BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Application Programming Interface (API) adalah sebuah antarmuka yang digunakan oleh pengguna untuk mengakses atau menggunakan sebuah layanan dalam jaringan Internet. Terkadang, ada sejumlah *endpoints* dalam API yang hanya dikhususkan untuk pengguna dengan tingkat otorisasi tertentu. Contohnya, API *endpoint* yang memberikan hak akses ke dalam data yang semestinya hanya bisa diakses oleh pengguna dengan tingkat peran tertentu pada sistem dan *endpoint* untuk melakukan aktivitas tertentu yang tidak bisa diakses oleh sembarang pengguna. Untuk mengamankan *endpoints* yang tersedia pada API, salah satu metode yang dapat digunakan adalah RFC 7617 yang dikenal pula dengan nama Basic Authentication dapat digunakan pada protokol *HyperText Transfer Protocol* (HTTP). Metode autentikasi tersebut dapat dilakukan pengembangan untuk mengirimkan *credentials* yang sudah di-*encode* kepada penyedia layanan melalui cara protokol *HyperText Transfer Protocol Secure* (HTTPS). Menggunakan metode tersebut tanpa menggunakan protokol HTTPS umumnya tidak cukup untuk menjaga keamanan API [1].

Manfaat dalam menggunakan sistem API adalah menambahkan abstraksi pada operasi dalam sistem yang digunakan oleh pengguna. Selain menambahkan abstraksi, penggunaan API juga tergolong lebih aman karena tidak mengekspos atau membuka sistem ke ranah publik. Penggunaan API juga membuat aliran data dari *client* ke *server* menjadi lebih terkontrol, karena data dapat diatur terlebih dahulu oleh API yang merupakan *middleware* sebelum diproses oleh sistem [2], baik itu dalam sistem basis data atau sistem pada umumnya. Kelebihan lainnya yang didapatkan ketika menggunakan API adalah dapat dilakukan konfigurasi sedemikian rupa sehingga hanya menerima permintaan dari sumber yang terpercaya saja (peraturan *cross origin resource sharing* atau CORS), menerima *request* yang memiliki metode HTTP tertentu, melakukan *parsing headers*, *parsing request body*, melakukan *preprocessing*, dan lain sebagainya.

Penelitian sebelumnya mengungkapkan bahwa API adalah *technology agnostic* dan merupakan contoh nyata dari *separation of concerns* [3]. Dengan konsep ini, migrasi dapat dilakukan secara mudah kepada teknologi lainnya, dan

pengguna atau pengembang umumnya tidak perlu terlalu memikirkan hal-hal teknis yang kompleks apabila ingin melakukan migrasi sistem. Zaman sekarang, sudah banyak instansi yang melindungi dan mengakses data sensitif dengan sistem API, bukan dengan akses sistem basis data secara langsung seperti yang umumnya dilakukan sebelumnya [4].

Pada tahun 2021, Salt Security mengungkapkan bahwa tingkat serangan yang menargetkan API meningkat sebanyak 681%. Perkembangan API dari Desember 2020 hingga Desember 2021 juga menunjukkan peningkatkan yang sangat drastis, yang memberikan fakta bahwa semakin banyak pengguna yang menggunakan API sebagai dasar dari sistem yang dibuat. Dua alasan utama alasan penggunaan API adalah *development efficiencies* (58%) dan *platform integrations* (44%) berdasarkan data yang diberikan oleh Salt Security. Data dari sumber yang sama juga mengungkapkan bahwa 95% pengguna API pernah mengalami *security incident* dalam 12 bulan terakhir. Tiga buah *security vulnerabilities* yang umumnya dialami oleh para responden ialah *vulnerability* (39%), *authentication problem* (32%), dan *sensitive data exposure* (30%) [5].

Pemegang sistem wajib melihat lebih dari sistem autentikasi nama pengguna dan kata sandi untuk melindungi sistem dan data yang ada. Salah satu cara yang dapat digunakan ialah menggunakan kata sandi sekali pakai yang dikirim kepada pengguna sebagai bentuk autentikasi multi-faktor atau autentikasi dua-faktor (MFA/2FA) [6]. RFC 6238, atau yang lebih dikenal sebagai kata sandi sekali pakai berbasis waktu yang memiliki singkatan TOTP atau *Time-Based One Time Password* merupakan pengembangan dari kata sandi sekali pakai berbasis HMAC atau HOTP (*HMAC-Based One Time Password*) yang menghasilkan kata sandi sekali pakai berdasarkan waktu pada saat kata sandi dihasilkan. Dengan metode ini, kelemahan sebuah sistem dapat menjadi semakin kecil karena pemanfaatan variabel waktu untuk menghasilkan kata sandi yang unik untuk setiap momen [7]. Dua buah keuntungan dari algoritme adalah bersifat *stateless* dan *zero knowledge*, dalam artian *server* tidak perlu menyiapkan *state* untuk menyimpan data *one-time password* (OTP) yang sedang diproses. Dengan adanya algoritme ini, terdapat sebuah potensi kombinasi algoritme, yaitu RFC 6238 (untuk generasi) dan RFC 7617 (untuk pengiriman) untuk mengamankan *endpoints* yang terdapat pada API.

Salah satu implementasi yang dapat digunakan oleh kata sandi sekali pakai berbasis waktu adalah untuk autentikasi sistem presensi. Sistem presensi merupakan sebuah sistem yang digunakan untuk menentukan apakah seseorang melakukan presensi sesuai dengan jadwal yang ditentukan. Konsep presensi

merupakan hal yang esensial yang terdapat dalam kehidupan sosial, misalnya seperti presensi di sekolah, presensi di kampus, presensi di kantor, presensi pada *event*, bioskop, dan lain sebagainya. Pada praktiknya, sistem presensi dapat menggunakan beberapa macam teknik, seperti presensi manual, presensi menggunakan pemindai sidik jari, presensi menggunakan kartu presensi, dan lain sebagainya. Sistem yang sering digunakan adalah sistem presensi biometrik dengan menggunakan pemindai sidik jari, dan digunakan untuk mengurangi tingkat kepalsuan yang mungkin dihadapi ketika menggunakan sistem presensi konvensional. Kata sandi sekali pakai berbasis waktu dapat dianggap sebagai *something you have* dan dapat digunakan menurut penelitian dari Matyas [8].

Sistem biometrik dengan menggunakan pemindai sidik jari memang efisien karena sifatnya yang cepat dan ringkas, tetapi sistem memiliki kelemahan, yaitu memiliki kemungkinan untuk mendapatkan *false positives* maupun *false negatives* yang mencapai hingga 5% [9]. Kelemahan lainnya yang ada adalah dapat dipermainkan dengan menggunakan jari palsu, menimbulkan *false negative* bila sidik jari terkena bahan kimia (umumnya terjadi pada orang yang bekerja dengan kimia), dan mesin pemindai dapat tidak mendeteksi sidik jari apabila sidik jari luka [10]. Hal ini tidak diinginkan oleh pengguna sistem. Belum lagi, dalam kondisi pandemi Covid-19 mengakibatkan perlunya melakukan *social distancing* yang tidak mampu dilakukan ketika menggunakan sistem pemindai sidik jari, karena harus bersentuhan dengan mesin tersebut. Salah satu cara yang dapat dikatakan aman sebagai alternatif dari sistem biometrik tersebut adalah dengan menggunakan sebuah informasi yang merupakan kombinasi *something you know* dan *something you have*, dalam hal ini salah satu implementasinya adalah kata sandi sekali pakai berbasis waktu. Dengan menggunakan *one-time password*, tidak perlu dilakukan kontak fisik dengan mesin maupun orang lain. Berbekal fakta dari sini, penelitian akan menerapkan API pada sebuah simulasi sistem presensi untuk sekaligus mengetahui alternatif dan *feasibility* dari sistem presensi yang sudah ada.

Dalam penelitian terdahulu, pernah dibuat sebuah sistem implementasi konsep *one-time password* pada sistem *login* [11] [12] dan untuk pengisian kartu rencana studi [13]. Ketiga penelitian ini memiliki kesimpulan penelitian yaitu implementasi yang berhasil dan terlihat bahwa sistem menjadi lebih aman karena diperlukan dua kali sistem autentikasi [11]. Adapun penelitian lain yang dibuat adalah dengan melakukan integrasi sistem telepon genggam ke dalam aplikasinya. Hasil dari penelitian menggunakan *user acceptance testing* berkesimpulan bahwa pengguna merasa lebih aman saat menggunakan autentikasi dua faktor [12].

Penelitian sebelumnya mendapatkan hasil bahwa konsep *something you have* dapat dijadikan sebagai alternatif untuk autentikasi biometrik, dimana metode yang digunakan adalah menggunakan kartu *radio-frequency identification* (RFID) dan teknologi *bluetooth* [14]. Dua buah penelitian oleh Kapila Purohit dan Anurag Rana mengungkapkan bahwa penggunaan kata sandi sekali pakai dapat mengurangi risiko kebocoran data dan meningkatkan keamanan sistem secara menyeluruh [15] [16]. Sistem yang dibangun oleh Kapila Purohit dan Anurag Rana belum menggunakan algoritme RFC 6238.

Berdasarkan sumber-sumber yang diamati sebelumnya, belum pernah ada penelitian terdahulu yang membahas secara khusus untuk pembuatan API dengan konsep ini. Adapun perancangan terdahulu secara khusus belum pernah ada yang mengirimkan *one-time password* dengan metode RFC 7617 ataupun melakukan kombinasi algoritme RFC 6238 dan RFC 7617. Kebanyakan sistem berbasis *one-time password* yang sudah ada umumnya bersifat komersial dan tidak bisa dipelajari bagaimana ia bekerja, mulai dari metode, algoritme cara untuk mendapatkan *one-time password* tersebut, tahap implementasinya, dan tingkat keamanan serta penerimaannya.

Penelitian ini dilakukan dengan cara membuat sebuah aplikasi berbasis web, beserta sebuah API. Arsitektur dari API menggunakan tipe *representational state transfer* (REST) [17]. Ketika seorang pengguna yang ter-autentikasi berhasil masuk ke dalam sistem dan ingin melakukan presensi atau aktivitas yang dianggap sensitif, pengguna tersebut akan dikirim sebuah kata sandi sekali pakai berbasis waktu dengan metode generasi RFC 6238. Pengguna yang berhasil memasukkan kata sandi sekali pakai berbasis waktu dengan benar akan diperbolehkan untuk mengakses data dan melakukan operasi terhadap data yang ada di dalam sistem (dengan perantara API dan tidak langsung pada sistem basis data). Informasi pengguna yang berhasil mengakses *endpoint* akan disimpan selama kurang lebih 15 menit. Pada saat itu, pengguna dapat mengakses *endpoint* yang dilindungi tersebut tanpa perlu memasukkan dan/atau mendapatkan *one-time password*.

Luaran dari penelitian ini adalah sebuah aplikasi berbasis web dan sebuah REST API yang berfungsi untuk menjalankan simulasi sistem presensi. Evaluasi penelitian akan dibagi menjadi dua, yaitu evaluasi keamanan dan evaluasi pengguna. Evaluasi keamanan akan menggunakan *framework* yang bernama *Open Web Application Security Project* (OWASP), yaitu merupakan standar yang sudah diakui oleh industri perangkat lunak, dengan mengambil metrik *Authorization Testing*, *Authentication Testing*, *API Security*, *OTP Security*, dan *Session Security*

beserta *framework* Snyk untuk *safe coding practices*. Evaluasi keamanan akan mengamankan sistem secara keseluruhan. Evaluasi pengguna dilakukan dengan cara menyebarkan kuesioner dan melakukan ujicoba aplikasi simulasi sistem presensi yang sudah dibuat. Hasil kuesioner tersebut akan dihitung dengan menggunakan *Technology Acceptance Model* (TAM) [18]. Dengan adanya API yang menerapkan metode keamanan algoritme RFC 6238 dan RFC 7617, sebuah sistem akan menjadi lebih aman, lebih fleksibel, dan lebih terpercaya untuk digunakan oleh masyarakat umum.

1.2 Rumusan Masalah

Rumusan permasalahan dari penelitian adalah sebagai berikut:

1. Bagaimana menerapkan REST API dan aplikasi untuk presensi yang memiliki sistem keamanan yang sesuai dengan RFC 6238 dan RFC 7617?
2. Bagaimana tingkat penerimaan dari keseluruhan sistem yang sudah dibangun dengan menggunakan *Technology Acceptance Model* (TAM)?
3. Bagaimana tingkat keamanan dari keseluruhan sistem yang sudah dibuat dengan menggunakan metrik *Open Web Application Security Project* (OWASP)?

1.3 Batasan Permasalahan

Batasan masalah dari penelitian adalah sebagai berikut:

1. Penelitian tidak akan membahas perbandingan keamanan RFC 6238 dengan implementasi algoritme *one-time password* lainnya.
2. Penelitian tidak akan membahas perbandingan dari detail implementasi yang dilakukan untuk mengerjakan penelitian ini.
3. Penelitian tidak akan membahas pencegahan serangan siber karena faktor *social engineering* atau faktor *malicious user* yang secara sengaja memberikan kredensial miliknya.
4. API hanya akan memiliki arsitektur berbasis *representational state transfer* (REST) dan bukan arsitektur lain seperti GraphQL, SOAP, gRPC, dan yang lain sebagainya.

5. API hanya akan menerima *request* dan memberikan *response* dalam bentuk JSON dan tidak menerima tipe data gambar, XML, BLOB, dan data dengan tipe konten lainnya.
6. API hanya akan menerapkan metode RFC 6238 dan RFC 7617 pada *endpoints* tertentu saja (yang merupakan inti, misalnya saat mengakses data sensitif atau melakukan presensi) dan tidak semua *endpoints* akan memerlukan autentikasi dengan metode tersebut.
7. Aplikasi hanya akan menunjukkan implementasi penelitian pada simulasi sistem presensi karyawan reguler yang hanya dapat melakukan presensi satu kali dalam satu hari.
8. Penelitian ini diterapkan untuk menunjukkan standar penerapan metode-metode keamanan yang diterapkan pada keseluruhan sistem terhadap standar OWASP.

1.4 Tujuan Penelitian

Tujuan dari penelitian adalah sebagai berikut:

1. Menerapkan REST API dan situs presensi yang memiliki sistem keamanan yang sesuai dengan RFC 6238 dan RFC 7617.
2. Mengetahui tingkat penerimaan dari keseluruhan sistem yang sudah dibuat dengan menggunakan *Technology Acceptance Model* (TAM).
3. Mengetahui tingkat keamanan dari keseluruhan sistem yang sudah dibuat dengan menggunakan metrik *Open Web Application Security Project* (OWASP).

1.5 Manfaat Penelitian

Manfaat dari penelitian adalah sebagai berikut:

1. Menerapkan simulasi sistem presensi dengan metode RFC 6238 dan RFC 7617 dalam sebuah REST API yang diterapkan pada aplikasi.
2. Mengetahui tingkat penerimaan dan keamanan keseluruhan sistem.

3. Mengetahui dan mendapatkan alternatif untuk sistem presensi selain presensi biometrik yang sudah ada dan presensi manual.
4. Menjadi sebuah landasan untuk yang ingin membuat API dengan metode keamanan RFC 6238 dan RFC 7617 secara aman dengan sesuai dengan standar OWASP yang berlaku saat Skripsi ini ditulis.

1.6 Sistematika Penulisan

Sistematika penulisan laporan adalah sebagai berikut:

- Bab 1 PENDAHULUAN
  Bagian ini menjelaskan seputar permasalahan yang diteliti. Hal ini meliputi latar belakang permasalahan, rumusan masalah, batasan permasalahan, tujuan penelitian, manfaat penelitian, serta sistematika penulisan yang digunakan dalam penelitian.
- Bab 2 LANDASAN TEORI
  Bagian ini menjelaskan tentang landasan teori yang berkaitan dengan pengerjaan penelitian, antara lain HMAC OTP (RFC 4226), TOTP (RFC 6238), jenis-jenis *encoding*, Basic Authentication (RFC 7617), JSON Web Tokens (RFC 7519), *Open Web Application Security Project*, dan *Technology Acceptance Model*.
- Bab 3 METODOLOGI PENELITIAN
  Bagian ini menjelaskan metodologi penelitian yang digunakan dan diterapkan, seperti *software development life cycle*, analisis kebutuhan sistem, *use-case diagram*, *database schema*, *data flow diagram*, seluruh API *endpoints* yang direncanakan, seluruh *web pages* yang direncanakan, perancangan sistem, beserta perencanaan alur kerja sistem dengan bantuan *flowchart*.
- Bab 4 HASIL DAN DISKUSI
  Bagian ini menjelaskan hasil implementasi dari penelitian, skenario pengujian serangan untuk mengukur tingkat keamanan berdasarkan metrik OWASP, serta hasil evaluasi berdasarkan *Technology Acceptance Model*.
- Bab 5 SIMPULAN DAN SARAN
  Bagian ini menjelaskan kesimpulan dari penelitian yang telah dilakukan beserta saran-saran untuk penelitian-penelitian selanjutnya.