**Dokumentasi Proyek Chatbot Kesehatan Mental "AuraMind"**

Kecerdasan Buatan



**Disusun oleh:**

Yafi Rizky 3.34.23.1.26

Mita Salsabilla 3.34.23.1.14

**JURUSAN TEKNIK ELEKTRO**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI SEMARANG TAHUN 2025**

1. **Abstrak Proyek**

AuraMind adalah sebuah aplikasi web chatbot yang dirancang untuk menjadi pendamping kesehatan mental. Proyek ini mengimplementasikan arsitektur hybrid, menggabungkan sistem berbasis aturan (Rule-Based) untuk memberikan panduan terstruktur pada topik-topik spesifik (seperti stres, depresi, dll.) dengan model AI (Machine Learning) untuk menangani percakapan umum dan obrolan ringan. Pendekatan ini memastikan chatbot dapat memberikan dukungan yang andal dan terarah, sekaligus tetap luwes dan natural dalam berinteraksi.

Aplikasi ini dibangun menggunakan framework Flask untuk backend, HTML/CSS/JavaScript untuk frontend, dan pustaka Transformers dari Hugging Face untuk integrasi model AI.

1. **Kebutuhan Sistem & Instalasi**

Untuk menjalankan proyek ini, dibutuhkan perangkat lunak dan dependensi berikut:

1. Perangkat Lunak:
   1. Python versi 3.9 atau lebih baru
   2. pip (Package Installer for Python), biasanya sudah terinstal bersama Python
2. Instalasi Dependensi

Semua pustaka Python yang dibutuhkan telah dicantumkan dalam file requirements.txt.

* Buat file baru bernama requirements.txt di dalam folder proyek Anda (Direktori C/D:\Parent Folder\Chatbot\)
* Salin dan tempel daftar pustaka di bawah ini ke dalam file requirements.txt tersebut:

|  |
| --- |
| Flask |
| transformers |
| torch |

1. Buka terminal (Command Prompt atau PowerShell) di dalam direktori proyek Anda, lalu jalankan perintah berikut untuk menginstal semua dependensi sekaligus:

|  |
| --- |
| pip install -r requirements.txt |

1. **Unduhan Model AI**: Saat aplikasi dijalankan pertama kali, ia akan secara otomatis mengunduh model AI facebook/blenderbot\_small-90M (sekitar 350MB). Proses ini memerlukan koneksi internet.

<https://huggingface.co/facebook/blenderbot-90M>

1. Struktur Project

|  |
| --- |
| /mental\_health\_chatbot/  |  |-- app.py # File utama server Flask  |-- chatbot.py # Otak dan logika utama chatbot  |-- requirements.txt # Daftar semua dependensi Python  |  |-- /data/ # Folder untuk menyimpan semua data  | |-- emergency.json # Info kontak darurat & keyword  | |-- knowledge\_base.json# Penjelasan detail untuk kondisi tertentu  | |-- qa\_pairs.json # Pertanyaan & jawaban umum (small talk)  | `-- topics.json # Struktur topik untuk ditampilkan di UI  |  |-- /static/ # Folder untuk file statis (CSS, JS, gambar)  | `-- style.css # File untuk styling halaman web  |  `-- /templates/ # Folder untuk template HTML  `-- index.html # File struktur utama halaman web |

1. Penjelasan Detail Setiap File:
   1. App.py:

* **Tujuan Utama:** Bertindak sebagai web server yang menjalankan aplikasi dan menjadi jembatan antara antarmuka pengguna (frontend) dan logika chatbot (backend)
* **Fungsi Kunci:**

1. Flask(\_\_name\_\_): Menginisialisasi aplikasi web menggunakan framework Flask.
2. @app.route("/"): Mendefinisikan rute utama aplikasi. Saat pengguna membuka halaman awal, fungsi ini akan merender file index.html dan mengirimkan data dinamis seperti daftar hotline darurat dan kategori topik.
3. @app.route("/get\_response", methods=["POST"]): Membuat sebuah "API endpoint". Ketika pengguna mengirim pesan dari frontend, JavaScript akan mengirim permintaan ke rute ini. Rute ini kemudian memanggil fungsi chatbot.generate\_response() untuk mendapatkan balasan dan mengirimkannya kembali ke frontend dalam format JSON.
4. chatbot.py:

* **Tujuan Utama:** Ini adalah "otak" dari chatbot. File ini berisi semua logika untuk memproses input pengguna dan menghasilkan respons yang sesuai
* **Fungsi Kunci:**

1. class MentalHealthChatbot: Kelas utama yang menampung semua fungsionalitas chatbot.
2. \_\_init\_\_(): Konstruktor yang dieksekusi saat chatbot dibuat. Fungsinya adalah memuat semua data dari file JSON dan memanggil \_load\_ml\_model() untuk menyiapkan model AI.
3. \_load\_ml\_model(): Mencoba memuat model AI blenderbot\_small-90M dari Hugging Face. Jika gagal (misalnya, tidak ada internet), chatbot akan tetap berjalan dalam mode non-AI.
4. \_initialize\_guided\_flows(): Mendaftarkan semua alur percakapan multi-langkah yang terstruktur untuk setiap topik (misalnya, \_stres\_step0, \_depresi\_step1, dll.).
5. generate\_response(): Fungsi sentral yang dieksekusi untuk setiap pesan pengguna. Ia memiliki urutan prioritas yang jelas:
   * + 1. Cek kata kunci darurat.
       2. Lanjutkan alur percakapan yang sedang aktif.
       3. Mulai alur baru jika input cocok dengan nama topik.
       4. Jawab obrolan ringan dari qa\_pairs.json.
       5. Sarankan topik jika input mengandung kata kunci relevan.
       6. **Gunakan model AI BlenderBot** jika tidak ada aturan lain yang cocok.
       7. Berikan respons fallback terakhir jika semua gagal.
6. Berkas Data (/data/\*.json)

* **Tujuan Utama:** Memisahkan data dari kode logika. Ini membuat chatbot lebih mudah dikelola dan diperbarui tanpa harus mengubah file Python.
* emergency.json: Berisi daftar kontak darurat yang ditampilkan di UI dan kata kunci untuk mendeteksi situasi krisis.
* topics.json: Mendefinisikan struktur, nama, dan deskripsi dari tombol-tombol topik yang ditampilkan di sisi kanan UI.
* qa\_pairs.json: Berfungsi sebagai "kamus" untuk obrolan ringan. Berisi pasangan pola pertanyaan dan kemungkinan jawaban.
* knowledge\_base.json: Basis pengetahuan yang berisi definisi dan saran mendalam untuk kondisi-kondisi tertentu seperti Stres, Insomnia, dan BPD.

1. Cara Menjalankan Aplikasi
   1. Buka terminal dan navigasikan ke direktori utama proyek (Direktori C/D:\Parent Folder\mental\_health\_chatbot)
   2. Pastikan semua dependensi sudah terinstal
   3. Ketikan ‘Python app.py’
   4. Lalu buka Browser **http://127.0.0.1:5000**