BAB 5
SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan perancangan yang telah dilakukan, API dengan metode keamanan RFC 6238 dan RFC 7617 yang diimplementasikan untuk simulasi sistem presensi berhasil dibangun dengan baik. Teknologi utama yang digunakan untuk membangun aplikasi ini adalah dengan bahasa pemrograman TypeScript, serta *minimalist framework* yakni Express.js untuk API dan Next.js untuk implementasi *front-end*. Sistem basis data yang digunakan ialah MariaDB/MySQL untuk sistem basis data utama, serta Redis sebagai *cache* dan *queue processing*. Pengiriman OTP dilakukan dengan media surat elektronik beserta aplikasi Authenticator. Keseluruhan sistem dilakukan *deployment* pada lingkungan produksi, lengkap dengan SSL/TLS yang berasal dari *Certificate Authority* yaitu Let's Encrypt, HTTP/2, beserta dengan nama domain 'https://www.fumi-no.com/'. *Deployment* aplikasi berada pada sebuah sistem yang memiliki sistem operasi Linux dan dijalankan secara aman dengan sebuah *firewall*, *least privileged users*, *continuous backups with cronjobs*, serta Nginx untuk melakukan *traffic control*.

Aplikasi yang berbasis API dan *single-page application* ini sudah dapat mengirim dan melakukan verifikasi *one-time password* secara baik, beserta dengan fitur-fitur pendukung seperti melakukan presensi, melihat data akun, melakukan perubahan profil, menghapus akun, melakukan operasi *create*, *read*, *update*, *delete* untuk administrator, pengiriman surat elektronik ketika gagal tiga kali mengisi OTP yang benar, dan pengiriman surat elektronik untuk mengingatkan pengguna untuk melakukan *check-out*.

Dari segi performa, aplikasi sudah diukur dengan sebuah *tool* yaitu Lighthouse. Berdasarkan hasil pengukuran yang dilakukan, aplikasi memiliki tingkat performa yang sangat baik apabila diukur berdasarkan Lighthouse. Metrik yang diukur ialah *performance*, *accessiblity*, *best practices*, *search engine optimization*, serta *progressive web application*. Skor yang didapatkan ialah 100, 97, 100, dan 100 untuk masing-masing metrik yang diukur (kecuali *progressive web application*) untuk kondisi *desktop*.

Keseluruhan aplikasi sudah dilakukan evaluasi dengan metode OWASP. Metrik yang digunakan adalah *Authentication Testing*, *Authorization Testing*, *API*

Security, *OTP Security*, *Session Management*, dan *Random Generator Security*, yang semuanya mendapatkan hasil bahwa API *back-end* sudah sesuai dengan kriteria keamanan yang ditetapkan oleh OWASP dan mendapatkan hasil yaitu Sangat Aman dan tidak terdapat *vulnerability* apapun dalam aplikasi ini. Aplikasi juga sudah dilakukan *automated penetration testing* dengan menggunakan OWASP ZAP, dan hasilnya tidak ada *vulnerability* pada aplikasi.

Sebagai evaluasi keamanan dari pihak *developer*, aplikasi juga dilakukan evaluasi dengan metode Snyk untuk melakukan *static code analysis* dan *dependencies security checking*. Tidak ada *vulnerability* yang berarti pada *dependencies* yang digunakan, serta *code quality* yang ditulis sudah berada dalam tingkat yang sangat tinggi. Berdasarkan analisis keamanan dan pendekatan empiris, dapat dikatakan pula bahwa metode RFC 6238 (untuk generasi OTP) dan RFC 7617 (untuk pengiriman OTP) dapat digunakan untuk sistem autentikasi dan otorisasi menggunakan TOTP secara aman dan praktis, tanpa mengorbankan *usability*.

Untuk melengkapi penelitian, penerapan API yang digunakan untuk simulasi sistem presensi juga dilakukan evaluasi dengan metode *Technology Acceptance Model*. Adapun metrik yang diukur adalah *perceived ease of use*, *perceived usefulness*, serta rasio *security and usability*. Terdapat 59 orang yang sudah melakukan evaluasi dengan cara pengisian kuesioner dan pengujian secara manual. Hasil evaluasi dari kuesioner dihitung menggunakan Skala Likert dan mendapatkan hasil yaitu tingkat penerimaan keseluruhan sebesar 91.81% (*ease of use* sebesar 94.37% *usefulness* sebesar 89.63% dan *security and usability* sebesar 94.23%), atau dapat dikatakan bahwa pengguna sangat setuju apabila aplikasi yang dibangun dijadikan sebagai alternatif untuk sistem presensi biometrik atau konvensional, terlebih dalam kondisi pandemi Covid-19 saat Skripsi ini ditulis karena beberapa alasan, seperti praktis, *integrity* yang terjaga, aman, mudah untuk digunakan, dan alasan-alasan lainnya. Selain itu, dalam penelitian ini, timbul bukti lain bahwa API juga terbukti merupakan alternatif yang *feasible* untuk menggantikan sistem yang biasanya dibangun. Metode keamanan yang diteliti pada penelitian ini juga dapat diterapkan pada API lain yang mungkin membutuhkannya sebagai sistem autentikasi dua faktor.

5.2 Saran

5.2.1 Saran Akademis

1. Menggunakan algoritme lain untuk OTP, seperti algoritme OCRA (untuk generasi) dan Digest Authentication (untuk pengiriman).
2. Mencoba *implementation detail* yang lain seperti MongoDB, PostgreSQL, KeyDB, *mail server* lain, *web server* lain, dan sebagainya untuk mendapatkan perbandingan yang kuantitatif terhadap teknologi yang digunakan.
3. Melakukan evaluasi tingkat penerimaan pada institusi sesungguhnya untuk menerapkan simulasi yang sudah dilakukan pada penelitian ini yaitu sistem presensi. Sistem sudah terbukti aman dan memiliki level *usability* yang baik berdasarkan evaluasi yang telah dilakukan.

5.2.2 Saran Praktis

1. Memisahkan / melakukan ekstraksi pada API untuk menjadi *Identity Provider* yang dapat digunakan oleh siapapun. *Identity Provider* yang dimaksud adalah pembuatan *Single Sign On* yang sudah lengkap dengan MFA.
2. Implementasi protokol autentikasi dan otorisasi yang membuat API dapat diakses secara eksternal, seperti OAuth2, OpenID Connect, atau SAML.
3. Implementasi sistem *refresh tokens* pada *special session* yang digunakan apabila sudah terautentikasi dengan metode RFC 6238 dan RFC 7617.
4. Mengimplementasikan ABAC (*Attribute Based Access Control*) untuk dan perkembangan *Identity Access Management* (IAM) pada API.
5. Implementasi metode pengiriman OTP dengan tambahan metode lain, seperti WhatsApp, Telegram, Signal, SMS, dan lain sebagainya.
6. Mengembangkan aplikasi untuk menjadi sebuah *progressive web application*, agar mempermudah pengguna untuk melakukan presensi karena aplikasi dapat dipasang pada *home screen* serta memiliki kemiripan seperti sebuah *native application*, baik dalam lingkungan *desktop* maupun *mobile*.