Arsitektur Chatbot & Mekanisme Chatbot Sapa Tazkia

Penjelasan Komponen Sistem

A. Frontend (Presentation Layer)

- Teknologi: React.js atau Vue.js dengan Tailwind CSS
- Fungsi: Interface yang dilihat dan digunakan oleh user
- Fitur: Chat window, login form, history panel, responsive design
- Karakteristik: Dapat diakses dari desktop, tablet, dan smartphone

B. Backend Server (Application Layer)

- Hosting: Vultr Cloud Compute (2 vCPU, 4 GB RAM, 50 GB SSD)
- Framework: Python FastAPI
- Fungsi: Memproses semua request, koordinasi antar komponen
- Modul Utama:
    - Authentication: Mengelola login/logout dan session management
    - RAG Module: Retrieval-Augmented Generation untuk pencarian dokumen
    - AI Processing: Interface ke AI API eksternal
    - Database Handler: Query dan manipulasi data
    - PDF Generator: Generate dokumen transkrip dalam format PDF

C. AI API (External Service)

- Provider: Google Gemini Flash-8B atau OpenAI GPT-4o mini
- Fungsi: Natural Language Processing dan text generation
- Karakteristik: Response time kurang dari 2 detik, handle 30.000 chat/bulan

D. Vector Database (Qdrant)

- Fungsi: Menyimpan embedding dokumen kampus untuk pencarian semantik
- Data: Buku panduan, SOP, FAQ, informasi program studi, dll
- Teknologi: Similarity search dengan cosine similarity

E. Relational Database (PostgreSQL/MySQL)

- Fungsi: Menyimpan data terstruktur
- Data: Akun mahasiswa, nilai akademik, transkrip, chat history

Mekanisme Kerja ChatBot

2.1 Alur Umum Percakapan

Berikut adalah tahapan proses ketika user berinteraksi dengan chatbot Sapa Tazkia:

Tahap 1: User Mengajukan Pertanyaan

User membuka aplikasi Sapa Tazkia melalui web browser dan mengetik pertanyaan. Contoh:

- "Apa saja program studi di STMIK Tazkia?"
- "Bagaimana cara mendaftar sebagai mahasiswa baru?"
- "Berapa biaya kuliah per semester?"

Tahap 2: Request Dikirim ke Backend

Pertanyaan dikirim melalui HTTPS (terenkripsi) dari browser ke backend server. Server menerima dan melakukan validasi:

- Validasi format input (tidak kosong, panjang maksimal 500 karakter)
- Rate limiting check (maksimal 50 chat per user per hari)
- Session validation (jika fitur memerlukan login)

Tahap 3: Pencarian Dokumen Relevan (RAG Process)

Sistem melakukan Retrieval-Augmented Generation (RAG):

a. Generate embedding untuk pertanyaan user menggunakan model embedding b. Query Vector Database (Qdrant) untuk mencari dokumen relevan c. Ambil top 5 dokumen dengan similarity score tertinggi (threshold > 0.7) d. Compile context dari dokumen-dokumen tersebut

Contoh hasil retrieval:

- Dokumen: "Buku Panduan Akademik 2024/2025", Section: "Program Studi"
- Similarity Score: 0.92 (sangat relevan)

Tahap 4: AI Processing

Sistem mengirim request ke AI API dengan format:

- User question: pertanyaan asli dari user
- Context: dokumen-dokumen relevan dari RAG
- System prompt: instruksi untuk menjawab berdasarkan context saja

AI memproses dan menghasilkan jawaban yang:

- Natural dan mudah dipahami
- Berdasarkan dokumen yang diberikan (tidak mengarang)
- Terstruktur dengan baik (menggunakan numbering/bullet points jika perlu)

Tahap 5: Response ke User

Jawaban dari AI dikirim kembali ke frontend dan ditampilkan dalam chat window dalam waktu kurang dari 2 detik.

Contoh response:

STMIK Tazkia memiliki 2 program studi:
1. Sistem Informasi (S1)
2. Teknik Informatika (S1)

Setiap program studi memiliki kurikulum yang disesuaikan dengan kebutuhan industri. Apakah Anda ingin informasi lebih detail

tentang salah satu program studi?

2.2 Keunggulan Pendekatan RAG

1. **Akurasi Tinggi** Jawaban selalu berdasarkan dokumen resmi kampus, bukan informasi umum dari internet atau "kreasi" AI.
2. **No Hallucination** Jika informasi tidak ada dalam knowledge base, sistem akan jujur menyatakan "tidak tahu" dan menyarankan kontak ke bagian terkait.
3. **Always Up-to-date** Knowledge base dapat diupdate sewaktu-waktu dengan dokumen terbaru tanpa perlu retrain model AI.
4. **Traceable** Setiap jawaban dapat dilacak ke dokumen sumbernya untuk verifikasi.