Dokumentasi Arsitektur Retrieval-Augmented Generation (RAG)

Chatbot AI Sapa Tazkia - STMIK Tazkia Bogor

Tanggal Dokumen: 9 Oktober 2025
Disusun oleh: Tim Development Sapa Tazkia

---

1. Pendahuluan

Dokumen ini menjelaskan arsitektur teknis sistem Retrieval-Augmented Generation (RAG) yang diterapkan dalam Chatbot Sapa Tazkia. RAG adalah pendekatan hybrid yang menggabungkan mekanisme **retrieval** (pengambilan dokumen relevan) dengan **generation** (pembuatan teks oleh AI) untuk menghasilkan jawaban yang akurat, berbasis fakta, dan dapat dilacak ke sumbernya.

---

2. Definisi RAG

Retrieval-Augmented Generation (RAG) adalah teknik yang memungkinkan Large Language Model (LLM) untuk:

- Mengambil informasi relevan dari Knowledge Base (dokumen kampus)
- Menggunakan informasi tersebut sebagai konteks dalam pembuatan jawaban
- Menghasilkan respons yang akurat, aman dari halusinasi, dan selalu up-to-date

Dalam konteks Sapa Tazkia, RAG memastikan bahwa setiap jawaban chatbot:

1. **Berbasis Dokumen Resmi:** Semua jawaban diambil dari dokumen kampus yang terautorisasi
2. **Dapat Dilacak:** Setiap jawaban dapat dikaitkan dengan sumber dokumennya
3. **Tidak Mengarang:** Jika informasi tidak ada, sistem akan jujur menyatakan keterbatasannya
4. **Mudah Diperbarui:** Dokumen dapat diperbarui tanpa perlu melatih ulang model AI

---

3. Komponen Arsitektur RAG Sapa Tazkia

3.1 Layer 1: Client (Presentation Layer)

Komponen:

- Web Browser (user interface)
- Chat Interface (Vue.js + Tailwind CSS)

Fungsi: Menyediakan antarmuka pengguna yang intuitif dan responsif untuk berinteraksi dengan chatbot.

Karakteristik:

- Responsif di desktop, tablet, dan smartphone
- Real-time chat display
- History panel untuk riwayat percakapan
- Tombol export PDF untuk nilai/transkrip

3.2 Layer 2: API Gateway (Application Layer)

Komponen:

- FastAPI Server (2 vCPU, 4GB RAM)
- Request Handler
- Rate Limiter

Fungsi: Mengelola semua request dari client, validasi, dan koordinasi antar komponen sistem.

Karakteristik:

- Protocol: HTTPS (enkripsi end-to-end)
- Response time target: <2 detik
- Rate limit: 50 chat per user per hari
- Menangani ±100 pengguna per hari

3.3 Layer 3: Authentication & Session Management

Komponen:

- Login Module
- Session Manager

- User Validation

Fungsi: Menangani autentikasi mahasiswa dan calon mahasiswa, serta manajemen session.

Alur:

1. User memasukkan NIM dan password
2. Sistem validasi dengan database akademik
3. Jika valid, session token dibuat
4. Session dipertahankan untuk percakapan berlanjutan

3.4 Layer 4: Data Sources (Knowledge Base)

Komponen:

- Dokumen Kampus (Buku Panduan, SOP, FAQ, dll)
- Program Studi
- Kalender Akademik
- Informasi Fasilitas

Format Data:

- PDF, TXT, DOC
- Terstruktur dengan metadata (tanggal, versi, kategori)

3.5 Layer 5: Preprocessing & Indexing

Proses:

5.1 Text Extraction

- Ekstrak teks dari berbagai format dokumen (PDF, DOC, TXT)
- Pertahankan struktur dan formatting

5.2 Text Cleaning

- Hapus noise, karakter spesial yang tidak perlu
- Normalisasi spasi dan line breaks
- Hapus header/footer yang tidak relevan

5.3 Chunking

- Bagi dokumen besar menjadi potongan (chunks) berukuran ~500 karakter

- Overlap antar chunks untuk menjaga konteks
- Setiap chunk disimpan dengan metadata (source, section, timestamp)

5.4 Embedding Generation

- Ubah setiap chunk menjadi vektor numerik (embedding) menggunakan embedding model
- Embedding berukuran dimensi tinggi (misalnya 768 atau 1536 dimensi)
- Vektor ini merepresentasikan makna semantik teks

3.6 Layer 6: Vector Database (Qdrant)

Komponen:

- Qdrant Vector Store
- Document Chunks + Embeddings
- Metadata Index

Fungsi:

- Menyimpan embedding vektor dari semua chunks dokumen
- Menyimpan metadata (source document, section, timestamp)
- Memungkinkan pencarian semantik cepat dengan similarity search

Keunggulan Qdrant:

- Open-source dan self-hosted
- Performa tinggi untuk similarity search
- Dukungan filtering metadata
- Mudah diintegrasikan dengan Python

3.7 Layer 7: Retrieval Pipeline (Inti RAG)

Alur Retrieval:

Step 1: Query Embedding

- User question: "Apa saja program studi di Tazkia?"
- Convert ke embedding menggunakan model yang sama dengan saat indexing
- Embedding berukuran sama (misalnya 768 dimensi)

Step 2: Similarity Search

- Query vector dicari di Vector Database (Qdrant)
- Menghitung cosine similarity antara query vector dengan semua document vectors
- Rumus: similarity = (A·B) / (||A|| × ||B||), range 0-1

Step 3: Ranking & Filtering

- Urutkan dokumen berdasarkan similarity score (tertinggi ke terendah)
- Filter dokumen dengan threshold similarity > 0.7 (70%)
- Ambil top-5 dokumen paling relevan

Step 4: Output Retrieval

- Bawa dokumen terpilih ke layer berikutnya (Generation)
- Sertakan metadata (source, relevance score)

Contoh Hasil Retrieval:

Top-1: "Buku Panduan Akademik 2024/2025" - Section "Program Studi"
Similarity: 0.92 (Sangat Relevan)

Top-2: "Panduan Pendaftaran 2024" - Section "Program yang Tersedia"
Similarity: 0.88 (Relevan)

Top-3: "FAQ Mahasiswa Baru" - Q&A tentang Program Studi
Similarity: 0.84 (Relevan)

3.8 Layer 8: Context & Prompt Engineering

Komponen:

- Format Retrieved Docs
- System Prompt Injection
- Prompt Construction

Fungsi: Mengkombinasikan user query dengan dokumen yang diambil untuk membuat prompt yang efektif.

Struktur Prompt:

SYSTEM PROMPT:
Anda adalah asisten virtual Sapa Tazkia, chatbot resmi STMIK Tazkia Bogor.
Tugas Anda menjawab pertanyaan pengguna berdasarkan dokumen kampus yang diberikan.
PENTING: Jawab HANYA berdasarkan informasi di dokumen berikut.
Jika informasi tidak ada, katakan "Saya tidak memiliki informasi tentang hal ini"

dan sarankan pengguna menghubungi bagian terkait.

RETRIEVED CONTEXT:
[Dokumen 1 dari Qdrant]
[Dokumen 2 dari Qdrant]
[Dokumen 3 dari Qdrant]
...

USER QUESTION:
Apa saja program studi di Tazkia?

INSTRUCTION:
- Berikan jawaban yang jelas dan terstruktur
- Gunakan numbering atau bullet points untuk daftar
- Berikan informasi terkait (misalnya jenjang pendidikan)
- Jika pengguna adalah mahasiswa login, tanyakan apakah mereka ingin info lebih detail

---

3.9 Layer 9: AI Processing (LLM)

Pilihan AI Provider:

- **Gemini Flash-8B API:** Biaya Rp 18.000/bulan, performa cukup baik
- **GPT-4o mini API:** Biaya Rp 25.000/bulan, kualitas respons lebih natural

Proses AI:

1. Menerima prompt terformat dari Layer 8
2. Melakukan natural language processing dan understanding
3. Menghasilkan respons berdasarkan konteks dokumen
4. Memastikan respons coherent, grammatically correct, dan sesuai instruksi system prompt
5. Mengirim respons kembali dalam format teks

Karakteristik:

- Response time: <2 detik per query
- Kapasitas: ±30.000 chat per bulan

- Model tidak mengarang (hallucination-free) karena terikat pada dokumen

---

3.10 Layer 10: Response Handling

Komponen:

- AI Generated Response
- Citation Extraction
- Format Response
- Add Context Memory

Proses:

Step 1: Extract Citation

- Identifikasi dokumen sumber dari response
- Catatan: "Jawaban ini didasarkan pada Buku Panduan Akademik 2024/2025"

Step 2: Format Response

- Clean up response dari model AI
- Tambahkan formatting (bold, italic, bullet points jika perlu)
- Pastikan readability dan clarity

Step 3: Context Memory

- Simpan user question dan AI response ke chat history
- Update conversation context untuk follow-up questions
- Mahasiswa login dapat melihat history percakapan mereka

Step 4: Save to Database

- Simpan ke chat history di PostgreSQL/MySQL
- Catat timestamp, user ID, session ID

3.11 Layer 11: Database Layer (Relational)

Teknologi: PostgreSQL atau MySQL

Tabel-tabel:

- **users:** User accounts, NIM, nama, email, password hash
- **academic_data:** Nilai, IPK, status akademik per mahasiswa

- **chat_history:** Riwayat percakapan pengguna
- **document_metadata:** Informasi dokumen di Knowledge Base (version, updated_at, category)

Fungsi:

- Menyimpan data user dan kredensial untuk autentikasi
- Menyimpan nilai akademik mahasiswa
- Menyimpan riwayat chat untuk transparansi dan audit

3.12 Layer 12: Additional Services

Komponen:

- PDF Generator

Fungsi:

- Generate transkrip/nilai dalam format PDF
- User (mahasiswa yang sudah login) dapat mengunduh nilai mereka
- Format profesional sesuai standar akademik

3.13 Layer 13: Client Output

Komponen:

- Response in Chat
- Display to User

Hasil Akhir:

- Jawaban ditampilkan di chat interface dalam waktu <2 detik
- Real-time display dengan smooth animation
- User dapat melanjutkan percakapan atau export data

---

4. Alur End-to-End Sistem RAG Sapa Tazkia

Skenario 1: User Calon Mahasiswa Bertanya Tentang Program Studi

1. User membuka sapatazkia.ac.id di browser

2. Input: "Apa saja program studi di Tazkia?"
3. FastAPI menerima request via HTTPS
4. Query embedding: "Apa saja program studi di Tazkia?" → vektor 768 dimensi
5. Qdrant similarity search → temukan top-5 dokumen relevan
6. Hasilnya: Buku Panduan, FAQ, Info Program Studi (similarity 0.92, 0.88, 0.84)
7. Prompt Construction:
   - System: "Jawab berdasarkan dokumen berikut..."
   - Context: [Dokumen program studi]
   - Question: "Apa saja program studi di Tazkia?"
8. Gemini/GPT-4o mini API process → generate response
9. Response: "STMIK Tazkia memiliki 3 program studi:
   1. Sistem Informasi (S1)
   2. Teknik Informatika (S1)
   3. Manajemen Informatika (D3)
   Apakah Anda ingin informasi lebih detail tentang salah satu program?"
10. Response ditampilkan di chat (<2 detik)
11. Chat history disimpan (jika user login)

Skenario 2: Mahasiswa Login dan Cek Nilai

1. Mahasiswa login dengan NIM dan password
2. Session validated
3. Input: "Tampilkan nilai saya semester ini"
4. Query embedding → Qdrant search
5. Retrieved: Dokumen tentang akademik, cara cek nilai
6. Simultaneously: Database query ke academic_data
7. Prompt Construction: Query + dokumen + data nilai
8. AI response: "Nilai Anda semester ini:
   - Algoritma: A (4.0)
   - Database: A- (3.7)
   - Web Programming: B+ (3.3)
   IPK Semester: 3.67"
9. User dapat click "Download as PDF"
10. PDF Generator fetch academic_data → generate professional PDF transkrip
11. PDF downloaded ke komputer user

---

5. Keunggulan Pendekatan RAG untuk Sapa Tazkia

5.1 Akurasi Tinggi

- Semua jawaban berbasis dokumen resmi kampus

- Tidak ada hallucination atau pembikiuan informasi
- Konsistensi respons terjaga

5.2 Transparansi & Traceable

- Setiap jawaban dapat dilacak ke dokumen sumber
- User tahu jawaban dari mana berasal
- Audit trail lengkap tersimpan di database

5.3 Easy Maintenance & Update

- Knowledge base dapat diperbarui kapan saja
- Tidak perlu retraining model AI
- Cukup update dokumen dan re-index ke Vector Database

5.4 Privasi & Keamanan

- Dapat di-host on-premise (Vector DB lokal)
- Tidak perlu mengirim data sensitif ke cloud storage eksternal
- Semua data akademik tetap aman di server kampus

5.5 Scalable

- Vector Database dapat menangani jutaan chunks dokumen
- API-based AI service dapat scale sesuai load
- Cloud infrastructure dapat di-upgrade sesuai kebutuhan

---

6. Batasan & Limitasi RAG

6.1 Kualitas Dokumen

- RAG hanya sebaik dokumen yang disimpannya
- Jika dokumen outdated, jawaban juga akan outdated
- Diperlukan discipline untuk update regular

6.2 Kapasitas API

- Gemini/GPT-4o mini memiliki kuota bulanan (±30.000 chat/bulan)
- Rate limit: ±100 pengguna per hari
- Jika melebihi, service akan throttle atau berbayar additional

6.3 Konteks Terbatas

- LLM memiliki context window terbatas (tidak bisa membaca seluruh dokumen sekaligus)
- Hanya top-5 dokumen paling relevan yang dipakai
- Informasi di tengah dokumen bisa terlewatkan jika tidak di-chunk dengan baik

6.4 Query Complexity

- Pertanyaan kompleks yang membutuhkan reasoning multi-step bisa sulit dijawab
- Pertanyaan yang ambigu perlu disambiguasi
- Pertanyaan out-of-domain akan dijawab "tidak tahu"

---

7. Tech Stack & Tools

Komponen	Teknologi	Alasan
Frontend	Vue.js + Tailwind CSS	Lightweight, responsive, mudah maintain
Backend	FastAPI (Python)	Cepat, async support, mudah integrasi AI
Vector DB	Qdrant	Open-source, performant, Python SDK tersedia
Relational DB	PostgreSQL/MySQL	Reliable, scalable, free
AI API	GPT-4o mini (OpenAI)	Kualitas respons tinggi, performa stabil, cost-effective
Hosting	Vultr Cloud Compute	Affordable, Indonesia-available, reliable
Version Control	GitHub	Standard industry practice
Containerization	Docker	Reproducible deployment, easy scaling
SSL	Let's Encrypt	Free, automatic renewal

---

8. Estimasi Biaya Operasional

Item	Spesifikasi	Biaya/Bulan
Hosting	Vultr 2vCPU, 4GB RAM, 50GB SSD, 1TB BW	Rp 320.000
AI API (GPT-4o mini)	±100 user/day, ±30.000 chat/month	Rp 25.000
Total		**Rp 345.000/bulan**

Total Biaya 4 Bulan (Periode Proyek):

- Operasional: Rp 345.000 × 4 = Rp 1.380.000
- Initial (domain, setup): Rp 150.000
- **Grand Total: ~Rp 1.530.000**

---

9. Kesimpulan

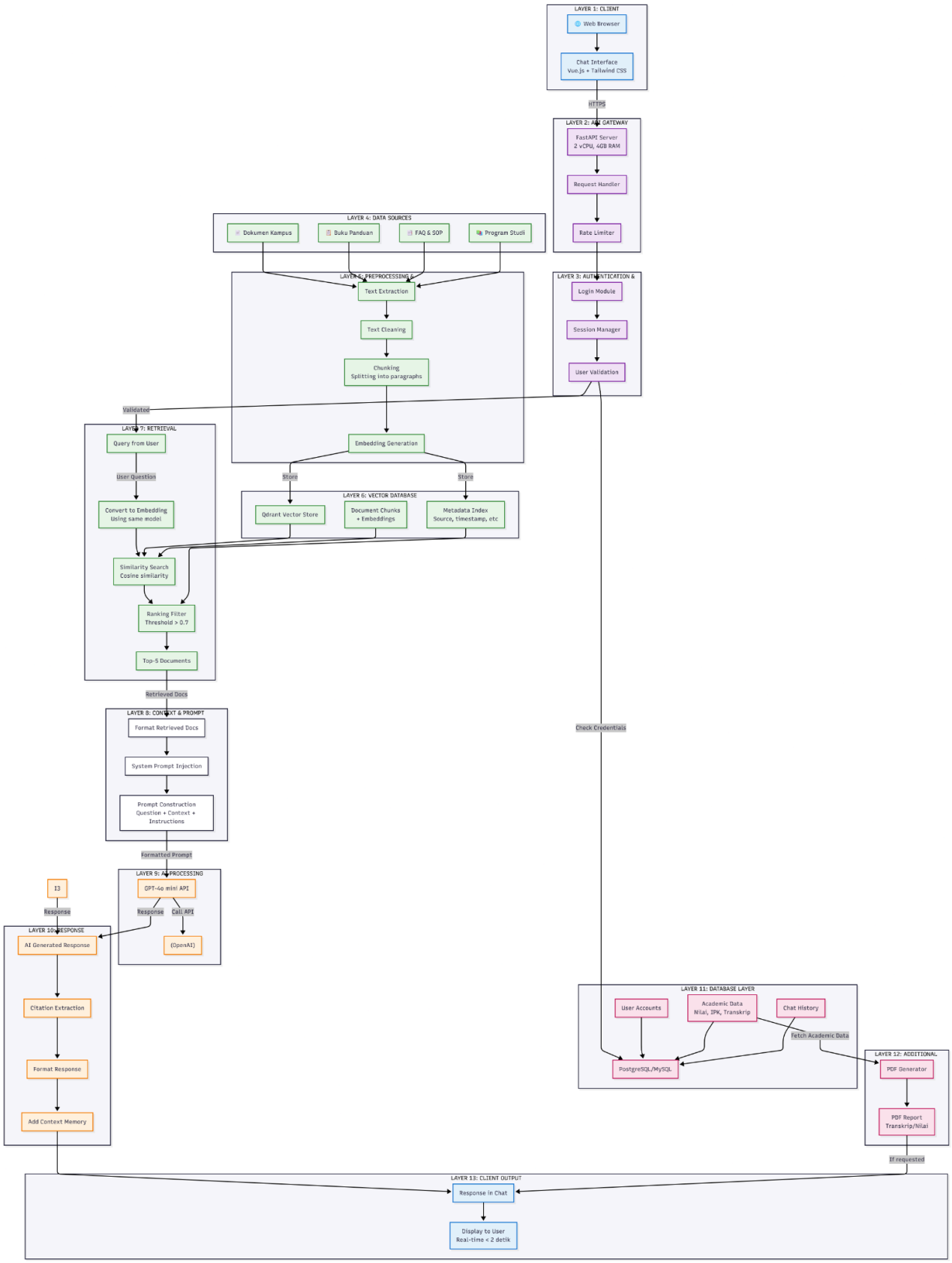
Arsitektur RAG yang diterapkan pada Sapa Tazkia memberikan solusi optimal untuk:

1. **Memberikan jawaban akurat** berdasarkan dokumen resmi kampus
2. **Mengurangi hallucination** dengan context-aware generation
3. **Meningkatkan transparansi** dengan citation dan traceable answers
4. **Memudahkan maintenance** dengan update dokumen tanpa retrain
5. **Menjaga keamanan data** dengan on-premise deployment option
6. **Memberikan nilai tambah** kepada mahasiswa dan calon mahasiswa melalui layanan 24/7

Pendekatan ini telah terbukti efektif pada berbagai aplikasi enterprise dan siap untuk implementasi di Sapa Tazkia Bogor.

---

9. Lampiran Diagram


LAYER 1: CLIENT
Web Browser
Chat Interface
Vue.js + Tailwind CSS
HTTPS
LAYER 2: API GATEWAY
FastAPI Server
2 vCPU, 4GB RAM
Request Handler
Rate Limiter
LAYER 4: DATA SOURCES
Dokumen Kampus
Buku Panduan
FAQ & SOP
Program Studi
LAYER 5: PREPROCESSING &
Text Extraction
Text Cleaning
Chunking
Splitting into paragraphs
Embedding Generation
LAYER 3: AUTHENTICATION &
Login Module
Session Manager
User Validation
Validated
Store
Store
LAYER 6: VECTOR DATABASE
Qdrant Vector Store
Document Chunks
+ Embeddings
Metadata Index
Source, timestamp, etc
LAYER 7: RETRIEVAL
Query from User
User Question
Convert to Embedding
Using same model
Similarity Search
Cosine similarity
Ranking Filter
Threshold > 0.7
Top-5 Documents
Retrieved Docs
LAYER 8: CONTEXT & PROMPT
Format Retrieved Docs
System Prompt Injection
Prompt Construction
Question + Context +
Instructions
Formatted Prompt
LAYER 9: AI PROCESSING
GPT-4o mini API
Response
Call API
(OpenAI)
I3
Response
LAYER 10: RESPONSE
AI Generated Response
Citation Extraction
Format Response
Add Context Memory
Check Credentials
LAYER 11: DATABASE LAYER
User Accounts
Academic Data
Nilai, IPK, Transkrip
Chat History
Fetch Academic Data
PostgreSQL/MySQL
LAYER 12: ADDITIONAL
PDF Generator
PDF Report
Transkrip/Nilai
If requested
LAYER 13: CLIENT OUTPUT
Response in Chat
Display to User
Real-time < 2 detik