pengujian, aplikasi DPS berhasil dikembangkan sesuai tujuan. Berikut hasil-hasil yang diperoleh :

**3.1.1 Desain Sistem**

Desain antarmuka dibuat untuk memberikan pengalaman pengguna yang intuitif. Desain ini dirancang dengan memperhatikan prinsip kemudahan akses, efisiensi penggunaan, dan estetika. Berikut adalah hasil dan pembahasan dari desain sistem :

1. **Typography**

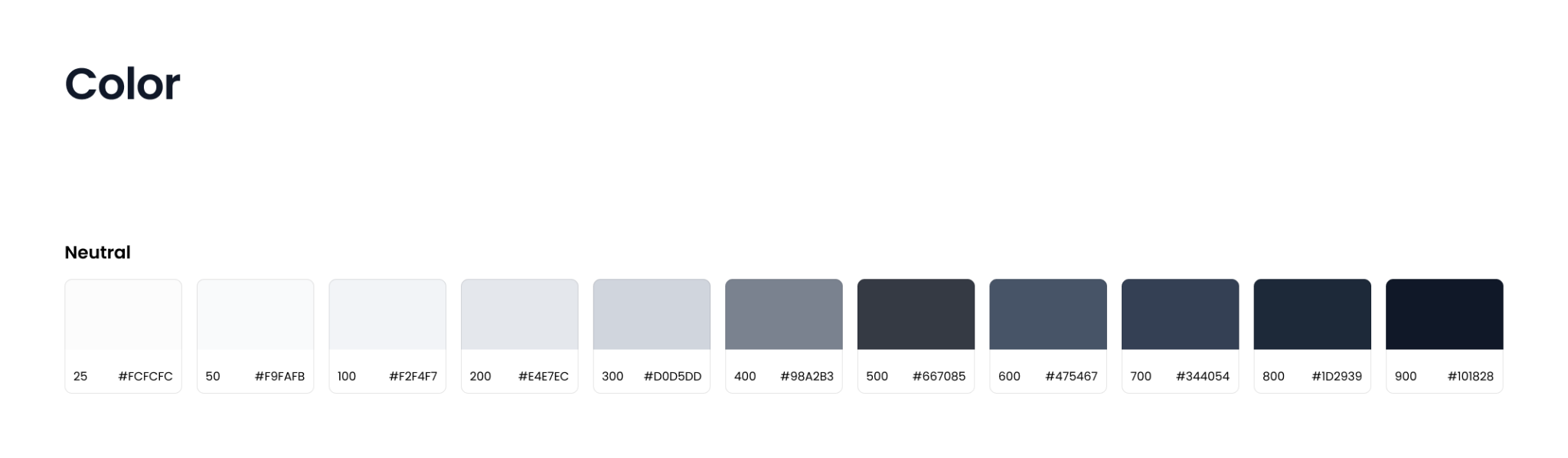
****

Dalam pemilihan tipografi, kami menggunakan font Source Sans Pro. Font ini dirancang secara khusus untuk memberikan keseimbangan antara estetika dan keterbacaan, sehingga sangat cocok untuk aplikasi bertema kesehatan.

Keunggulan Source Sans Pro terletak pada karakter hurufnya yang minimalis namun profesional, memastikan teks dapat dibaca dengan mudah oleh pengguna dari berbagai usia. Selain itu, font ini mampu menyesuaikan diri dalam berbagai ukuran layar, mulai dari perangkat kecil seperti ponsel hingga monitor desktop, sehingga mendukung pengalaman pengguna yang konsisten.

Kami juga mempertimbangkan kombinasi ukuran font untuk meningkatkan hierarki informasi, seperti penggunaan ukuran font lebih besar untuk judul halaman dan ukuran lebih kecil untuk deskripsi atau informasi tambahan.

1. **Color system**

****

Pemilihan skema warna dalam aplikasi ini bertujuan untuk menciptakan suasana yang menenangkan dan profesional. Kami menggunakan dominasi warna netral seperti abu-abu muda (#F5F5F5) dan putih (#FFFFFF) sebagai latar belakang untuk menjaga kesan bersih dan sederhana. Untuk memberikan kontras, elemen penting seperti tombol aksi diberi warna orangelembut (#C36B0D).

Pemilihan warna biru didasarkan pada asosiasinya dengan kepercayaan, ketenangan, dan profesionalisme, yang sangat penting dalam konteks aplikasi kesehatan. Kami juga menerapkan hover effect pada tombol, di mana perubahan warna yang halus terjadi ketika pengguna mengarahkan kursor, memberikan umpan balik visual yang intuitif.

Keseluruhan skema warna telah diuji dengan alat contrast checker untuk memastikan tingkat keterbacaan yang optimal, terutama bagi pengguna dengan gangguan penglihatan seperti buta warna.

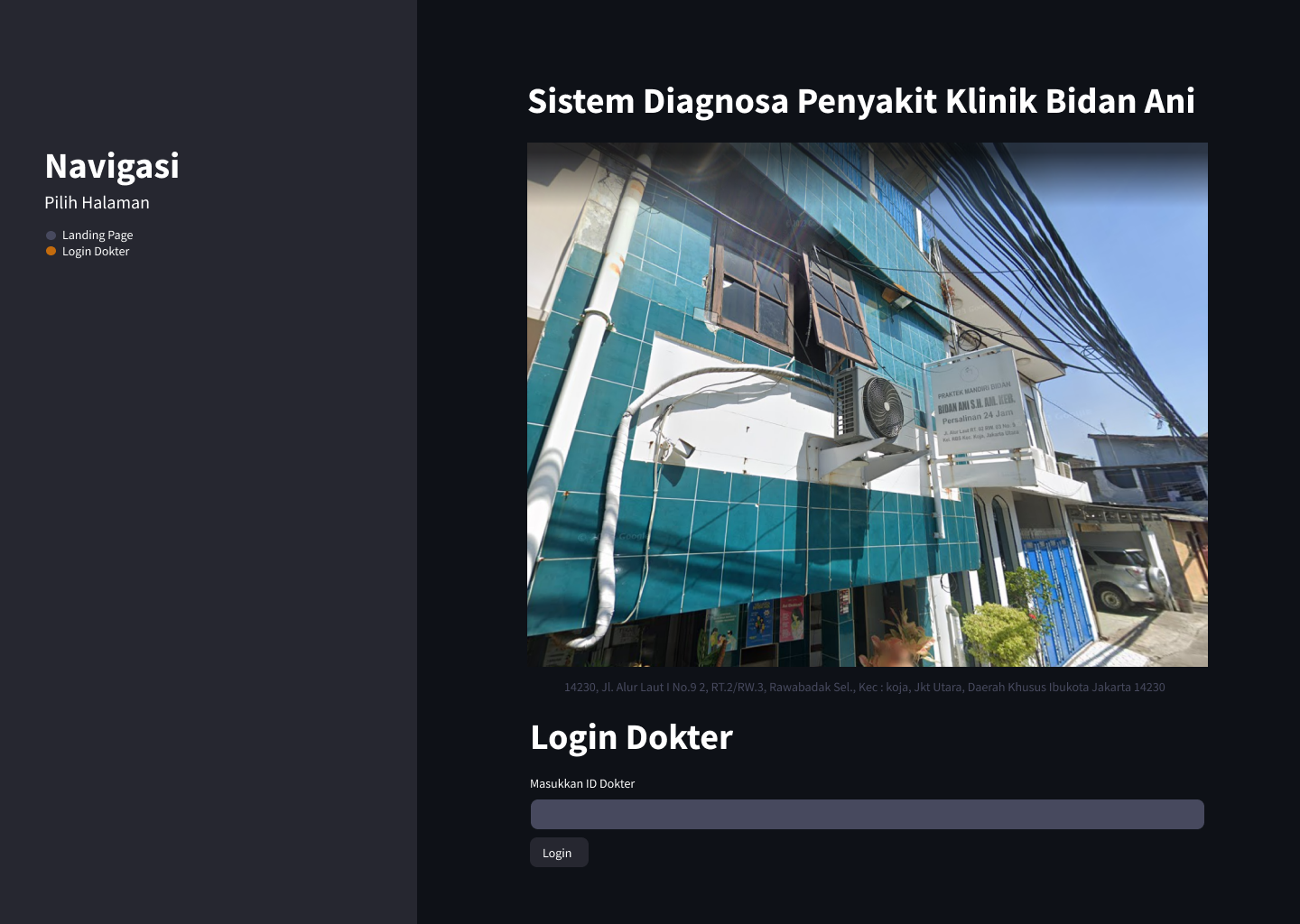
1. **Landing Page UI**

****

Landing Page dirancang dengan prinsip UI yang intuitif dan langsung menyampaikan tujuan aplikasi. Struktur halaman menggunakan elemen hero section yang menampilkan deskripsi singkat aplikasi bersama tombol aksi utama (Call-to-Action) seperti "Masuk" dan "Daftar".

Di bagian bawahnya terdapat informasi tambahan tentang fitur utama aplikasi untuk memberikan gambaran menyeluruh kepada pengguna baru. Kami juga menambahkan ikon yang relevan untuk memudahkan pemahaman visual pengguna.

1. **Login Page UI**

****

Pada halaman login, kami berfokus pada kemudahan pengguna dalam melakukan autentikasi. Formulir login hanya mencakup elemen-elemen esensial seperti kolom id dokter untuk validasi Username dan password. Kedepannya akan dilengkapi dengan tombol "Lupa Kata Sandi" untuk mempermudah pemulihan akun.

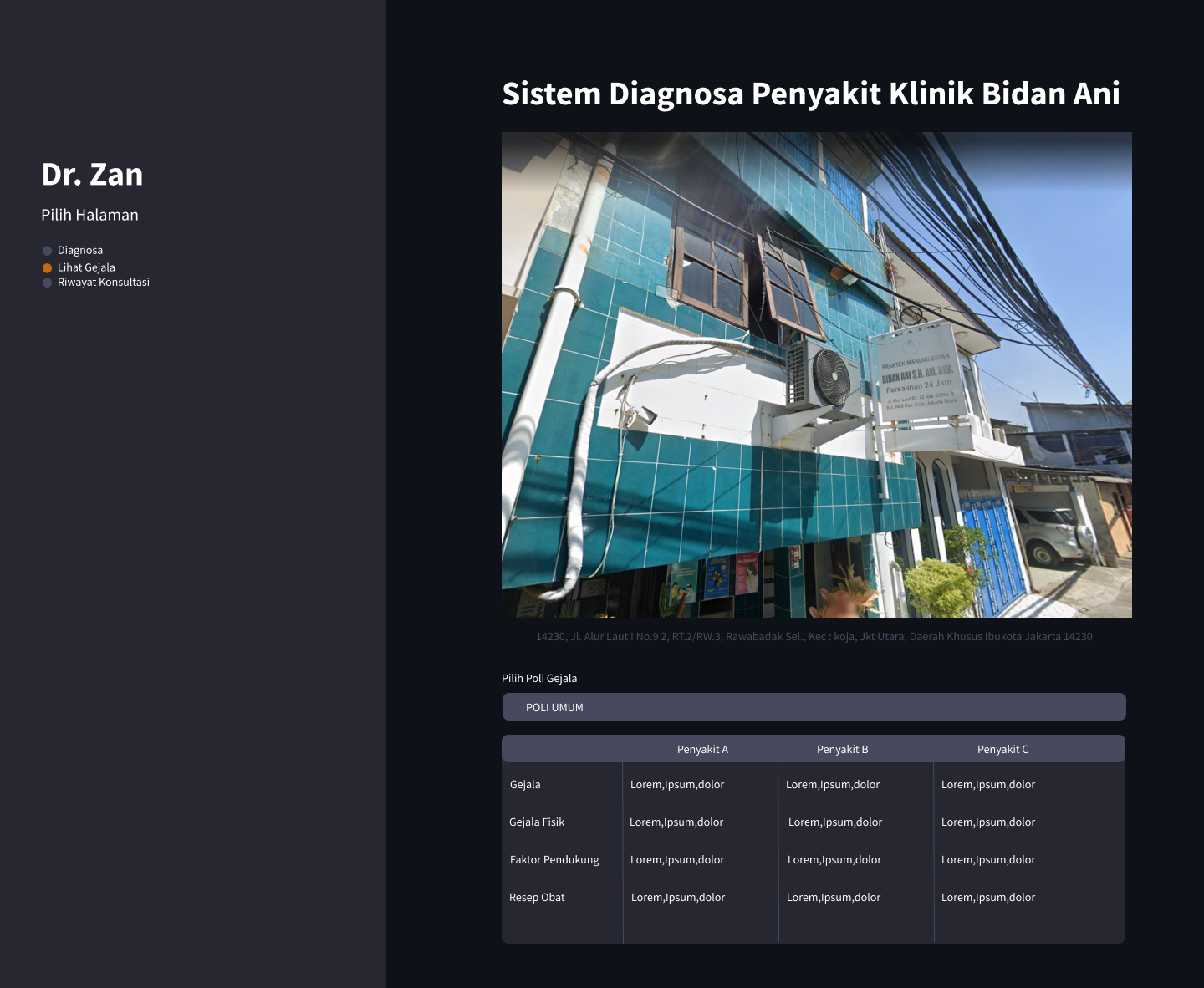
Tombol aksi di halaman ini diberi warna biru cerah untuk menarik perhatian, sementara elemen pendukung seperti deskripsi kecil di bawah formulir menggunakan warna abu-abu untuk mengurangi distraksi.

1. **Diagnosa Page UI**

****

Halaman ini dirancang untuk memberikan alur kerja yang efisien kepada pengguna. Kami menggunakan layout tabular untuk menampilkan daftar gejala yang dapat dipilih dokter. Setiap baris pada tabel dirancang dengan warna alternatif untuk memudahkan pembacaan. Selain itu, hasil diagnosa ditampilkan dalam panel terpisah dengan format yang rapi, sehingga meminimalkan kemungkinan kesalahan interpretasi.

1. **Gejala Page UI**

****

Gejala Page menggunakan struktur card-based layout untuk menampilkan daftar gejala. Setiap kartu mencantumkan nama gejala beserta deskripsi singkat. Kami juga menambahkan fitur pencarian dan filter di bagian atas halaman untuk membantu pengguna menemukan gejala tertentu dengan cepat. Warna latar belakang kartu dibuat kontras dengan teks agar informasi tetap jelas terlihat.

1. **Riwayat Konsultasi Page UI**

****

Halaman ini menyajikan daftar riwayat konsultasi dalam bentuk tabel yang interaktif. Kolom dalam tabel mencakup informasi seperti tanggal konsultasi, nama pasien, dan hasil diagnosa.

Untuk memperkaya pengalaman pengguna, kami menambahkan fitur sort dan filter pada tabel, memungkinkan pengguna menyusun data berdasarkan parameter tertentu. Setiap baris dalam tabel juga dapat diklik untuk membuka rincian konsultasi, memberikan fleksibilitas navigasi.

**3.1.2 Implementasi**

Tahap implementasi berfokus pada pengembangan aplikasi berdasarkan desain yang telah dibuat sebelumnya. Proses ini mencakup penerapan fitur-fitur utama seperti autentikasi pengguna, sistem diagnosa berbasis gejala, dan pencatatan data konsultasi ke dalam file Excel secara otomatis. Framework Streamlit digunakan sebagai dasar pengembangan antarmuka interaktif, memungkinkan integrasi mulus antara elemen UI dan logika aplikasi. Implementasi dilakukan secara bertahap untuk memastikan setiap fitur berjalan sesuai kebutuhan dan dapat diakses dengan mudah oleh pengguna. Berikut adalah hasil dan pembahasan dari Implementasi :

1. **Teknologi**

Dalam pengembangan aplikasi ini, kami menggunakan berbagai teknologi untuk memastikan efisiensi dan kemudahan proses pengembangan. Teknologi utama yang digunakan meliputi:

1. **Python version 3.12.4**

Python dipilih karena fleksibilitasnya dan dukungannya yang luas dalam pengembangan aplikasi berbasis data. Versi 3.12.4 digunakan untuk memastikan stabilitas dan kompatibilitas dengan library terbaru.

1. **Framework Streamlit**

Streamlit digunakan untuk membangun antarmuka pengguna (UI) yang responsif dan interaktif. Framework ini memungkinkan pengembang membuat aplikasi berbasis web dengan cepat tanpa perlu menulis banyak kode frontend. Streamlit juga mendukung integrasi langsung dengan berbagai library Python, sehingga mempermudah proses pengembangan.

1. **Library Pandas**

Pandas digunakan untuk mengolah dan menganalisis data gejala serta hasil diagnosa. Library ini mempermudah proses manipulasi data, seperti membaca, menulis, dan mengatur format data yang akan disimpan ke file Excel.

1. **Library OpenPyXL**

OpenPyXL digunakan untuk mengelola file Excel, termasuk mencatat data hasil konsultasi pasien. Dengan library ini, sistem dapat menambahkan data baru secara otomatis tanpa mengubah format atau struktur file Excel yang sudah ada.

1. **Library OS**

Library ini digunakan untuk menangani operasi sistem, seperti membaca atau menulis file, membuat direktori, dan memastikan bahwa file Excel dapat diakses atau diperbarui sesuai kebutuhan aplikasi.

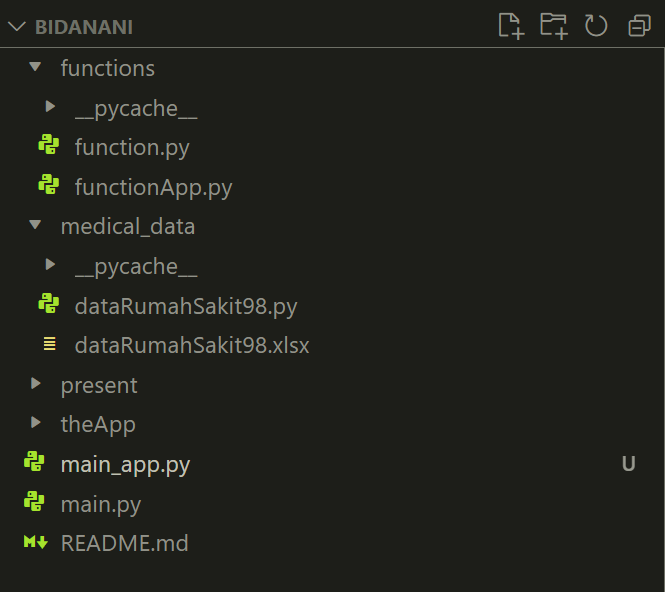
1. **Library Datetime**

Datetime digunakan untuk menangani data waktu, seperti mencatat tanggal dan waktu konsultasi pasien. Data waktu ini sangat penting untuk memastikan bahwa riwayat konsultasi dapat diorganisir dengan baik dan mudah dilacak.

1. **Arsitektur Sistem**

Tahap implementasi juga mencakup pengorganisasian struktur folder dan kode program untuk memastikan aplikasi mudah dipelihara, dikembangkan, dan diakses. Berikut adalah penjelasan masing-masing komponen yang dirancang:

1. **Struktur Folder**



Struktur folder dalam aplikasi ini dirancang dengan pendekatan modular untuk memisahkan fungsi utama, data, dan antarmuka aplikasi. Folder functions berisi fungsi-fungsi utama untuk mendukung logika aplikasi, seperti autentikasi dan pengolahan data.

Folder medical\_data menyimpan data rumah sakit berupa file Python untuk pemrosesan serta file Excel sebagai penyimpanan data riwayat konsultasi.

File utama, seperti main\_app.py dan main.py, bertindak sebagai penghubung antara antarmuka pengguna dan backend aplikasi. Dengan struktur ini, proses pengembangan dan pemeliharaan aplikasi menjadi lebih terorganisir.

1. **Main Code**

Tahap implementasi mencakup dua file utama, yaitu main.py dan main\_app.py. Berikut adalah penjelasan terkait fungsi dan isi dari masing-masing file:

1. **main.py**

File ini berfungsi sebagai kode inti yang dapat dijalankan melalui terminal. Berikut fitur-fitur utama yang terdapat pada file ini:

* **Autentikasi Dokter**:

# Autentikasi Dokter

nama\_dokter, poli\_dokter = auth\_dokter(df\_dokter)

Dokter harus melalui proses login berbasis nama dan spesialisasi menggunakan data dari dataRumahSakit98.

* **Proses Diagnosa**:

# Proses diagnosa Poli Umum

penyakit, persentase, saran\_obat = proses\_diagnosa(gejala, gejala\_fisik, faktor\_pendukung, gejalaUmum)

Berdasarkan gejala yang diinput oleh dokter, kode ini memproses kemungkinan penyakit, tingkat keparahan, dan saran obat.

* **Pencatatan Riwayat Konsultasi**:

# Simpan history ke Excel setelah dokter selesai

simpan\_history(history\_konsultasi)

Semua data pasien disimpan secara otomatis ke file Excel menggunakan fungsi simpan\_history\_excel.

1. **main\_app.py**

File ini merupakan versi antarmuka berbasis Streamlit dari main.py. Fungsi utama dari file ini meliputi:

Login Page :

result = auth\_dokter(df\_dokter, id\_dokter)

Dokter harus login melalui halaman interaktif dengan nama dan poli.

Navigasi Halaman :

page = st.sidebar.radio(

"Pilih Halaman",

["Diagnosa", "Lihat Gejala", "Riwayat Konsultasi"]

)

Antarmuka aplikasi terdiri dari beberapa halaman seperti Diagnosa, Lihat Gejala, dan Riwayat Konsultasi.

1. **function.py**

File ini berisi tiga fungsi utama yang mendukung proses autentikasi dokter, diagnosa pasien, dan penyimpanan riwayat konsultasi.

**Fungsi auth\_dokter()**

Fungsi ini digunakan untuk melakukan autentikasi dokter berdasarkan ID. Fungsi akan:

* Meminta input ID dokter.
* Mengecek ID tersebut dalam DataFrame berisi data dokter.
* Jika ID cocok, fungsi akan mengembalikan nama dokter dan poli yang diampunya. Jika tidak, akan muncul pesan error untuk memasukkan ID yang valid.

**Fungsi proses\_diagnosa()**

Fungsi ini digunakan untuk menganalisis gejala yang dimasukkan oleh pengguna dan mencocokkannya dengan dataset gejala penyakit yang ada. Prosesnya adalah sebagai berikut:

* Menghitung jumlah gejala pasien yang cocok dengan data gejala penyakit (termasuk gejala umum, fisik, dan faktor pendukung).
* Menghitung persentase kecocokan dengan membandingkan jumlah gejala yang cocok dengan total gejala terkait penyakit.
* Menentukan penyakit yang paling mungkin diderita pasien berdasarkan persentase kecocokan tertinggi dan memberikan rekomendasi obat.

**Fungsi simpan\_history\_excel()**

Fungsi ini digunakan untuk menyimpan riwayat konsultasi ke dalam file Excel. Jika file sudah ada, data baru akan digabungkan dengan data sebelumnya. Jika file belum ada, akan dibuat file Excel baru yang berisi riwayat konsultasi.

1. **datarumahsakit.py**

File ini berfungsi sebagai pengelola data statis yang dibutuhkan dalam sistem, termasuk data dokter dan dataset gejala. Berikut komponen utamanya:

* **Data Dokter**

Data dokter diambil dari file Excel (sheet "Dokter") dan diubah menjadi list of dictionaries yang disimpan dalam variabel dataDokter.

* **Data Gejala Umum**

Data gejala penyakit umum dibaca dari file Excel (sheet "GejalaUmum"). Setiap penyakit memiliki atribut gejala, gejala fisik, faktor pendukung, dan resep obat. Data ini diolah menjadi struktur dictionary yang memudahkan proses diagnosa.

**3.1.3 Debugging**

Tahap debugging dilakukan untuk memastikan bahwa sistem berjalan sesuai dengan spesifikasi yang telah ditentukan serta bebas dari error atau bug. Proses ini mencakup beberapa langkah utama:

1. **Identifikasi Masalah**

Debugging diawali dengan pengamatan terhadap error atau perilaku program yang tidak sesuai saat kode dieksekusi. Hal ini bisa dilihat melalui output di terminal untuk main.py atau pada log dan feedback di antarmuka Streamlit untuk main\_app.py.

1. **Penggunaan Tools Debugging**

* **Python Debugger :**

digunakan untuk menelusuri eksekusi program pada main.py melalui terminal. Dengan pdb, developer dapat mengecek nilai variabel secara real-time dan menemukan titik error.

* **Streamlit Debugging :**

untuk main\_app.py dengan fitur bawaan seperti st.write() atau st.error() yang memudahkan pengecekan variabel dan hasil proses.

1. **Perbaikan Code**

Setelah menemukan penyebab masalah, kode diperbaiki dan dijalankan kembali untuk memastikan tidak ada error tambahan yang muncul.

1. **Pengujian Ulang**

Program diuji ulang menggunakan beberapa skenario input untuk memastikan perbaikan berjalan optimal dan tidak menimbulkan bug baru.

1. **KESIMPULAN**

Dalam jurnal ini, telah dikembangkan sistem konsultasi kesehatan berbasis Python yang berfokus pada diagnosis penyakit berdasarkan gejala pasien. Sistem ini dirancang untuk membantu dokter dalam mengidentifikasi kemungkinan penyakit dengan menggunakan data yang terstruktur dalam file Excel. Beberapa kesimpulan yang dapat diambil dari implementasi ini antara lain:

1. **Fungsi Autentikasi dokter**

Sistem memastikan bahwa hanya dokter dengan ID yang valid yang dapat mengakses dan menggunakan fitur diagnostik. Proses autentikasi ini meningkatkan keamanan dan akurasi dalam penggunaan sistem.

1. **Proses Diagnosa Sistem Berbasis Gejala**

Fungsi proses\_diagnosa() berhasil mengolah gejala yang dimasukkan pasien dan membandingkannya dengan dataset gejala yang tersedia. Hasil diagnosa ditentukan berdasarkan persentase kecocokan gejala, yang memudahkan identifikasi penyakit secara sistematis.

1. **Manajemen Data Terpusat**

Penggunaan file Excel sebagai sumber data memungkinkan integrasi yang fleksibel dan mudah diperbarui. Data dokter, gejala penyakit umum, dan gejala gigi tersimpan dalam struktur yang terorganisir dan dapat diakses dengan cepat oleh sistem.

1. **Penyimpanan Riwayat Konsultasi**

Fungsi simpan\_history\_excel() memungkinkan riwayat konsultasi tersimpan secara otomatis dalam format Excel. Hal ini mempermudah pemantauan dan pelacakan histori diagnosa pasien untuk keperluan analisis lebih lanjut.

1. **REFERENCES**

(Times New Roman, 9)

Semua pengutipan referensi yang dikutip di artikel ini WAJIB TERDAPAT PADA ISI ARTIKEL dan WAJIB untuk menggunakan **alat referensi seperti MENDELEY dengan format IEEE**, 80% literatur **PRIMER** (jurnal, prosiding, laporan penelitian, paten, standar, dokumen sejarah, buku hasil riset) dan 20% literatur **SEKUNDER** (buku, website yang dapat dipercaya) dalam kurun waktu sekurang-kurangnya 5 tahun terakhir

Jumlah referensi yang digunakan **minimum sebanyak 10 referensi**.

**(Hapus semua informasi berwarna merah tambahan seperti ini di naskah pengiriman Anda)**