

# Dokumen Functional Requirement (FR)

Proyek: Sapa Tazkia Chatbot AI Berbasis Web

## 1. Pendahuluan Proyek

### 1.1. Tujuan Aplikasi

Tujuan utama proyek ini adalah menerapkan **Large Language Model (LLM)** untuk membangun chatbot interaktif **Sapa Tazkia** sebagai asisten virtual kampus berbasis web. Aplikasi ini bertujuan untuk mengatasi keterbatasan akses informasi akademik yang tersebar dan tidak tersedia 24/7, sehingga mahasiswa dan calon pendaftar Tazkia Bogor dapat memperoleh layanan informasi kampus (akademik, administrasi, pendaftaran) secara **cepat, otomatis, aman**, dan tersedia **24/7**.

### 1.2. Lingkup Fungsional

Aplikasi ini akan mencakup pengembangan chatbot berbasis web dengan integrasi **Retrieval-Augmented Generation (RAG)** terhadap dokumen kampus, fitur **autentikasi** pengguna (mahasiswa), dan kemampuan **ekspor nilai/transkrip** otomatis.

### 1.3. Kriteria Keberhasilan (Success Criteria)

- Akurasi Jawaban:** Chatbot dapat menjawab  $\geq 90\%$  pertanyaan umum dengan akurat.
- Kecepatan Respon:** Waktu respon chatbot  $< 2$  detik (realtime).
- Kapasitas Layanan:** Sistem dapat menangani minimal **100 pengguna per hari**.
- Fungsionalitas Utama:** Mahasiswa dapat mengakses dan mengunduh nilai/transkrip dalam bentuk PDF.

## 2. Aktor Pengguna (User Roles)

Aktor	Deskripsi	Fungsi Utama
<b>Mahasiswa</b>	Pengguna terdaftar yang memiliki akun akademik.	Mengajukan pertanyaan, login, melihat nilai, mengunduh transkrip/nilai.

<b>Calon Pendaftar</b>	Pengguna anonim/belum terdaftar.	Mengajukan pertanyaan seputar pendaftaran, program studi, dan fasilitas kampus.
<b>Admin Akademik</b>	Pengguna internal yang mengelola basis pengetahuan (knowledge base).	Mengelola/memperbarui dokumen kampus (korpus data), memantau performa chatbot (tidak termasuk dalam <i>in-scope</i> pengembangan <i>Admin Panel</i> ).
<b>Tim Developer</b>	Pengelola teknis sistem.	Monitoring server, fine-tuning model AI, <i>bug fix</i> .

### 3. Kebutuhan Fungsional (Functional Requirements)

#### 3.1. Fungsi Chatbot Dasar (General Information)

FR-ID	Deskripsi Fungsional	Aktor
FR-01	Sistem harus menyediakan <b>antarmuka chat</b> berbasis web yang intuitif.	Semua
FR-02	Sistem harus mampu memproses pertanyaan dalam <b>bahasa sehari-hari</b> (natural language processing).	Semua
FR-03	Sistem harus <b>mengambil data dari dokumen kampus (RAG)</b> untuk menjawab pertanyaan terkait: Lokasi, Program Studi, Jadwal Akademik, Kalender, Panduan Pendaftaran, Fasilitas, dan Unit Layanan.	Calon Pendaftar, Mahasiswa
FR-04	Sistem harus memiliki <b>memori kontekstual</b> untuk mempertahankan alur percakapan dalam sesi yang sama ( <i>Context Memory Conversation</i> ).	Mahasiswa, Calon Pendaftar
FR-05	Sistem harus menyediakan fitur <b>history percakapan</b> yang dapat diakses kembali oleh Mahasiswa yang <i>login</i> .	Mahasiswa

### 3.2. Fungsi Autentikasi & Akses Data Akademik

FR-ID	Deskripsi Fungsional	Aktor
FR-06	Sistem harus menyediakan <b>modul autentikasi (Login)</b> untuk Mahasiswa dengan kredensial yang terintegrasi dengan database akademik.	Mahasiswa
FR-07	Setelah <i>login</i> , sistem harus dapat <b>mengakses database akademik</b> Mahasiswa (nilai, IPK, status akademik).	Mahasiswa
FR-08	Sistem harus dapat menjawab pertanyaan spesifik Mahasiswa terkait <b>prosedur Layanan Akademik</b> (KRS, pembayaran kuliah, beasiswa, cuti/aktif kembali).	Mahasiswa
FR-09	Sistem harus mampu menampilkan <b>nilai dan status akademik (IPK)</b> Mahasiswa secara langsung di jendela chat.	Mahasiswa
FR-10	Sistem harus menyediakan fitur untuk <b>mengunduh nilai atau transkrip</b> dalam format <b>PDF</b> secara otomatis berdasarkan data akademik Mahasiswa.	Mahasiswa

## 4. Kebutuhan Non-Fungsional (Non-Functional Requirements)

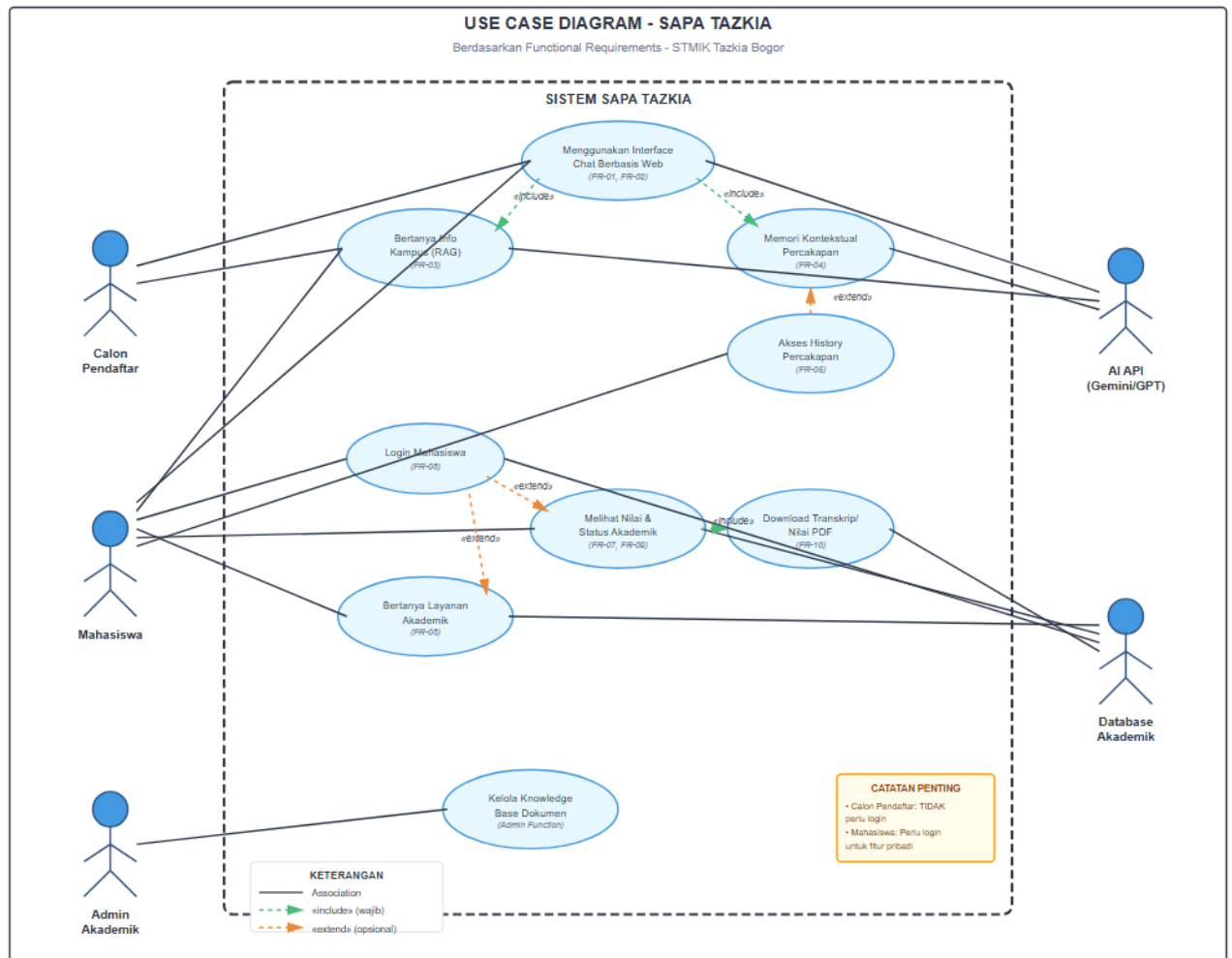
Kategori	Kebutuhan	Deskripsi
<b>Kinerja</b>	Kecepatan Respon (NFR-01)	Waktu respon chatbot harus <2 detik.
	Skalabilitas (NFR-02)	Sistem harus dapat menangani minimal <b>100 pengguna per hari</b> dan $\pm 30.000$ chat per bulan.
<b>Keamanan</b>	Autentikasi (NFR-03)	Modul <i>login</i> harus aman.
	Protokol (NFR-04)	Komunikasi harus menggunakan <b>SSL/HTTPS</b> (termasuk pada API).

<b>Kehandalan</b>	Ketersediaan (NFR-05)	Layanan harus tersedia <b>24 jam / 7 hari</b> .
<b>Teknis</b>	Infrastruktur (NFR-06)	Di-deploy pada <b>Cloud Server/VPS</b> (Vultr/Hostinger).
	Teknologi AI (NFR-07)	Menggunakan <b>Gemini Flash-8B API</b> atau <b>GPT-4o mini API</b> .
<b>Usabilitas</b>	Antarmuka (NFR-08)	Antarmuka web harus intuitif dan <b>responsif</b> ( <i>responsive design</i> ).

## 5. Keterbatasan & Batasan Proyek (Constraints & Limitations)

ID	Kategori	Deskripsi
CL-01	<b>Model AI</b>	Tidak mencakup pengembangan model AI dari nol, hanya menggunakan <i>API existing</i> .
CL-02	<b>Integrasi Eksternal</b>	Tidak mencakup integrasi dengan sistem eksternal seperti <b>PDDIKTI</b> atau Sistem Pembayaran Online.
CL-03	<b>Tipe Aplikasi</b>	Hanya dikembangkan sebagai <b>aplikasi berbasis web</b> , bukan aplikasi <i>mobile native</i> .
CL-04	<b>Kapasitas Layanan</b>	Kapasitas layanan <b>dibatasi</b> oleh kuota API (~100 pengguna/hari, ~30.000 chat/bulan).
CL-05	<b>Admin Panel</b>	Pengembangan Admin Panel untuk manajemen <i>knowledge base</i> dan monitoring <b>tidak menjadi fokus utama in-scope</b> proyek.

## **5. Use Case Diagram**



## Komponen Diagram:

### Aktor (5 aktor):

1. **Calon Mahasiswa** - mengakses info pendaftaran dan kampus
2. **Mahasiswa** - akses penuh termasuk nilai
3. **Admin/Staf Akademik** - kelola sistem
4. **Sistem AI (Gemini/GPT)** - proses AI
5. **Database Akademik** - sumber data

### Use Case (12 use case):

1. Chat dengan AI Bot (pusat)
2. Melihat Info Pendaftaran
3. Bertanya Seputar Kampus
4. Login Sistem
5. Cek Nilai & Transkrip
6. Export Transkrip PDF

- 7. Bertanya Info Akademik
- 8. Melihat Data Mahasiswa
- 9. Kelola Knowledge Base
- 10. Lihat Statistik Penggunaan
- 11. Kelola Data Pengguna
- 12. Context Memory

## Test Scenario:

### Bagian 1: Test Scenario

1.1 Test Case: Chat Interface & NLP (FR-01, FR-02)

#### TC-01: User Membuka Aplikasi Pertama Kali

Test Case ID	TC-01
Nama Test	User membuka aplikasi pertama kali
Tujuan	Memastikan interface chat dapat diakses dan menampilkan pesan sambutan
Precondition	Browser support, koneksi internet stabil
Test	URL: <a href="https://sapatazkia.ac.id">https://sapatazkia.ac.id</a>

**Postcondition:** User siap untuk mengetik pertanyaan

#### TC-02: User siap untuk mengetik pertanyaan

Test Case ID	TC-02
Nama Test	User mengirim pertanyaan sederhana
Tujuan	Memastikan sistem dapat memproses dan merespons pertanyaan
Precondition	Interface chat sudah terbuka (TC-01 berhasil)
Test	Input:"Halo"

**Postcondition:** User dapat melanjutkan chat

#### TC-03: User siap Mengirim Input Kosong

Test Case ID	TC-03
--------------	-------

Nama Test	Validasi input kosong
Tujuan	Memastikan sistem tidak memproses input kosong
Precondition	Interface chat sudah terbuka
Test	Input: "" (kosong)

**Postcondition: Sistem tidak mengirim request ke API**

## 1.2 Test Case: RAG - Info Kampus (FR-03)

### TC-04: Pertanyaan Tentang Program Studi

Test Case ID	TC-04
Nama Test	Pertanyaan tentang program studi
Tujuan	Memastikan RAG dapat retrieve informasi dari knowledge base
Precondition	knowledge base sudah terisi dengan dokumen program studi
Test	Input: "Apa saja program studi di Tazkia?"

**Postcondition: User mendapat informasi akurat**

### TC-05: Pertanyaan di Luar Knowledge Base

Test Case ID	TC-05
Nama Test	Pertanyaan tidak ada di knowledge base
Tujuan	Memastikan sistem tidak memberikan jawaban yang salah (hallucination)
Precondition	Knowledge base tidak memiliki informasi tentang topik tertentu
Test	Input: "Berapa gaji lulusan Tazkia?" (data tidak ada)

**Postcondition: User memahami limitasi sistem**

## 1.3 Test Case: Context Memory (FR-04)

### TC-06: Follow-up Question dengan Context

Test Case ID	TC-06
Nama Test	Follow-up question dengan referensi context

Tujuan	Memastikan sistem dapat mengingat context percakapan
Precondition	user sudah bertanya minimal 1 pertanyaan sebelumnya
Test	Pertanyaan 1:"Apa saja program studi di Tazkia?" Pertanyaan 2:"Berapa biaya kuliahnya?"

**Postcondition: Percakapan mengalir natural dengan context**

1.4 Test Case: Login Mahasiswa (FR-06)

**TC-07: Login dengan Kredensial Valid**

Test Case ID	TC-07
Nama Test	Login dengan NIM dan password yang benar
Tujuan	Memastikan mahasiswa dapat login ke sistem
Precondition	Mahasiswa sudah memiliki akun di database
Test	NIM: 2021010001, Password: Test123

**Postcondition: Mahasiswa terautentikasi dan dapat akses fitur mahasiswa**

**TC-08: Login dengan Kredensial Salah**

Test Case ID	TC-08
Nama Test	Login dengan password salah
Tujuan	Memastikan sistem menolak kredensial salah
Precondition	Form login terbuka
Test	NIM: 2021010001, Password: wrongpass(salah)

**Postcondition: Akun aman dari brute force attack**

1.5 Test Case: Melihat Nilai (FR-09)

**TC-09: Mahasiswa Melihat Nilai via Chat**

Test Case ID	TC-07
Nama Test	Mahasiswa bertanya nilai via chat
Tujuan	Memastikan mahasiswa dapat nilai mereka
Precondition	Mahasiswa sudah login (TC-07 berhasil)
Test	Input:"Tampilkan nilai saya semester ini"

**Postcondition: Mahasiswa dapat melihat nilai mereka**

#### **TC-10: User Tidak Login Mencoba Akses Nilai**

Test Case ID	TC-10
Nama Test	User yang tidak login mencoba akses nilai
Tujuan	Memastikan fitur nilai hanya untuk mahasiswa yang login
Precondition	user TIDAK login
Test	Input:"Tampilkan nilai saya"

**Postcondition: Mahasiswa harus login terlebih dahulu untuk melihat nilai**

### 1.6 Test Case: Download PDF (FR-10)

#### **TC-07: Download Transkrip Nilai**

Test Case ID	TC-11
Nama Test	Download transkrip nilai dalam format PDF
Tujuan	Memastikan mahasiswa dapat download transkrip
Precondition	Mahasiswa sudah login dan melihat nilai (TC-09 berhasil)
Test	NIM: 2021010001

**Postcondition: Mahasiswa memiliki transkrip PDF**

### 1.7 Test Case: Performance & Load (NFR-01, NFR-02)

#### **TC-12: Response Time Test**

Test Case ID	TC-12
--------------	-------

Nama Test	Mengukur waktu response chatbot
Tujuan	Memastikan response time <2 detik
Precondition	Sistem dalam kondisi normal load
Test	10 pertanyaan standar

**Postcondition: Performance memenuhi NFR-01**

#### **TC-13: Concurrent Users Test**

Test Case ID	TC-13
Nama Test	Test dengan 100 user concurrent
Tujuan	Memastikan sistem dapat handle 100 user/hari
Precondition	Load testing tools ready (JMeter/Locust)
Test	100 virtual users

**Postcondition: Sistem scalable sesuai NFR-02**

#### 1.8 Test Case: Security (NFR-03, NFR-04)

##### **TC-14: HTTPS Connection Test**

Test Case ID	TC-14
Nama Test	Verifikasi koneksi menggunakan HTTPS
Tujuan	Memastikan komunikasi terenskripsi
Precondition	SSL certificate sudah installed
Test	URL aplikasi

**Postcondition: Komunikasi aman sesuai NFR-04**

**Vector database Purpose store document embedding for RAG**