

TixGo

Identity Service & Attendance Integration Portal



TixGo merupakan sistem terintegrasi untuk identity management dan portal operasional attendance yang terhubung dengan Ratu Attendance Service. Sistem ini dikembangkan sebagai submission UAS II3160 Teknologi Sistem Terintegrasi. Implementasi TixGo terdiri dari dua bagian utama: Identity Service sebagai layanan autentikasi (Tugas 2) dan Identity Portal yang mengintegrasikan proses attendance/check-in ke Ratu (Tugas 3).

Pada Tugas 2, Identity Service berperan sebagai layanan autentikasi inti yang menyediakan fitur registrasi pengguna, login, serta validasi token menggunakan JWT (HS256). Sistem menerapkan role-based access control dengan role seperti admin, committee, dan user. Password pengguna tidak disimpan dalam bentuk plaintext karena diproteksi menggunakan bcrypt hashing.

Pada Tugas 3, dikembangkan Identity Portal (portal admin/panitia) yang menjadi antarmuka untuk mengakses fitur autentikasi sekaligus melakukan integrasi

terhadap layanan attendance dari Ratu. Portal ini menyediakan fitur seperti dashboard attendance overview, verifikasi check-in, dan statistik attendance. Dari sisi backend, Portal melakukan proxy/forward request autentikasi ke Identity Service (untuk endpoint auth) dan meneruskan request attendance/check-in ke Ratu Attendance Service menggunakan token JWT yang valid.

Fitur Utama Sistem

Fitur utama yang diimplementasikan pada TixGo meliputi:

1. User Authentication berbasis JWT (HS256) untuk login dan akses endpoint yang dilindungi.
2. Role-based access control (admin, committee, user) untuk pengaturan hak akses.
3. Integrasi attendance dengan Ratu Attendance Service, termasuk pengambilan data attendance dan pembuatan check-in.
4. Real-time check-in management melalui endpoint Portal yang terhubung ke Ratu.
5. Arsitektur berbasis microservices (Identity Service dan Portal Service terpisah).
6. Dukungan deployment produksi menggunakan Docker dan Docker Compose.

Akses Layanan

Sistem dapat diakses melalui browser dan terminal.

Akses Browser (direkomendasikan)

Portal utama dapat diakses secara publik melalui HTTPS pada:

- Portal (Identity + Attendance Integration): <https://dina.theokaitou.my.id>
Portal ini digunakan untuk registrasi, login, melihat dashboard attendance, verifikasi check-in, dan statistik attendance.

Sementara itu, layanan partner:

- Ratu Attendance Service: <https://ratu.theokaitou.my.id>
Ratu menyediakan fitur event management, attendance tracking, dan mekanisme check-in.

Untuk kebutuhan pengujian, tersedia kredensial default:

- Username: admin
- Password: admin123
- Role: admin

Pengguna juga dapat membuat akun baru melalui menu Register pada halaman login.

Akses Terminal / API Testing

Pada environment development, sistem dijalankan pada:

- Identity Service: <http://localhost:18080>
- Portal Service: <http://localhost:18081>
- (Opsional) Ratu lokal: <http://localhost:18082>

Untuk environment produksi di STB, akses dilakukan melalui SSH menggunakan Cloudflare tunnel, kemudian pengujian dapat dilakukan dari STB menggunakan endpoint health check dan endpoint API terkait.

Quick Start Development

Untuk menjalankan sistem pada development environment, dilakukan langkah berikut:

1. Clone repository dari GitHub.
2. Menjalankan semua service (identity + portal + database) menggunakan Docker Compose.
3. Memastikan container berjalan melalui perintah docker-compose ps.
4. Melakukan health check melalui endpoint /health pada masing-masing service.

Contoh API yang Digunakan

Identity Service menyediakan endpoint utama:

- Health check untuk memastikan service aktif.
- Register untuk membuat user baru (username, password, role).
- Login untuk memperoleh JWT token.
- /me sebagai endpoint protected untuk mengambil informasi user berdasarkan token.

Portal Service menyediakan endpoint integrasi:

- Endpoint auth seperti /api/auth/register, /api/auth/login, dan /api/auth/me yang diproxy ke Identity Service.
- Endpoint integrasi attendance seperti /api/attendance/{event_id} untuk mengambil data attendance dari Ratu.
- Endpoint /api/checkins untuk membuat check-in pada Ratu, serta endpoint get check-ins yang mendukung fallback jika fitur belum tersedia di API Ratu.

Arsitektur Sistem

Arsitektur TixGo menggunakan pendekatan microservices yang terdiri dari:

1. Portal Service (FastAPI) sebagai entry point pengguna (UI dan API gateway).

2. Identity Service (FastAPI) sebagai layanan autentikasi JWT dan manajemen user.
3. PostgreSQL untuk penyimpanan data user/credential dan data terkait.
4. Nginx reverse proxy sebagai gateway internal pada deployment produksi.
5. Cloudflare Tunnel untuk routing domain publik menuju service di STB.
6. Ratu Attendance Service sebagai external service yang menyediakan API attendance.

Alur singkatnya: client mengakses Portal via domain publik : Cloudflare tunnel meneruskan ke Nginx/Portal → Portal memvalidasi token ke Identity Service dan melakukan request attendance/check-in ke Ratu sesuai kebutuhan.

Keamanan Sistem

Keamanan sistem menekankan beberapa aspek:

- Autentikasi menggunakan JWT HS256 dengan masa berlaku token 60 menit.
- Password disimpan menggunakan bcrypt hashing (salt rounds 12).
- Endpoint protected menggunakan Authorization: Bearer token.
- Secret dan credential disimpan pada environment variable (tidak di-commit ke git).
- Komunikasi eksternal menggunakan HTTPS/TLS, dan komunikasi internal berada dalam jaringan Docker yang terisolasi.
- Error handling dilakukan dengan aman untuk menghindari kebocoran informasi sensitif.

Deployment Produksi di STB

Deployment produksi dilakukan pada Set Top Box (STB) dengan langkah umum:

- 1.** Login ke STB via SSH menggunakan Cloudflare tunnel.
- 2.** Clone repository pada direktori /opt.
- 3.** Menyiapkan file environment produksi (.env.production) berisi secret JWT, base URL internal, dan konfigurasi database.
- 4.** Build dan menjalankan service menggunakan docker-compose -f docker-compose.prod.yml up -d.
- 5.** Verifikasi status container dan melakukan test endpoint publik melalui domain dina.theokaitou.my.id.

Monitoring dilakukan melalui log Docker Compose untuk portal, identity, dan database.