

Technology Stack - Sapa Tazkia

1. Overview

Sapa Tazkia adalah sistem chatbot akademik berbasis web yang dikembangkan untuk mahasiswa STMIK Tazkia. Aplikasi ini memungkinkan mahasiswa berinteraksi melalui chatbot untuk mengakses data akademik seperti IPK, nilai semester, jadwal kuliah, serta riwayat akademik. Sistem ini juga menyediakan fitur ekspor data ke PDF.

2. Frontend Stack

Komponen	Teknologi	Keterangan
Framework UI	React.js 18.2.0	Library JavaScript untuk membangun antarmuka pengguna berbasis komponen yang reactive dan reusable
CSS Framework	Tailwind CSS 3.4.1	Utility-first CSS framework untuk membuat tampilan responsif dan modern dengan cepat
State Management	Redux / Zustand	Mengelola state global aplikasi seperti data user, session, dan chat history antar komponen
HTTP Client	Axios	Library untuk melakukan HTTP requests ke backend API dengan interceptor support
UI Component Library	shadcn/ui	Komponen UI yang customizable dan accessible untuk mempercepat development
Routing	React Router v6	Client-side routing untuk navigasi multi-page tanpa reload halaman

3. Backend Stack

Komponen	Teknologi	Keterangan
Bahasa	Node.js 20.12.2	JavaScript runtime environment untuk menjalankan aplikasi backend
Framework	Express.js 4.18.2	Web framework minimal dan flexible untuk menyediakan struktur routing dan middleware
Authentication	JWT (JSON Web Token)	Token-based authentication untuk sistem autentikasi pengguna yang stateless dan secure
API Architecture	RESTful API	Arsitektur API standar untuk komunikasi antara frontend dan backend
Password Hashing	Bcrypt	Library untuk hashing password dengan salt untuk keamanan maksimal
PDF Generator	PDFKit / Puppeteer	Library untuk generate dokumen PDF seperti transkrip dan laporan akademik

4. Database Stack

Komponen	Teknologi	Keterangan
Database Utama	MySQL 8.0	Relational database untuk menyimpan data pengguna, percakapan, nilai, jadwal, dan log aktivitas
ORM	Prisma 5.x	Modern ORM untuk menghubungkan backend dengan database secara efisien dengan type-safe queries
Database Migration	Prisma Migrate	Tool untuk mengelola schema database dan migration secara versioned
Caching	Redis (Opsional)	In-memory data store untuk mempercepat permintaan data yang sering digunakan dan mengurangi load database
Connection Pooling	Prisma Connection Pool	Built-in connection pooling untuk mengoptimalkan koneksi database

5. DevOps & Infrastruktur

Komponen	Teknologi	Keterangan
Hosting Frontend	Vercel	Platform hosting untuk deploy React aplikasi dengan auto-scaling dan CDN global
Hosting Backend	Railway / Render	Platform hosting untuk menjalankan Node.js backend dengan database support
CI/CD	GitHub Actions	Automated pipeline untuk testing, building, dan deployment otomatis
Containerization	Docker	Container platform untuk menyederhanakan proses pengembangan dan deployment dengan environment yang konsisten
Container Orchestration	Docker Compose	Tool untuk mendefinisikan dan menjalankan multi-container Docker applications
Version Control	Git + GitHub	Distributed version control system untuk manajemen kode sumber dan kolaborasi tim
SSL/TLS	Let's Encrypt	Free SSL certificate untuk HTTPS encryption

6. Security & Monitoring

Komponen	Teknologi	Keterangan
HTTP Security	Helmet.js	Middleware Express untuk mengamankan aplikasi dengan setting HTTP headers yang proper
HTTPS Protocol	TLS 1.3	Enkripsi komunikasi end-to-end antara client dan server
Rate Limiting	express-rate-limit	Middleware untuk membatasi jumlah requests dan mencegah brute force attacks
Logging	Winston	Production-grade logging library untuk merekam aktivitas server, errors, dan audit trail
Error Tracking	Sentry (Opsional)	Real-time error tracking dan monitoring untuk debugging

Monitoring	Grafana + Prometheus	Visualisasi metrics dan pemantauan performa aplikasi secara real-time
------------	----------------------	---

7. Struktur Folder

```
/sapa-tazkia |
├── frontend/      # Source code React.js
│   ├── backend/   # Source code Express.js
│   ├── database/  # Skema dan migrasi MySQL
│   ├── docs/      # Dokumentasi proyek
│   ├── .env       # Konfigurasi variabel lingkungan
│   └── docker-compose.yml # Pengaturan kontainer
```

8. Versi & Dependensi

Technology Versions

- **Node.js:** 20.12.2 LTS
 - **React.js:** 18.2.0
 - **Tailwind CSS:** 3.4.1
 - **Express.js:** 4.18.2
 - **MySQL:** 8.0.35
 - **Prisma:** 5.x
 - **Redis:** 7.2 (optional)
 - **Docker:** 24.0+
 - **Docker Compose:** 2.20+
-

9. Alasan Pemilihan Teknologi

Teknologi dipilih berdasarkan kemudahan integrasi dan ketersediaan ekosistem. React dan Tailwind memberikan tampilan modern, sementara Express dan MySQL menawarkan kestabilan tinggi dalam pembuatan RESTful API. Penggunaan Prisma mempercepat pengembangan dengan model ORM yang efisien.