

Buku Panduan Pengguna Health Checker Rule Based

Disusun Guna Memenuhi Tugas Akhir Semester Mata Kuliah Sistem Cerdas

Dosen Pengampu : NUR AENI WIDIASTUTI, S.Pd., M.Kom.



Disusun Oleh :

Nama : Muhammad Ashab Ibnu Abdul Aziz (231240001399)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NAHDLATUL ULAMA JEPARA
TAHUN AKADEMIK 2025/2026**

DAFTAR ISI

DAFTAR ISI.....	2
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang	4
1.2 Tujuan	4
1.3 Manfaat	4
BAB II GAMBARAN UMUM SISTEM	5
2.1 Deskripsi Sistem	5
2.2 Fitur Unggulan	5
2.3 Persyaratan Sistem	5
BAB III INSTALASI DAN KONFIGURASI.....	7
3.1 Prosedur Instalasi	7
Langkah 1: Persiapan Direktori	7
Langkah 2: Akses Terminal	7
Langkah 3: Isolasi Lingkungan Python (Virtual Environment)	7
Langkah 4: Instalasi Dependensi	7
3.2 Konfigurasi Sistem.....	7
1. Perolehan API Key	8
2. Konfigurasi Variabel Lingkungan	8
3.3 Menjalankan Aplikasi	8
BAB IV PETUNJUK OPERASIONAL	11
4.1 Modul Diagnosis (Beranda).....	11
4.2 Kalkulator BMI.....	13
4.3 Komparasi Tinggi Badan	14
4.4 Asisten Kesehatan Virtual (AI Chatbot)	15
4.5 Panel Administrasi	17
BAB V SPESIFIKASI TEKNIS	20
5.1 Arsitektur Backend	20
5.2 Arsitektur Frontend.....	20
5.3 Basis Pengetahuan (Knowledge Base).....	20

5.4 Logika Sistem Pakar (Inference Engine)	21
BAB VI PENUTUP	22
6.1 Batasan Sistem	22
6.2 Pemecahan Masalah (Troubleshooting).....	22

BAB I PENDAHULUAN

1.1 Latar Belakang

Kesehatan merupakan aspek fundamental dalam kehidupan manusia yang memerlukan perhatian serius. Di era digital saat ini, teknologi informasi memegang peranan krusial dalam meningkatkan aksesibilitas layanan kesehatan. Seringkali, masyarakat mengalami kesulitan dalam mengidentifikasi gejala awal penyakit karena keterbatasan pengetahuan medis atau akses ke fasilitas kesehatan yang memadai. Oleh karena itu, diperlukan sebuah sistem cerdas yang mampu memberikan diagnosis awal secara cepat dan akurat sebagai langkah preventif.

1.2 Tujuan

Pengembangan aplikasi Health Checker Rule Based ini bertujuan untuk:

1. Menyediakan alat bantu diagnosis mandiri (*self-diagnosis*) yang dapat diakses dengan mudah oleh masyarakat.
2. Mengimplementasikan sistem pakar berbasis aturan (*rule-based*) untuk mendeteksi potensi penyakit berdasarkan gejala klinis.
3. Mengintegrasikan teknologi kecerdasan buatan (AI) guna memberikan pengalaman konsultasi kesehatan yang interaktif.

1.3 Manfaat

Aplikasi ini diharapkan dapat memberikan manfaat sebagai berikut:

- Bagi Pengguna: Memberikan informasi kesehatan awal yang akurat dan rekomendasi tindakan yang tepat sebelum berkonsultasi dengan dokter.
- Bagi Tenaga Medis: Membantu dalam proses *screening* awal pasien melalui data riwayat gejala yang terstruktur.
- Bagi Akademisi: Menjadi referensi implementasi sistem pakar hibrida yang menggabungkan metode konvensional dan AI modern.

BAB II GAMBARAN UMUM SISTEM

2.1 Deskripsi Sistem

Health Checker Rule Based merupakan sistem pakar berbasis web yang dikembangkan untuk memfasilitasi diagnosis awal kesehatan secara mandiri. Sistem ini mengintegrasikan dua pendekatan utama:

1. Sistem Berbasis Aturan (Rule-Based System): Mengadopsi logika *Forward Chaining* dengan aturan medis baku (IF-THEN rules) untuk menghasilkan diagnosis yang deterministik dan konsisten.
2. Kecerdasan Buatan (AI Chatbot): Memanfaatkan teknologi *Large Language Model* (Google Gemini) untuk menyediakan layanan konsultasi kesehatan yang interaktif dan responsif terhadap bahasa alami.

2.2 Fitur Unggulan

- Sistem Diagnosis Gejala: Mampu mengidentifikasi potensi penyakit berdasarkan basis data yang mencakup lebih dari 90 gejala klinis.
- Kalkulator BMI & BMR: Modul perhitungan Indeks Massa Tubuh (IMT) dan Laju Metabolisme Basal (BMR) untuk pemantauan status gizi.
- Komparasi Tinggi Badan: Visualisasi interaktif untuk membandingkan tinggi badan pengguna dengan referensi lain.
- Asisten Kesehatan Virtual: Chatbot cerdas yang siap menjawab pertanyaan seputar kesehatan umum.
- Panel Administrasi: Antarmuka untuk meninjau basis pengetahuan (*Knowledge Base*) yang digunakan oleh sistem.

2.3 Persyaratan Sistem

Untuk memastikan aplikasi berjalan dengan optimal, perangkat keras dan perangkat lunak yang digunakan harus memenuhi spesifikasi berikut:

- Sistem Operasi: Mendukung Windows, macOS, dan distribusi Linux.
- Bahasa Pemrograman: Python versi 3.8 atau yang lebih baru.

- Peramban Web (Browser): Google Chrome, Mozilla Firefox, atau Microsoft Edge versi terbaru.
- Koneksi Jaringan: Diperlukan akses internet stabil untuk fitur AI Chatbot.

BAB III INSTALASI DAN KONFIGURASI

3.1 Prosedur Instalasi

Berikut adalah langkah-langkah teknis untuk melakukan instalasi aplikasi pada lingkungan lokal (*Local Environment*).

Langkah 1: Persiapan Direktori

Anda dapat mengunduh kode sumber proyek melalui repositori GitHub resmi atau menggunakan file arsip yang disediakan.

Opsi A: Clone dari GitHub

```
git clone https://github.com/Abibsa/Health-Checker.git
```

Opsi B: Ekstrak File ZIP Jika Anda memiliki file proyek dalam format terkompresi (.zip), silakan ekstrak ke lokasi yang diinginkan.

Langkah 2: Akses Terminal

Buka terminal atau *Command Prompt* (CMD/PowerShell) dan arahkan direktori kerja ke folder proyek:

```
cd "path/to/health-checker-rule-based"
```

Langkah 3: Isolasi Lingkungan Python (Virtual Environment)

Disarankan untuk membuat lingkungan virtual agar dependensi proyek tidak berkonflik dengan sistem global.

Untuk Pengguna Windows:

```
python -m venv venv  
venv\Scripts\activate
```

Untuk Pengguna macOS/Linux:

```
python3 -m venv venv  
source venv/bin/activate
```

Langkah 4: Instalasi Dependensi

Unduh dan pasang pustaka (*library*) yang diperlukan menggunakan manajer paket pip:

```
pip install -r requirements.txt
```

Mohon tunggu hingga proses instalasi selesai sepenuhnya.

3.2 Konfigurasi Sistem

Sistem memerlukan kunci akses (API Key) dari Google Gemini untuk mengaktifkan fitur Chatbot.

1. Perolehan API Key

1. Akses laman [Google AI Studio](#).
2. Masuk menggunakan akun Google Anda.
3. Pilih opsi **Create API Key**.
4. Salin kunci API yang dihasilkan.

2. Konfigurasi Variabel Lingkungan

1. Temukan file `.env.example` pada direktori utama proyek.
2. Ubah nama file tersebut menjadi `.env`.
3. Sunting file `.env` menggunakan editor teks.
4. Masukkan API Key Anda pada variabel `GEMINI_API_KEY`.

Contoh konfigurasi:

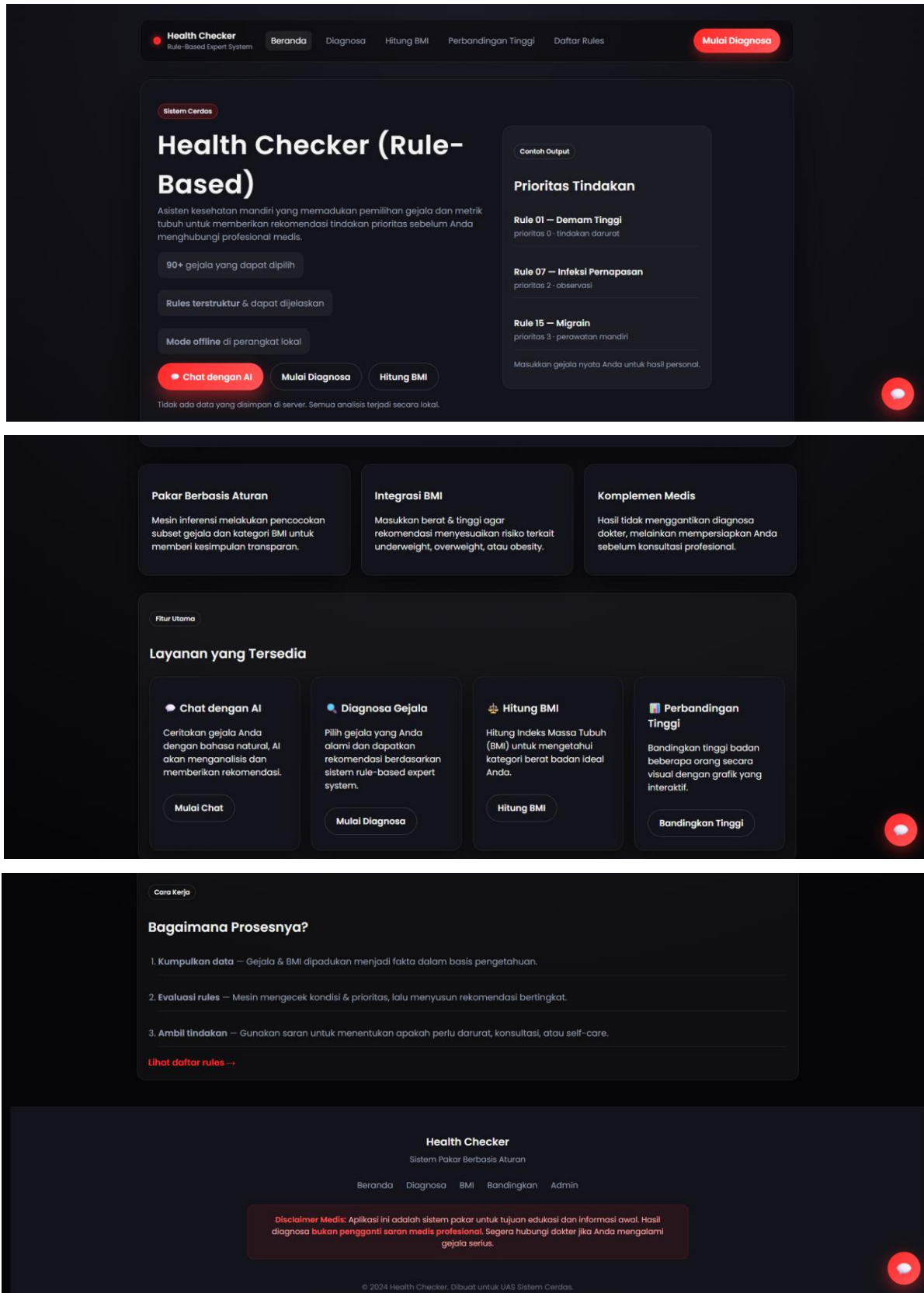
`GEMINI_API_KEY=AIzaSyB1yplt-hF3ucUZd5vN3xhupY40VF9udhs`

Catatan: Pengguna Windows dapat menggunakan skrip otomatis `scripts\setup_api_key.bat` untuk mempermudah proses ini.

3.3 Menjalankan Aplikasi

Setelah konfigurasi selesai, aplikasi siap untuk dijalankan.

1. Pastikan terminal masih aktif di direktori proyek.
2. Eksekusi perintah berikut untuk memulai server Flask:
`python app.py`
3. Jika berhasil, terminal akan menampilkan status server berjalan, misalnya:
* Running on `http://0.0.0.0:5000/` (Press CTRL+C to quit)
4. Buka peramban web dan akses alamat: **`http://localhost:5000`**
Tampilan Halaman Utama >



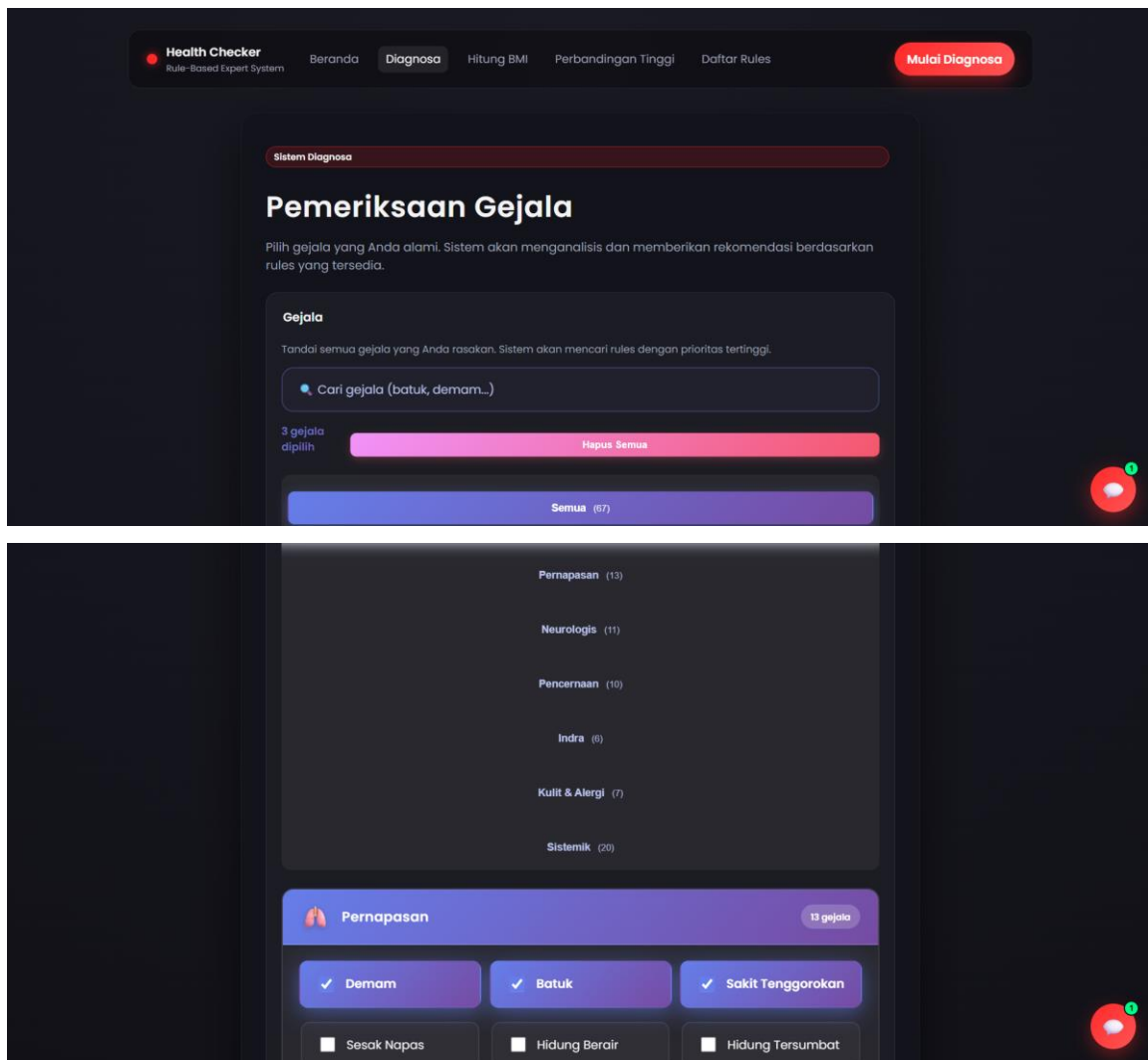
Gambar 1: Antarmuka Halaman Utama Aplikasi

BAB IV PETUNJUK OPERASIONAL

4.1 Modul Diagnosis (Beranda)

- Input Gejala: Pilih gejala yang dialami dengan mencentang kotak yang tersedia. Gunakan fitur pencarian untuk efisiensi.
- Filter Kategori: Gejala dikelompokkan berdasarkan sistem organ (misal: Pernapasan, Pencernaan) untuk memudahkan navigasi.
- Data Fisik: Masukkan data tinggi dan berat badan (opsional) untuk meningkatkan akurasi analisis.

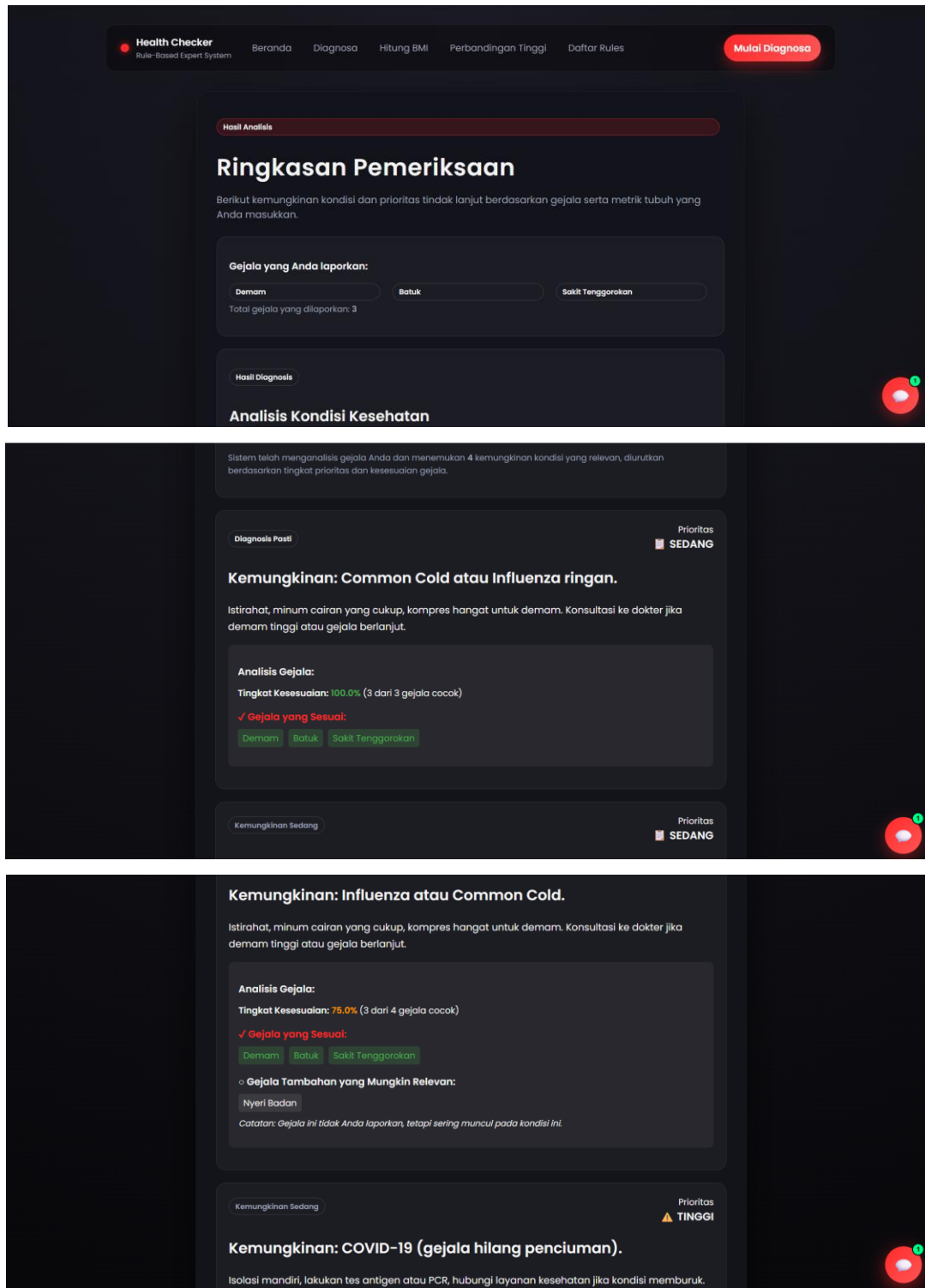
Halaman Input Diagnosa >



Gambar 2: Antarmuka Input Gejala dan Kategori

- Proses Analisis: Tekan tombol “Periksa Sekarang”. Sistem akan memproses data dan menampilkan hasil diagnosis beserta tingkat probabilitasnya.

Halaman Hasil Diagnosa >



Gambar 3: Hasil Diagnosis dengan Tingkat Prioritas

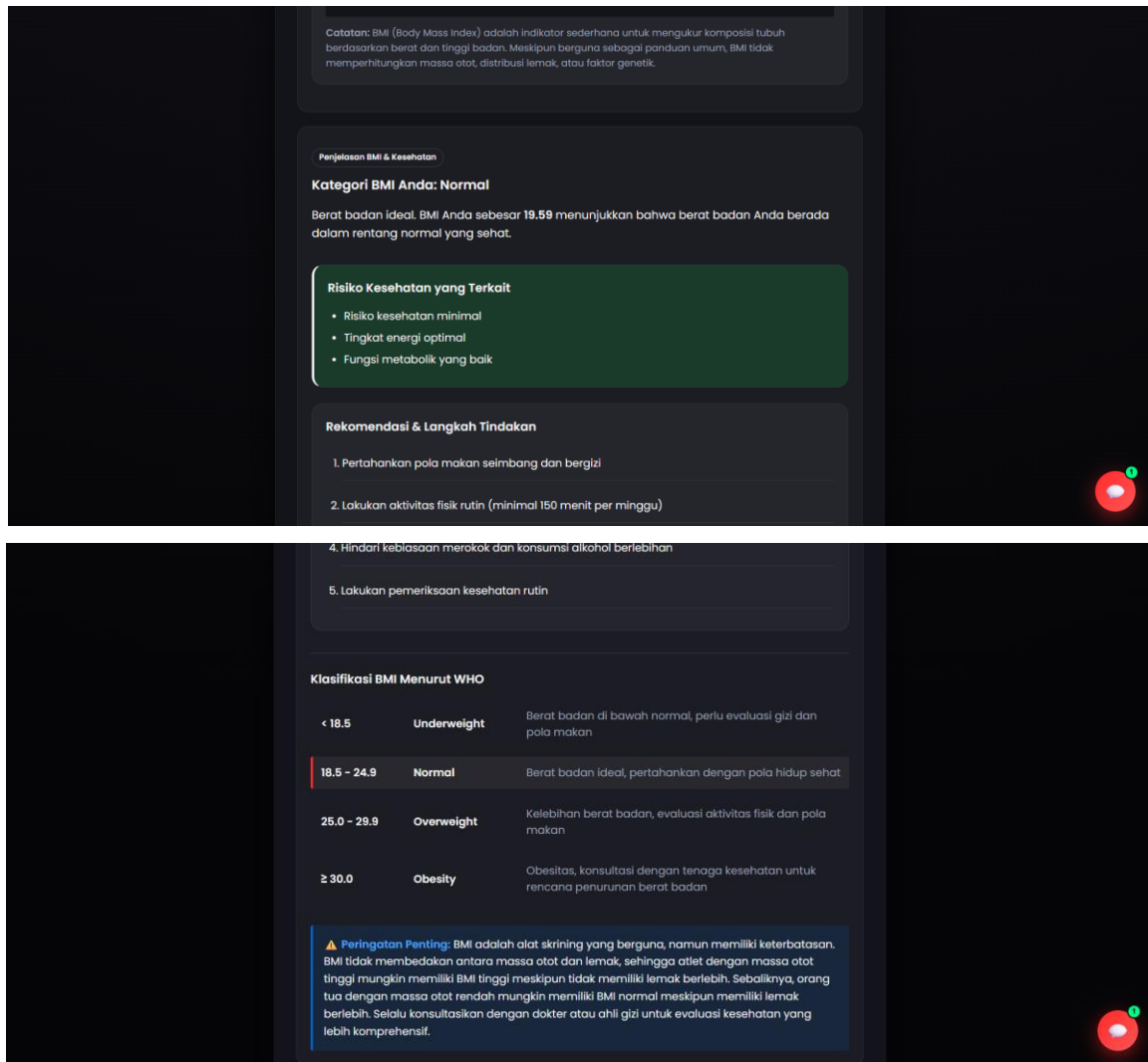
4.2 Kalkulator BMI

- Modul ini berfungsi untuk mengevaluasi proporsi berat badan terhadap tinggi badan.
- Masukkan parameter Berat (kg) dan Tinggi (cm).
- Tekan tombol Hitung.
- Sistem akan menyajikan kategori BMI (Kurus, Normal, Gemuk, Obesitas) serta estimasi kebutuhan kalori harian.

Kalkulator BMI >

The screenshot shows the 'Hitung BMI' (Calculate BMI) section of a web application. The header includes 'Health Checker Rule-based Expert System' and navigation links: 'Beranda', 'Diagnosa', 'Hitung BMI', 'Perbandingan Tinggi', and 'Daftar Rules'. A red 'Mulai Diagnosa' button is in the top right. The main heading is 'Kalkulator BMI' and 'Hitung Berat Badan Ideal'. Below this, instructions state: 'Masukkan berat dan tinggi badan untuk menghitung Indeks Massa Tubuh (BMI) dan mengetahui kategori berat badan Anda.' The 'Profil Anda' section contains input fields for 'Nama (opsional)' (Muhammad Ashab Ibnu Abdul Aziz), 'Berat' (60), and 'Tinggi' (175). There are radio buttons for 'Metric (kg / cm)' (selected) and 'Imperial (lbs / inches)'. A large red 'Hitung BMI' button is at the bottom. A 'Kembali ke Beranda' link is in the bottom left corner.

The screenshot shows the results of the BMI calculation. The header is 'Hasil Perhitungan' and the user's name 'Muhammad Ashab Ibnu Abdul Aziz' is displayed. The BMI result is 'BMI: 19.59' with a green 'Normal' label. Below this, it says 'Berat badan ideal'. A table shows the input data: 'Berat Badan 60.0 kg (60.0 kg)', 'Tinggi Badan 175.0 cm (175.0 cm)', and 'BMR (estimasi) 1549.0 kkal (175.0 cm)'. Below the table, the 'Rentang Berat Badan Ideal' section shows 'Minimum (BMI 18.5): 56.7 kg' and 'Maksimum (BMI 24.9): 76.3 kg'. A green box indicates 'Berat Badan Ideal' and 'Berat badan Anda sudah dalam rentang ideal!'. The 'Detail Perhitungan' section shows the formula:
$$\text{Rumus BMI: BMI} = \text{Berat (kg)} / (\text{Tinggi (m)})^2$$
 and the calculation:
$$\text{Perhitungan: } 60.0 \text{ kg} / (1.75 \text{ m})^2$$
 and the result:
$$\text{Hasil: } 60.0 / 3.0625 = 19.59$$
. A 'Kembali ke Beranda' link is in the bottom left corner.

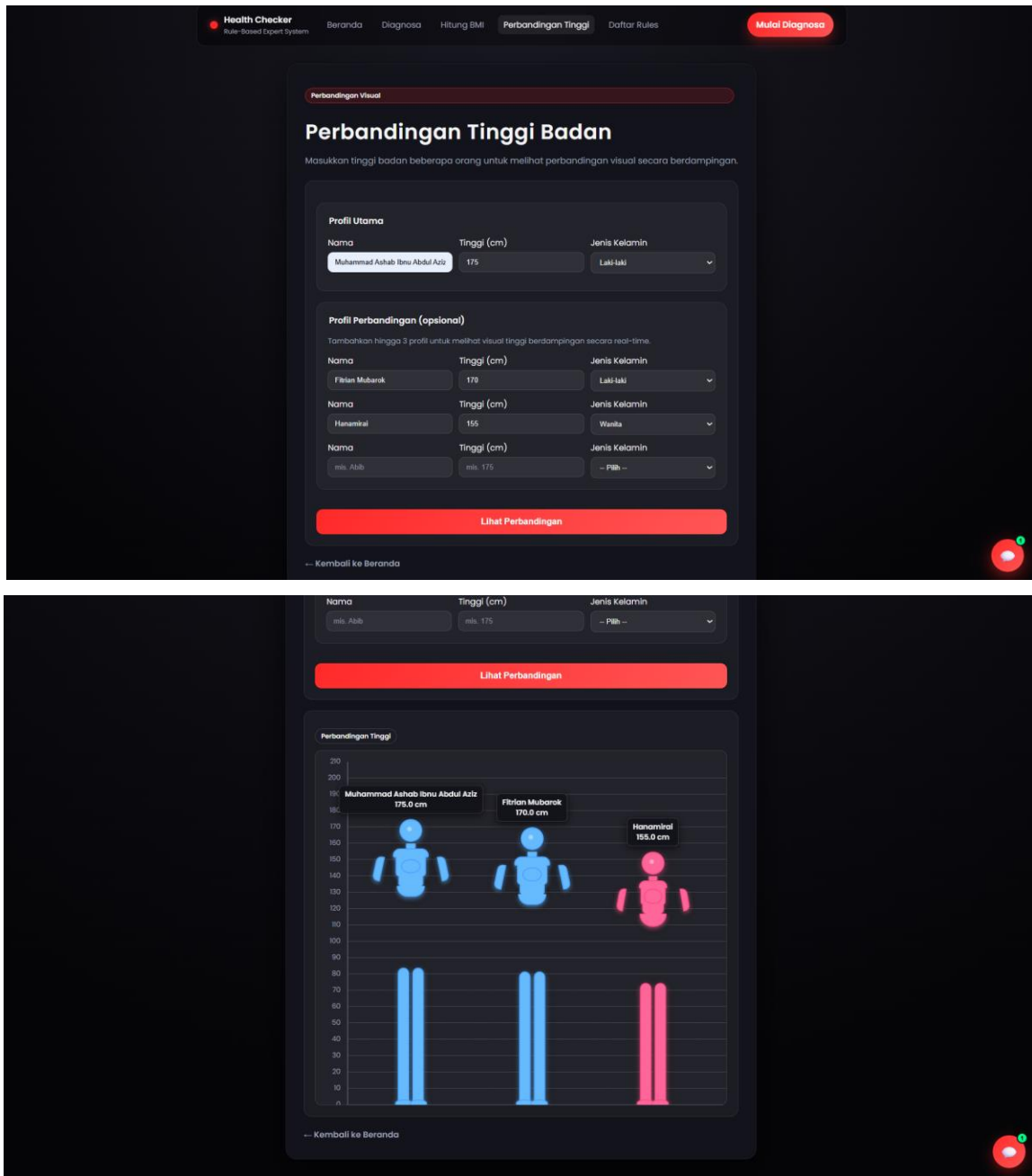


Gambar 4: Analisis BMI dan Status Berat Badan

4.3 Komparasi Tinggi Badan

- Fitur visualisasi untuk membandingkan postur tubuh pengguna dengan referensi lain.
- Masukkan nama dan tinggi badan pengguna serta data pembandingan.
- Tekan Bandingkan untuk menghasilkan grafik siluet.

Perbandingan Tinggi >

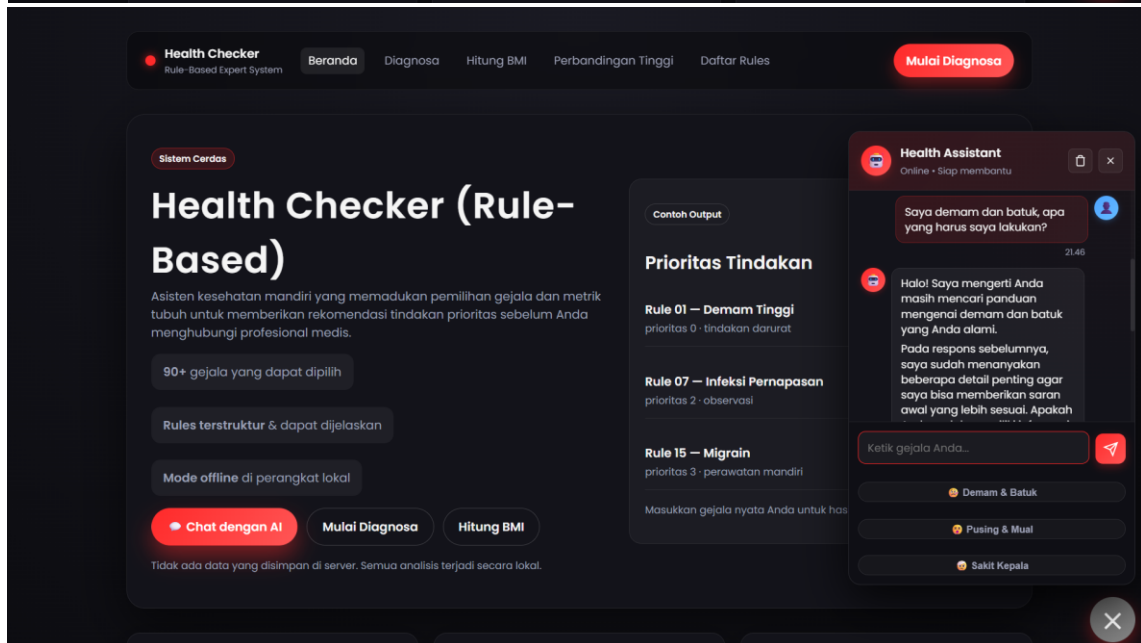
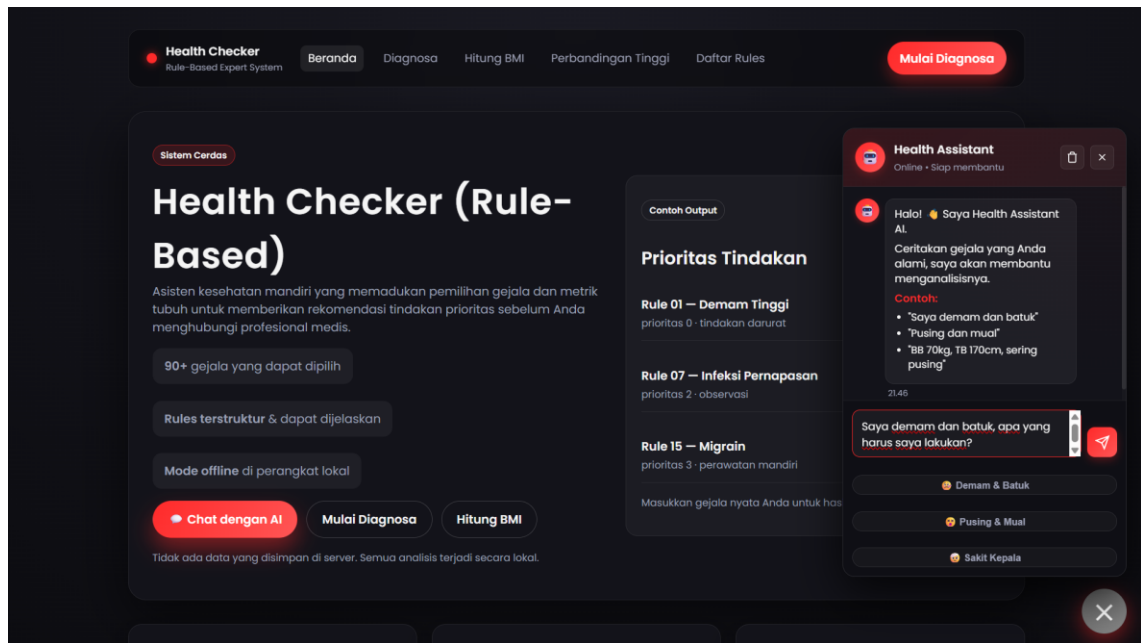


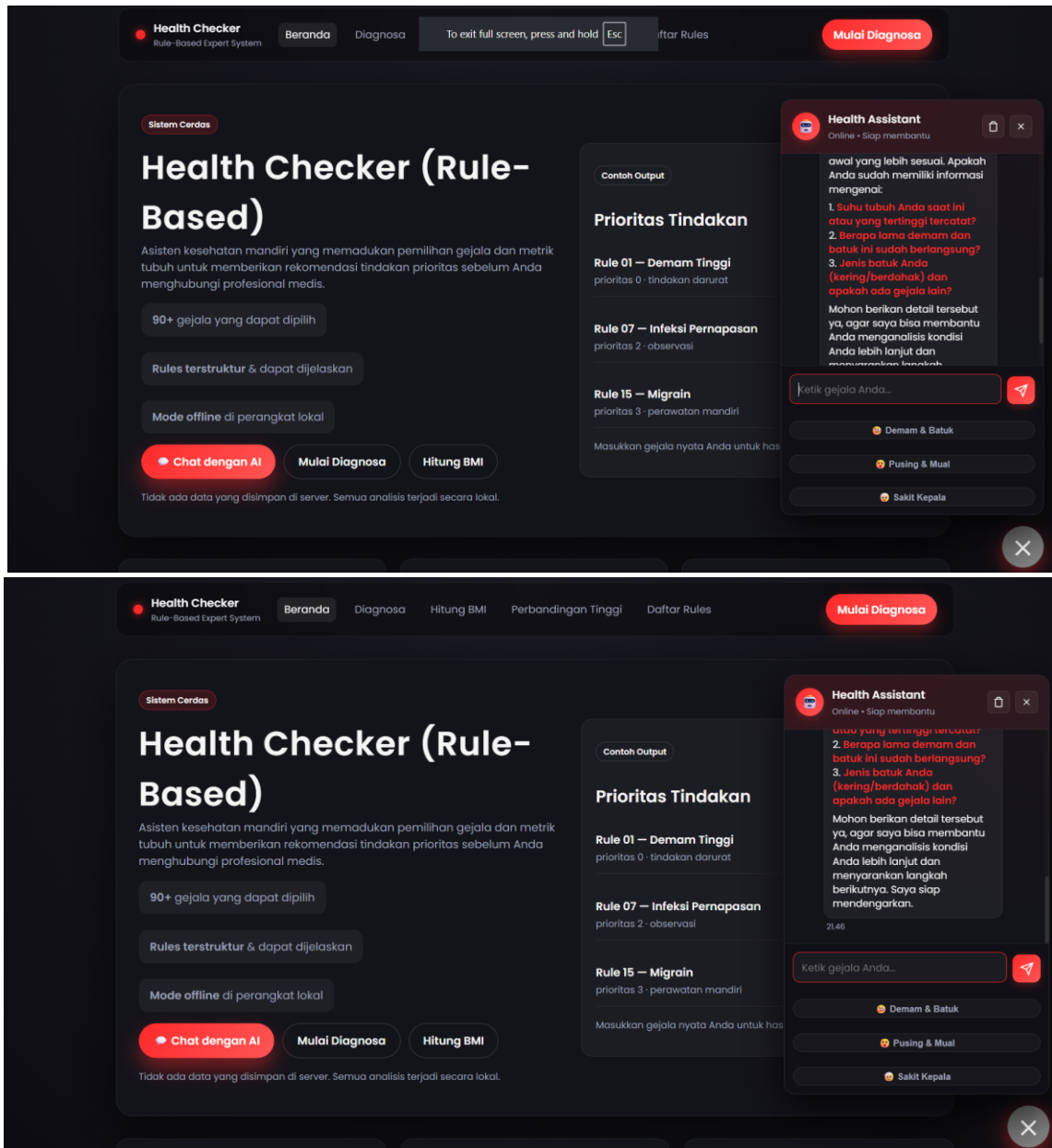
Gambar 5: Visualisasi Komparasi Tinggi Badan

4.4 Asisten Kesehatan Virtual (AI Chatbot)

- Akses widget obrolan di sudut kanan bawah layar.
- Ajukan pertanyaan kesehatan menggunakan bahasa alami.
- AI akan memberikan respons yang relevan dan informatif.

AI Chatbot >



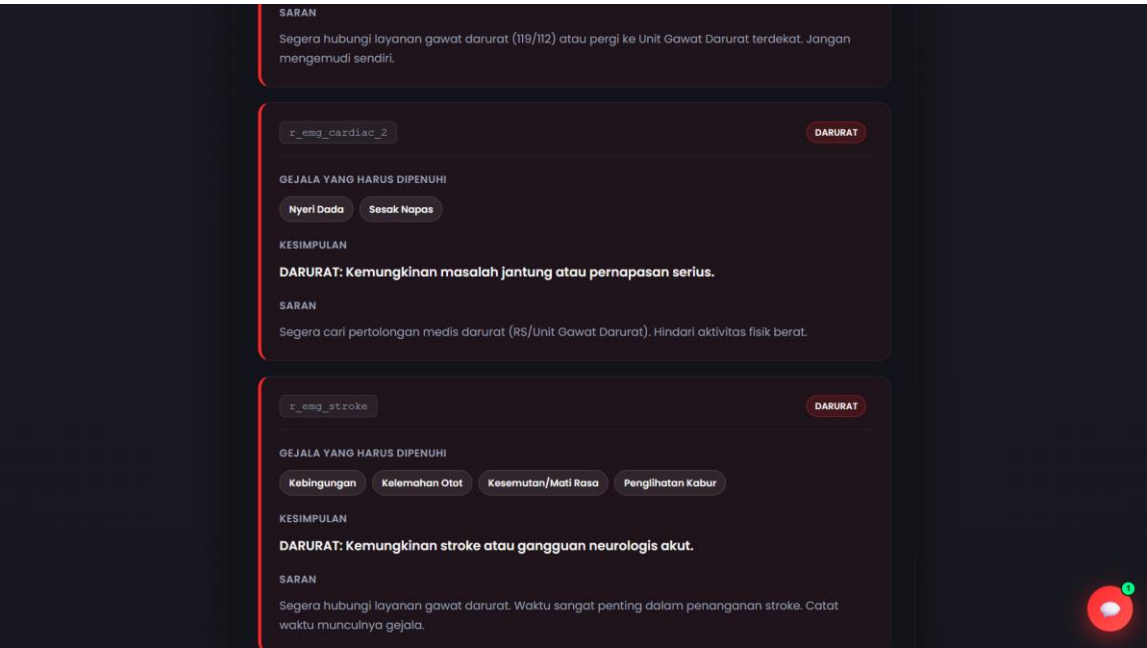
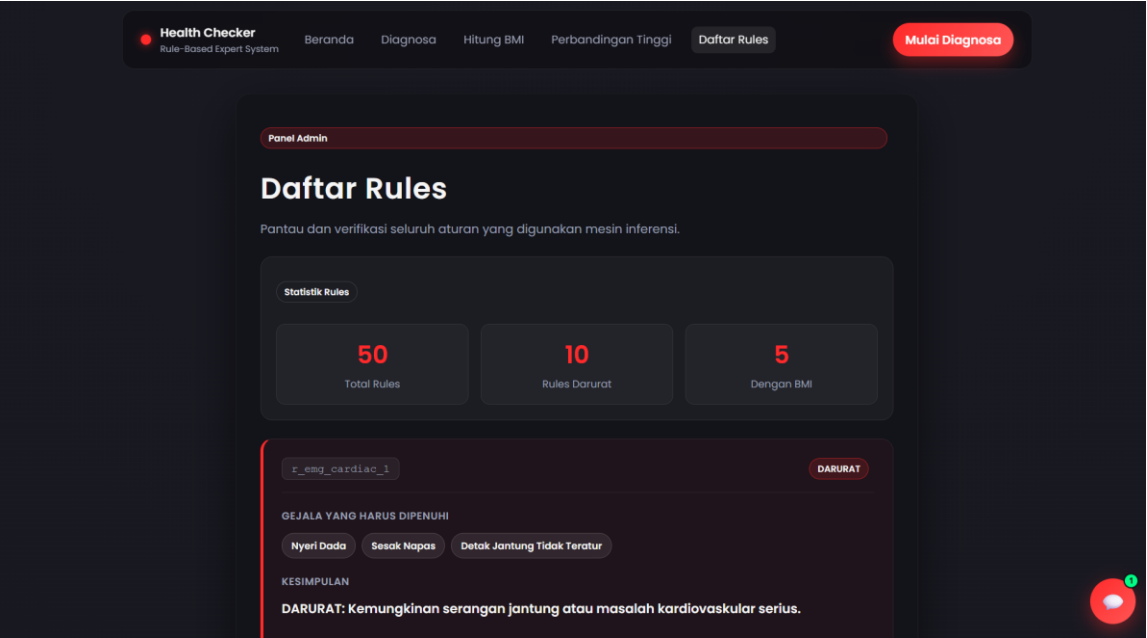


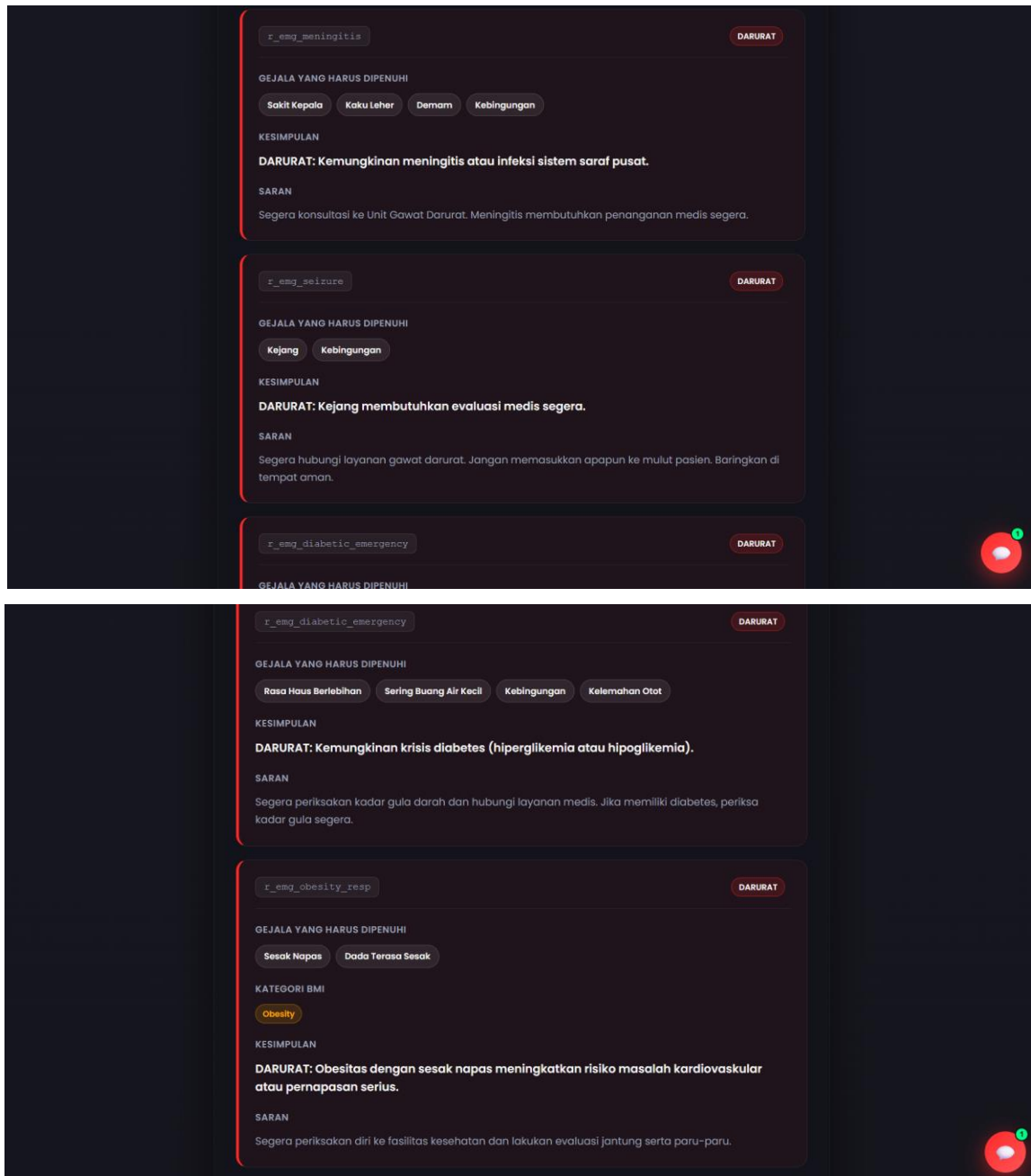
Gambar 6: Interaksi dengan Asisten Virtual

4.5 Panel Administrasi

- Dapat diakses melalui URL: **<http://localhost:5000/admin>**
- Halaman ini menyajikan transparansi logika sistem dengan menampilkan seluruh aturan (*rules*) yang aktif dalam basis pengetahuan.

Admin Panel >





Gambar 7: Tampilan Basis Pengetahuan pada Panel Admin

BAB V SPESIFIKASI TEKNIS

Bagian ini menguraikan struktur teknis dan logika yang mendasari sistem, ditujukan untuk keperluan pengembangan dan tinjauan akademis.

5.1 Arsitektur Backend

Sistem dibangun di atas kerangka kerja Flask (Python) dengan struktur modular:

- ❖ app.py: Pengendali utama (*Main Controller*) yang mengatur rute URL dan logika aplikasi.
- ❖ src/inference_engine.py: Mesin inferensi yang menjalankan algoritma pencocokan gejala terhadap basis aturan.
- ❖ src/ai_chat.py: Modul antarmuka untuk komunikasi dengan Google Gemini API.
- ❖ src/nlp_processor.py: Pemroses bahasa alami (*Natural Language Processing*) untuk ekstraksi gejala dari teks.
- ❖ src/health_metrics.py: Pustaka fungsi untuk perhitungan metrik kesehatan (BMI, BMR).

5.2 Arsitektur Frontend

Antarmuka pengguna dikembangkan menggunakan HTML5, CSS3, dan JavaScript, dengan mesin templat Jinja2:

- ❖ Desain Modular: Menggunakan base.html sebagai kerangka dasar (*layout*) untuk menjamin konsistensi tampilan antar halaman.
- ❖ Responsivitas: Desain antarmuka menyesuaikan diri dengan berbagai ukuran layar (Desktop, Tablet, Mobile).

5.3 Basis Pengetahuan (Knowledge Base)

Data aturan medis disimpan dalam format JSON pada file data/rules.json untuk fleksibilitas dan kemudahan pemeliharaan. Struktur data mencakup:

- ❖ ID Aturan
- ❖ Prioritas (Darurat, Tinggi, Sedang, Rendah)
- ❖ Kondisi (Daftar Gejala)
- ❖ Kesimpulan (Diagnosis)
- ❖ Saran Medis

5.4 Logika Sistem Pakar (Inference Engine)

Sistem menerapkan metode Forward Chaining dengan mekanisme pembobotan (*Scoring Algorithm*) untuk menentukan diagnosis paling relevan. Mekanisme Perhitungan Skor:

- ❖ Pencocokan Gejala: Menghitung interseksi antara gejala input pengguna dengan premis aturan.
- ❖ Persentase Kecocokan: $(\text{Jumlah Gejala Cocok} / \text{Total Gejala Aturan}) * 100\%$.
- ❖ Skor Keyakinan (Confidence Score): Nilai akhir yang menentukan peringkat diagnosis, dipengaruhi oleh:
 - Persentase kecocokan dasar.
 - Bonus untuk kecocokan sempurna (+50 poin).
 - Bobot prioritas (Darurat mendapat bobot tertinggi).
 - Penalti jika kriteria BMI tidak terpenuhi.
- ❖ Ambang Batas (Threshold): Sistem memfilter hasil diagnosis dengan tingkat kecocokan minimal 60%. Jika tidak ditemukan, sistem akan menerapkan mekanisme *fallback* dengan ambang batas 40%.

BAB VI PENUTUP

6.1 Batasan Sistem

Pengguna diharapkan memahami batasan-batasan berikut dalam penggunaan sistem:

1. Sifat Informatif: Hasil diagnosis yang dihasilkan sistem hanyalah prediksi algoritma dan tidak menggantikan diagnosis medis profesional oleh dokter.
2. Ketergantungan Data: Akurasi sistem bergantung sepenuhnya pada kelengkapan dan kebenaran aturan dalam basis pengetahuan. Penyakit yang tidak terdaftar tidak akan terdeteksi.
3. Konektivitas: Fitur AI Chatbot memerlukan koneksi internet aktif.
4. Validitas Input: Keakuratan hasil sangat bergantung pada ketepatan pengguna dalam memilih gejala yang dirasakan.

6.2 Pemecahan Masalah (Troubleshooting)

Kendala 1: “ModuleNotFoundError: No module named ‘flask’”

Solusi: Dependensi belum terpasang. Jalankan perintah `pip install -r requirements.txt`.

Kendala 2: Chatbot tidak merespons

Solusi:

1. Periksa koneksi internet.
2. Verifikasi validitas API Key pada file `.env`.
3. Mulai ulang aplikasi server.

Kendala 3: Tampilan antarmuka tidak sempurna pada perangkat seluler

Solusi: Lakukan penyegaran halaman (*refresh*) atau bersihkan *cache* peramban.