Analisis Dan Implementasi Keamanan *Authentication* Menggunakan *Multi Factor Authentication* (MFA) pada Aplikasi Web

Komang Gede Jaya Wira Buana, Lilik Widyawati, Ondi Asroni

Universitas Bumigora, Mataram, Indonesia

Correspondence : *e-mail*: adisuryadin598@gmail.com

Abstrak

Dalam era digital, aplikasi web menjadi kebutuhan utama di berbagai bidang karena memudahkan aktivitas sekaligus menyimpan banyak data penting, sehingga keamanan menjadi aspek krusial yang mengakibatkan meningkatnya ancaman seperti phishing, pencurian kredensial, dan brute force attack yang pada tahun 2023 mencapai 403 juta anomali trafik di Indonesia. Serangan brute force dan dictionary kerap dimanfaatkan peretas untuk menebak kata sandi dan menembus sistem autentikasi, sehingga diperlukan metode mitigasi yang lebih kuat. Penelitian ini bertujuan menganalisis serta mengimplementasikan Multi Factor Authentication (MFA) dengan Google Authenticator dan verifikasi email sebagai lapisan keamanan tambahan. Hasil pengujian penetration testing menunjukkan bahwa tanpa MFA, sistem dapat ditembus dalam waktu 54 sampai 66 detik, sedangkan setelah penerapan MFA, seluruh upaya brute force gagal karena token dinamis Google Authenticator berubah setiap menit, membuat serangan tidak dapat memvalidasi login. Implementasi ini menggunakan metode SSDLC yang dimodifikasi dan berhasil meningkatkan keamanan autentikasi aplikasi web. Penelitian menyimpulkan bahwa MFA efektif menolak akses tidak sah, menggagalkan serangan brute force, serta memperkecil risiko kebocoran kredensial, sehingga dapat menjadi solusi yang relevan dalam menghadapi ancaman keamanan modern.

Kata kunci: *Google Authenticator, MFA, Verifikasi Email, Web.*

Abstract

In the digital era, web applications have become a major requirement in various fields because they facilitate activities while storing a lot of important data, so security has become a crucial aspect resulting in increased threats such as phishing, credential theft, and brute force attacks which in 2023 reached 403 million traffic anomalies in Indonesia. Brute force and dictionary attacks are often used by hackers to guess passwords and penetrate authentication systems, so stronger mitigation methods are needed. This research aims to analyze and implement Multi Factor Authentication (MFA) with Google Authenticator and email verification as an additional layer of security. The penetration testing results show that without MFA, the system can be penetrated within 54 to 66 seconds, while after the implementation of MFA, all brute force attempts fail because the Google Authenticator dynamic token changes every minute, making the attack unable to validate the login. This implementation uses a modified SSDLC method and successfully improves the authentication security of web applications. The research concludes that MFA effectively denies unauthorized access, thwarts brute force attacks, and minimizes the risk of credential leakage, making it a relevant solution to modern security threats.

Keywords*: Google Authenticator, MFA, Email Verification, Web.*

1. Pendahuluan

Dalam era digital yang semakin maju, tentunya banyak masyarakat yang akan berinteraksi dengan dunia digital salah satunya adalah aplikasi web. Aplikasi berbasis web telah menjadi salah satu kebutuhan utama untuk mendukung berbagai aktivitas, baik di bidang bisnis, pendidikan, maupun hiburan hal tersebut dikarenakan aplikasi web memudahkan banyak pekerjaan masyarakat. Pada aplikasi web masyarakat banyak menyimpan data-data penting di dalamnya, hal ini menjadikan keamanan pada aplikasi web sangat penting [1]. Dengan semakin banyaknya data sensitif yang disimpan dan diakses melalui aplikasi web,

ancaman terhadap keamanan data, seperti peretasan akun dan akses tidak sah, semakin meningkat. Berdasarkan laporan dari beberapa lembaga keamanan siber, serangan seperti *phishing*, pencurian kredensial, dan *brute force attack* terus menjadi ancaman utama yang dihadapi oleh pengguna aplikasi web [2]

Menurut metrotvnews.com mengatakan badan sandi dan siber negara mencatat ada 403 juta anomali *trafik* atau serangan siber ke Indonesia sepanjang 2023. Hal itu terungkap dari data total *trafik* anomali atau serangan siber di Indonesia selama 2023 yang mencapai 403.990.813 anomali. Sementara anomali *trafik* tertinggi terjadi pada Agustus 2023. Jumlahnya mencapai 78 juta anomali. Aktivitas anomali *trafik* ini dapat berdampak pada penurunan performa perangkat dan jaringan, pencurian data sensitif, perusakan reputasi, hingga penurunan kepercayaan terhadap suatu organisasi. Serangan-serangan tersebut tentunya akan sangat berbahaya bagi keamanan data pengguna pada aplikasi web. *Bruteforce attack* dengan teknik *dectionary* menjadi serangan yang sangat sering dipakai pada sistem keamanan *otentikasi* [3].

Teknik *bruteforce* merupakan metode peretasan dengan menggunakan *trial and error* untuk mendapatkan akses *kedalam* akun, kredensial *login*, ataupun kunci enkripsi. Dalam serangan *bruteforce* peretas akan bekerja dengan semua kemungkinan kombinasi dengan harapan dapat menemukan kombinasi yang benar [4] sedangkan teknik *dictionary attack* merupakan serangan pencarian kata sandi dengan mencoba semua kata sandi yang ada dalam kamus atau daftar kata yang sudah ada Kamus ini bisa berupa kata-kata umum, kombinasi angka, huruf, dan simbol, serta kata-kata yang mungkin terkait dengan pengguna atau organisasi tertentu. Serangan ini bertujuan untuk menemukan kata sandi yang tepat atau mendekati kata sandi yang digunakan oleh [5] Untuk mengurangi risiko ancaman dari serangan *bruteforce* dapat menggunakan *Multi Factor Autentication* (MFA).

Multi Factor Authentication (MFA) hadir sebagai solusi untuk memperkuat sistem keamanan *autentikasi*. Dengan mengharuskan pengguna memberikan lebih dari satu faktor untuk membuktikan identitas mereka, MFA mampu mengurangi risiko akses tidak sah secara signifikan[6]. Faktor-faktor dalam MFA biasanya meliputi sesuatu yang diketahui oleh pengguna (seperti kata sandi), sesuatu yang dimiliki (seperti perangkat ponsel atau *token*), dan sesuatu yang melekat pada pengguna (seperti sidik jari atau pengenalan wajah) [7]. Pendekatan ini memastikan bahwa meskipun salah satu faktor terkompromi, lapisan keamanan lainnya tetap melindungi akun pengguna [8]. Pada penelitian ini akan digunakan google atuhetication dan verifikasi email sebagai metode *multi factor authentication*.

Google *Authentication* adalah sistem yang digunakan untuk *mengautentikasi* pengguna saat mereka masuk ke aplikasi atau layanan Google. Ini sering melibatkan penggunaan dua faktor, di mana pengguna harus memasukkan kata sandi dan kemudian menyelesaikan langkah tambahan, seperti memasukkan kode yang dikirim ke ponsel mereka atau menggunakan aplikasi *autentikator* [9][10]. Sedangkan verifikasi email adalah proses yang digunakan untuk memastikan bahwa alamat email yang diberikan oleh pengguna saat mendaftar untuk sebuah akun adalah valid dan dapat diakses oleh [11].

Berdasarkan studi terdahulu, terlihat bahwa penelitian mengenai *Multi Factor Authentication* (MFA) telah banyak dilakukan dengan fokus, teknik, dan konteks penerapan yang berbeda. Wang dan Wang menyoroti kegagalan MFA dalam melindungi aplikasi mobile, sedangkan penelitian ini menekankan keberhasilan MFA dengan Google *Authenticator* dan verifikasi email pada aplikasi web[12]. Nugroho dan Sidqon menerapkan MFA pada sistem manajemen surat digital menggunakan tanda tangan digital, berbeda dengan penelitian ini yang berfokus pada *autentikasi login* web[13]. *Badeges* dan Fauzi menekankan implementasi MFA pada *phpMyAdmin*[14], sementara penelitian ini menggeneralisasi penerapan pada aplikasi web dengan teknik serupa. Lie dan Engel menggunakan NFC dan Google *Authenticator* dalam sistem perpustakaan berbasis Android dan web dengan metode SDLC[9], sedangkan penelitian ini mengadaptasi metode *Secure Software Development Life Cycle* (SSDLC) dengan verifikasi email sebagai tambahan faktor. Sementara itu, Prabakaran dan *Ramachandran* menerapkan MFA pada transaksi keuangan berbasis *cloud* dengan ECC dan *biometrik*, berbeda dengan penelitian ini yang menekankan keamanan *login* berbasis *token* dinamis[7]. Dengan demikian, penelitian ini menghadirkan *state of the art* berupa penerapan MFA menggunakan Google *Authenticator* dan verifikasi email pada aplikasi web dengan pendekatan SSDLC, sehingga menghasilkan sistem *autentikasi* yang lebih aman dan adaptif terhadap ancaman *brute force*.

Berdasarkan pemaparan di atas, penelitian ini bertujuan untuk menganalisis dan mengimplementasikan keamanan *otentikasi* dengan menerapkan teknik *multi factor autentication* dengan menggunakan Google *authenticator* dan verifikasi email, dengan hasil dari penelitian ini merupakan contoh dari bagaimana teknik *multi factor authentication* (MFA) dengan menggunakan teknik Google *authenticator* dan verifikasi email dapat menjadi pengamanan pada sistem *otentikasi* guna memperkecil dampak dari serangan yang menyerang sistem *otentikasi* pada aplikasi web

Berdasarkan latar belakang yang telah dipaparkan penulis merumuskan masalah yaitu bagaimana teknik *multi factor authentication* (MFA) dengan menggunakan teknik Google *authenticator* dan verifikasi email dapat menjadi pengaman pada aplikasi web? Dengan tujuan dari penelitian ini adalah menganalisis dan mengimplementasikan teknik *multi factor authentication* (MFA) dengan menggunakan teknik *google authenticator* dan verifikasi email dapat menjadi pengaman pada sistem *otentikasi* aplikasi web.

2. Metode Penelitian

Metode penelitian adalah cara atau langkah-langkah sistematis yang digunakan untuk mengumpulkan, menganalisis, dan menginterpretasi data guna menjawab pertanyaan atau menguji hipotesis dalam sebuah studi. Secara umum, metode penelitian dibagi menjadi dua jenis utama yakni metode kuantitatif (menggunakan data numerik dan analisis statistik, seperti survei atau eksperimen) dan metode kualitatif (menggunakan data non-numerik, seperti wawancara, observasi, atau studi literature). Pemilihan metode bergantung pada tujuan penelitian, jenis data yang dibutuhkan, serta pendekatan analisis yang relevan dengan permasalahan yang dikaji.

Pada penelitian kali ini penulis memilih studi literatur sebagai metode penelitian. Studi literatur memiliki tujuan untuk mengembangkan konsep teoritis baru dan membentuk kerangka berpikir yang kuat dengan dasar pemahaman yang didapat dari referensi yang dianalisis. Studi literatur membantu peneliti membangun dasar teori yang kokoh sebagai landasan penelitian lebih lanjut.

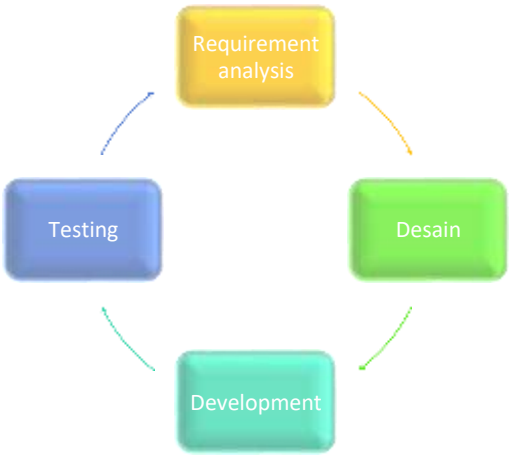
2.1. Studi Literatur

Menurut Permadi & Prihanto, *brute force* dan *dectionary attack* adalah dua metode yang umum digunakan dalam peretasan untuk mendapatkan akses ke sistem atau akun dengan cara mencoba kombinasi kata sandi. Dua metode tersebut memiliki tujuan yang sama yaitu menebak kata sandi, dua metode ini akan terus berjalan hingga mendapatkan kata sandi yang tepat[15]. Dua metode ini sering digunakan untuk membobol kata sandi dikarenakan beberapa alasan seperti sederhana dan efektif, ketersediaan alat dan sumber daya, dan penggunaan *password* lemah yang dengan berbagai alasan tersebut *bruteforce* dan *dectionary attack* tetap menjadi metode populer di kalangan penyerang siber untuk membobol *passoword*.

Menurut [11] Verifikasi email adalah proses yang digunakan untuk memastikan bahwa alamat email yang diberikan oleh pengguna saat mendaftar untuk sebuah akun adalah valid dan dapat diakses oleh mereka. Proses ini tidak cukup kuat sebagai pengamanan dikarenakan email bisa saja di tembus oleh *hacker* oleh karena itu pada penelitian kali ini akan di tambahkan Google *authenticator*. Google *Authenticator* adalah aplikasi mobile yang digunakan untuk *Two-Factor Authentication* (2FA) berbasis Time-Based One-*Time Password* (TOTP) atau HMAC-*Based* One-*Time Password* (HOTP). Aplikasi ini meningkatkan keamanan login dengan menyediakan kode OTP yang berubah setiap beberapa detik, sehingga mengurangi risiko akses yang tidak sah akibat pencurian *password* [14].

2.2. Metode Pengembangan Sistem

Pada tahap ini akan menjelaskan tahapan metode *Secure Software Development Life Cycle* (SSDLC) dengan alur yang digunakan antara lain *Requirements analysis*, desain, *development*, *development*, testing. Gambar 1 menjelaskan mengenai alur dari tahap pengerjaan penelitian:



Gambar 1. Alur SSDLC yang di gunakan

Pada penelitian ini dibuat sistem pengamanan *otentikasi* menggunakan verifikasi email dan Google *authenticator*. Secara garis besar tahapan penelitian ini menggunakan langkah metode pengembangan sistem *Secure Software Development Life Cycle* (SSDLC) yang telah dimodifikasi yang di jelaskan sebagai berikut:

2.1.1. *Requiremnt analysis*

Menganalisis Kerentanan dan kebutuhan untuk membuat contoh sistem keamanan bagaimana teknik *multi factor authentication* (MFA) dengan menggunakan teknik *google authenticator* dan verifikasi email dapat menjadi pengamanan pada sistem *otentikasi* guna memperkecil dampak dari serangan yang menyerang sistem *otentikasi* pada aplikasi web yang dilakukan untuk mengetahui apa saja yang dibutuhkan pada sistem. Kebutuhan yang diperlukan pada penelitian ini yaitu, kebutuhan perangkat lunak, kebutuhan perangkat keras dan kebutuhan sumber daya manusia.

1. Risk *Assesment*

Langkah pertama yang dilakukan *cracker* untuk melakukan serangan *bruteforce* ialah dengan menganalisis target. Analisis dilakukan untuk menemukan celah yang dapat di manfaatkan oleh cracker, cracker akan menekan tombol f12 dan menganalisis struktur dari web yang di targetkan. Selanjutnya cracker akan memanfaatkan ide dari setiap *inputan* untuk dijadikan parameter serangan. Selanjutnya yang dibutuhkan ialah saat uji coba gagal parameter yang dapat digunakan untuk mengulang *itrasi* pengujian. Setelah mendapatkan parameter tersebut selanjutnya akan di buat script *bruteforce* dengan menggunakan python. Setelah script jadi akan disiapkan *dectionary* yang berisi kumpulan *username* dan *password* yang akan diujikan secara kombinasi acak dan akan di lakukan pengujian yang di mana jika pengujian berhasil script akan dapat menembus aplikasi web dan jika tidak script akan eror. Dari hasil pengujian *script* bruteforce dapat menembus sistem dalam waktu seperti gambar berikut:

Gambar 2. Hasil Risk Assesment Dengan Teknik Bruteforce

Terlihat pada terminal yang terdapat pada gambar 2 untuk menemukan email dan *password* yang benar dengan target web yang tidak mengimplementasikan multi factor authentication (MFA) dalam waktu 54.87 detik.

2. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan dalam penelitian ini untuk membangun sistem adalah:

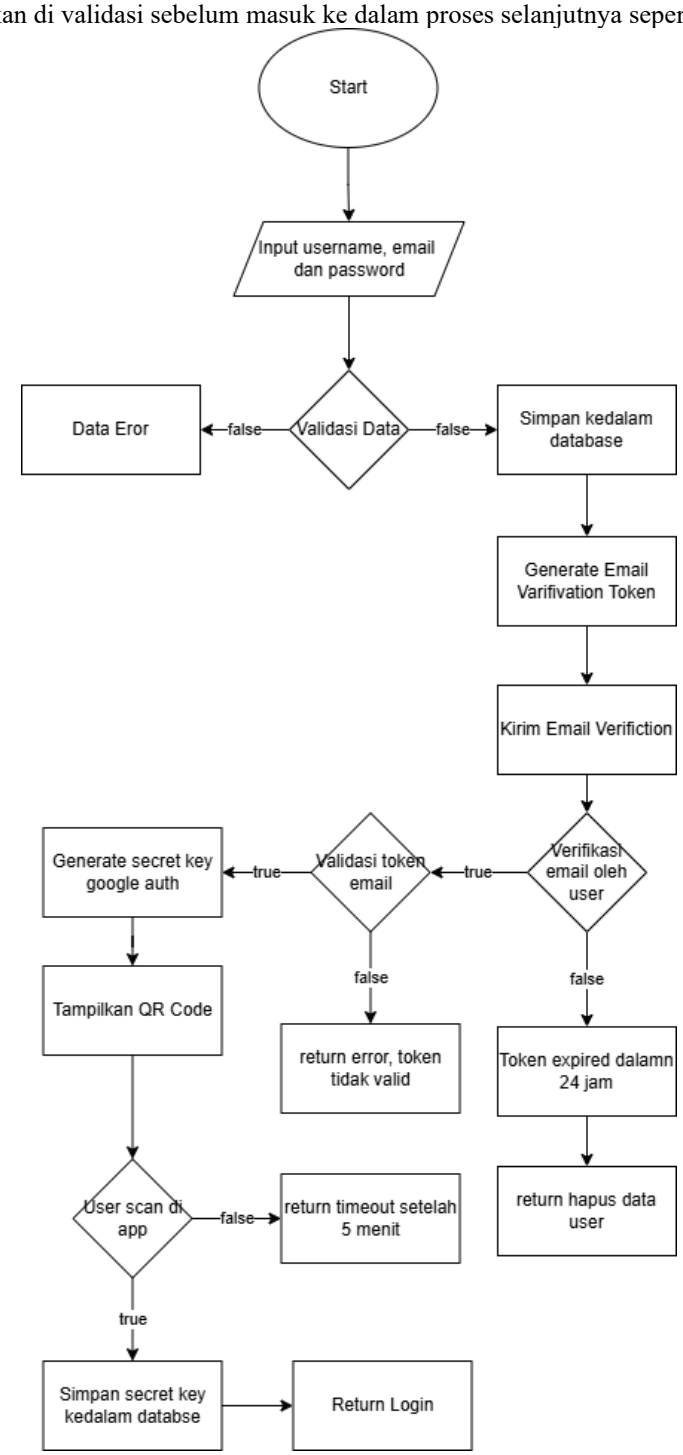
1) Teks editor (Visual Studio Code)
2) Sistem Operasi (Windows 11)
3) NodeJS
4) Bahasa pemrograman (Java Script)
5) Framework (ExpressJS dan React + Vite)
6) Database (MySql)
7) Server database (Laragon)
8) Bahasa pemrograman penetration testing (Python)
9) Google authenticator

2.1.2. Desain Penelitian

Tahap ini dilakukan *thread* modeling dan desain *review* sehingga dapat dihasilkan model proses sistem saat registrasi dan *login* sehingga dapat dilihat secara keseluruhan sebagai berikut:

1. Registrasi

Proses registrasi dimulai dari saat user mendaftarkan akunya yang berisi email aktif, *username* dan *password* yang akan di validasi sebelum masuk ke dalam proses selanjutnya seperti gambar berikut:



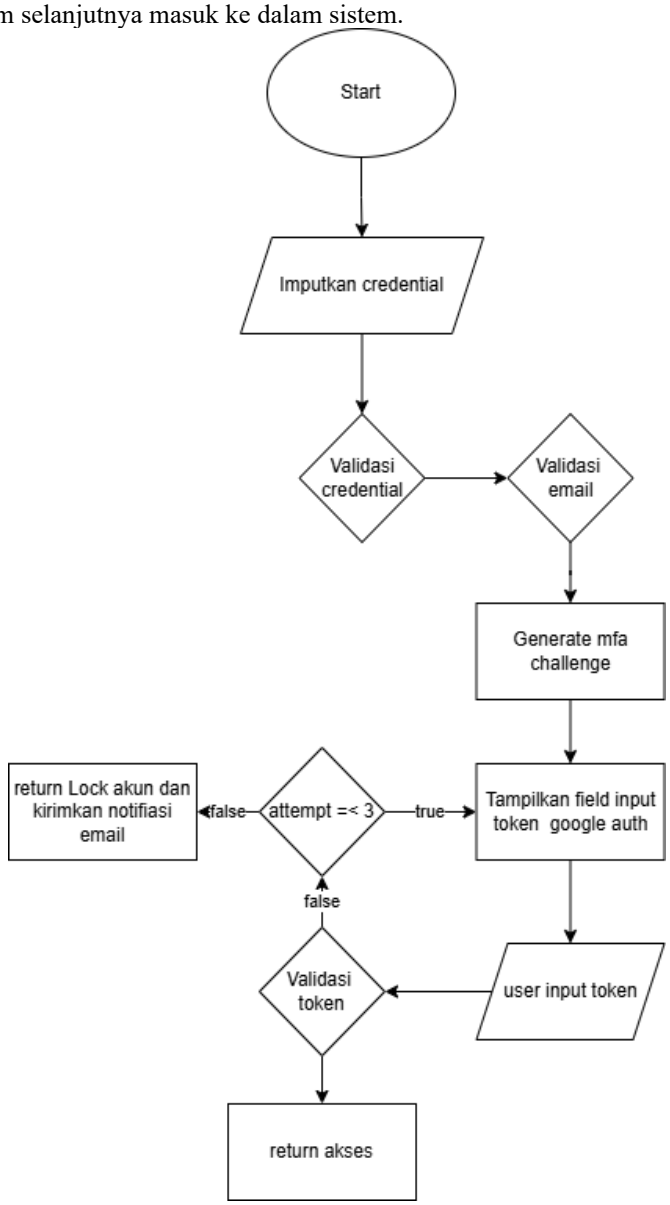
Gambar 3. Flowchart Registrasi

Flowchart pada gambar 3 menjelaskan alur proses registrasi menggunakan multi factior authentication menggunakan Google Authenticator. Proses dimulai dengan pengguna menginput username, email, dan password. Data kemudian divalidasi; jika gagal, sistem akan menampilkan pesan kesalahan. Jika valid, data disimpan ke database dan sistem menghasilkan token verifikasi email yang dikirimkan ke pengguna. Pengguna harus memverifikasi email tersebut dalam 24 jam, jika tidak maka data pengguna akan dihapus. Setelah verifikasi berhasil, token email divalidasi. Jika token tidak valid, sistem mengembalikan pesan kesalahan. Jika valid, sistem membuat secret key Google Authenticator dan menampilkan QR code

untuk discan oleh pengguna melalui aplikasi autentikator. Pengguna memiliki waktu lima menit untuk melakukan scan; jika melebihi waktu, proses dibatalkan. Jika berhasil, secret key disimpan ke database dan proses berakhir dengan pengguna diarahkan ke halaman login.

2. Login

Proses login dimulai saat user menginputkan credential dan akan dilakukan validasi credential dan validasi email sebelum selanjutnya masuk ke dalam sistem.



Gambar 4. Flowchart Login

Flowchart pada gamabr 4 menggambarkan proses otentikasi pengguna dengan dukungan Multi-Factor Authentication (MFA) menggunakan Google Authenticator. Proses dimulai saat pengguna memasukkan kredensial *login*, kemudian dilakukan validasi. Jika kredensial valid, sistem melanjutkan dengan validasi email dan menghasilkan tantangan MFA. Pengguna diminta memasukkan token dari aplikasi Google Authenticator. Jika token tidak valid, sistem akan mencatat jumlah percobaan; jika percobaan melebihi tiga kali, akun akan dikunci dan notifikasi dikirim melalui email. Namun jika token valid, maka akses diberikan. Alur ini bertujuan untuk meningkatkan keamanan dengan menambahkan lapisan verifikasi tambahan.

3. Hasil dan Pembahasan

Tahap ini akan menyajikan hasil analisis dan implementasi dari sistem keamanan autentikasi menggunakan metode Multi Factor Authentication (MFA) pada aplikasi web. Fokus utama pada bab ini

adalah penerapan dua faktor keamanan tambahan, yaitu Google Authenticator dan verifikasi email, untuk meningkatkan lapisan perlindungan terhadap akses pengguna. Data yang diperoleh dari proses implementasi diuji dan dianalisis guna mengevaluasi efektivitas serta keandalan metode yang digunakan dalam mencegah akses tidak sah. Selain itu, pembahasan juga mencakup penyesuaian sistem terhadap kebutuhan pengguna serta evaluasi keamanan berdasarkan hasil pengujian yang telah dilakukan.

3.1. Development

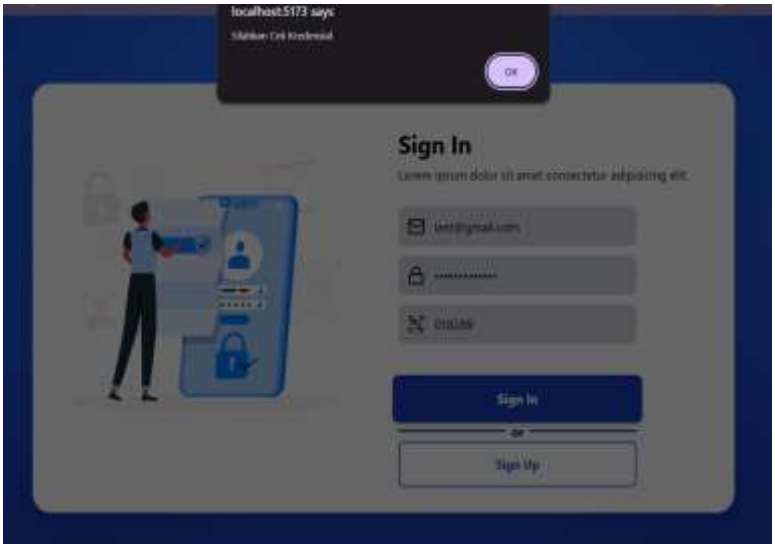
Sub bab ini akan dilakukan implementasi kode membahas tahapan penerapan Multi Factor Authentication (MFA) pada aplikasi web dengan mengintegrasikan password, Google Authenticator berbasis TOTP, dan verifikasi email. Sistem memiliki lima halaman utama, yaitu registrasi, overlay penunjuk langkah, verifikasi email, pemberitahuan email terverifikasi, scan QR code, dan login. Pada tahap registrasi, sistem memvalidasi input, mengenkripsi password, membuat secret untuk Google Authenticator, menghasilkan QR code, serta mengirimkan token verifikasi email yang berlaku 24 jam. Proses verifikasi email dilakukan dengan mencocokkan token yang dikirim ke pengguna, lalu memperbarui status verifikasi di database dan menghapus token yang sudah dipakai. Untuk keamanan tambahan, pengguna harus melakukan verifikasi token dari Google Authenticator, di mana sistem memeriksa kecocokan kode OTP dengan secret yang tersimpan. Pada proses login, sistem mengecek kredensial pengguna, kemudian memastikan 2FA aktif dengan memvalidasi kode OTP; jika valid, sistem menghasilkan JWT token untuk mengakses aplikasi. Dengan alur ini, hanya pengguna yang berhasil melewati seluruh lapisan verifikasi yang dapat login, sehingga keamanan sistem meningkat signifikan.

3.2. Testing

Langkah pertama yang dilakukan *cracker* untuk melakukan serangan *brute-force* adalah menganalisis dari target. Analisis dilakukan untuk melihat struktur dari halaman web yang akan diserang dengan menekan tombol f12 dan memilih menu elemen untuk melihat struktur web yang dipakai seperti yang terlihat pada gambar

Gambar 5. Identifikasi Atribut Aplikasi WEB Multi Factor Authentication

Terlihat pada gambar 5 web memiliki tiga *input field* untuk email, *password* dan *token* yang menggunakan *id* email, *password* dan *otp*. Tiga ide tersebut dapat dijadikan parameter untuk melakukan uji coba serangan *brute force*. Gambar tersebut juga menunjukkan web memiliki *button* yang berfungsi untuk mengirim kredensial yang memiliki teks *Sign In*. Selanjutnya yang dibutuhkan adalah hal yang terjadi jika mengirim kredensial yang salah yang di gunakan sebagai parameter untuk skrip *brute force* memulai percobaan selanjutnya. Yang terlihat pada gambar 6



Gambar 6. Identifikasi Respon Sistem Jika Kredensal Salah

Terlihat saat kredensial salah di inputkan muncul sebuah alert yang bertuliskan “silahkakn Cek Kredensial” alert tersebut dapat dijadikan parameter yang mengindikasikan kredensial salah dan lanjut menuju percobaan selanjutnya. Setelah semua parameter dibutuhkan telah di dapatkan maka script dapat dibuat dengan menggunakan bahasa pemrograman python. Script yang digunakan pada percobaan pada sistem yang telah di lengkapi multi factor authentication sama dengan script yang digunakan saat tahapan risk assessment dan modifikasi sedikit karna harus menyesuaikan field yang di miliki target yang memiliki inputan tambahan untuk memasukkan kode Google authenticator.

Adapun hasil dari penetration testing pada sistem yang telah di bangun menggunakan pengamanan *multi factor authentication* dapat dilihat pada gambar berikut

Gambar 7. Hasil Uji Coba Bruteforce

Gambar 7 menunjukkan hasil dari uji serangan *bruteforce* dengan menggunakan skrip *python* yang telah di buat. Uji coba serangan *beruteforce* di lakukan dengan *dectionary* yang telah di susun untuk menemukan kredensial yang tepat tetapi hingga akhir *dectionary* tidak di temukan kombinasi yang tepat.

3.3. Perbandingan Hasil Uji Coba

Dari hasil uji coba penetration dengan menggunakan teknik *bruteforce* pada web yang sebelum di terapkan teknik multi factor *authentication* (MFA) dan setelah diterapkan teknik MFA maka di dapatkan hasil yang dapat dilihat pada tabel berikut

Tabel 1 Hasil Uji Coba

Percobaan	**Waktu yang di butuhkan menembus sistem dengan menggunakan *Bruteforce* dengan teknik *dectionary***	
	Sebelum	**Sesudah**
1	54.87 detik.	Eror
2	65.76 detik	Eror
3	52.89 detik	Eror

memperlihatkan hasil dari uji coba *penetration* yang di lakukan. Tabel tersebut memperlihatkan bahwa sebelum teknik *multi factor authenticator* di terapkan skrip *bruteforce* dapat menembus sistem dalam waktu 54.87 detik sedangkan saat telah di terapkan teknik MFA skrip bruteforce tidak dapat menembus web tersebut dan mengembalikan eror.

4. Kesimpulan

Hasil penelitian ini menegaskan bahwa penerapan Multi Factor Authentication (MFA) dengan kombinasi Google Authenticator dan verifikasi email mampu meningkatkan keamanan sistem autentikasi pada aplikasi web secara signifikan. Ancaman seperti serangan brute force dan pencurian kredensial yang sering terjadi pada sistem login berbasis kata sandi tunggal dapat diminimalisasi dengan adanya lapisan keamanan tambahan ini. MFA memberikan perlindungan lebih karena pengguna tidak hanya bergantung pada sesuatu yang mereka ketahui, seperti kata sandi, tetapi juga membutuhkan faktor lain yang dimiliki secara langsung, dalam hal ini kode dari Google Authenticator dan konfirmasi melalui email.

Dalam proses pengembangan sistem, metode Secure Software Development Life Cycle (SSDLC) yang dimodifikasi digunakan untuk memastikan keamanan diperhatikan sejak tahap awal hingga akhir. Pendekatan ini mencakup analisis kebutuhan, perancangan, implementasi, serta pengujian. Hasil implementasi menunjukkan bahwa sistem berhasil mencegah akses yang tidak sah ketika salah satu faktor autentikasi tidak terpenuhi. Selain itu, sistem juga terbukti mampu menurunkan risiko keberhasilan serangan brute force, karena penyerang tidak cukup hanya mengetahui kata sandi, tetapi juga memerlukan kode autentikasi yang berubah secara berkala.

Dengan demikian, penerapan MFA terbukti efektif sebagai solusi keamanan tambahan pada sistem login aplikasi web. Integrasi metode ini tidak hanya meningkatkan keandalan autentikasi, tetapi juga memberikan nilai tambah dalam membangun kepercayaan pengguna terhadap keamanan aplikasi. Oleh karena itu, penggunaan MFA dapat direkomendasikan sebagai standar praktik terbaik dalam pengembangan sistem autentikasi modern.

Daftar Pustaka

[1] J. D. Santoso, “ANALISIS PASSWORD CRACKCING MNEGGUNAKAN GPU PROCESS,” *Jurnal Mantik Penusa*, vol. 3, no. 1, 2019.

[2] muhammad fery afrizal Ramadhan and A. S. Ilmananda, “ANALISIS ANCAMAN KEAMANAN PADA SISTEM INFORMASI AKADEMIK KAMPUS MENGGUNAKAN METODE OWASP ZAP,” *JATI(Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 4, 2024.

[3] F. Yeovandi, S. Sabariman, and S. E. Prasetyo, “Evaluasi Keamanan Sistem Autentikasi Biometrik pada Smartphone dan Rekomendasi Implementasi Optimal,” *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 7, no. 1, pp. 133–148, Jan. 2025, doi: 10.35746/jtim.v7i1.653.

[4] P. V Revenkov, A. A. Berdyugin, and P. V Makeev, “Research on Brute Force and Black Box Attacks on ATMs,” 2021. [Online]. Available: https://www.cbr.ru/Content/Document/File/83253/onrib_2021.pdf

[5] A. Sunandar Informatika, U. Singaperbangsa Karawang Jl HSRonggo Waluyo, T. Timur, and J. Barat, “IMPLEMENTASI PENETRATION TESTING DAN WORDLIST GENERATOR DALAM PENGUJIAN KEAMANAN JARINGAN MENGGUNAKAN METODE DICTIONARY ATTACK,” 2024.

[6] B. O. Alsaleem and A. I. Alshoshan, “Multi-Factor Authentication to Systems Login,” *National Computing Colleges Conference (NCCC)*, pp. 1–4, 2021, doi: 10.1109/NCCC49330.2021.9428806./$33.00©20XX.

[7] D. Prabakaran and S. Ramachandran, “Multi-factor authentication for secured financial transactions in cloud environment,” *Computers, Materials and Continua*, vol. 70, no. 1, pp. 1781–1798, 2021, doi: 10.32604/cmc.2022.019591.

[8] M. Haikal Arief, K. Arifa Fitri, and E. Malays Sari, “Analisis Kesadaran Cyber Crime Di Kalangan Masyarakat Menengah Kebawah,” vol. 25, no. 2, pp. 24–39, 2024, doi: 10.37817/tekinfo.v25i2.

[9] H. David Lie and M. Maoeretz Engel, “LIBRARY SELF SERVICE SYSTEM USING NFC AND 2FA GOOGLE AUTHENTICATOR,” *Jurnal Teknik Informatika (JUTIF)*, vol. 3, no. 3, 2022, doi: 10.20884/1.jutif.2022.3.3.345.

[10] H. David Lie and M. Maoeretz Engel, “LIBRARY SELF SERVICE SYSTEM USING NFC AND 2FA GOOGLE AUTHENTICATOR,” *Jurnal Teknik Informatika (JUTIF)*, vol. 3, no. 3, 2022, doi: 10.20884/1.jutif.2022.3.3.345.

[11] A. Z. Mubarok, Carudin, and A. Voutama, "Perancangan User Interface/User Experience Pada Aplikasi Baby Spa Berbasis Mobile Untuk User Customer Dan Terapis Menggunakan Metode User Centered Design," *Jurnal Pendidkan dan Konseling*, vol. 4, no. 5, 2022.
[12] Q. Wang and D. Wang, "Understanding Failures in Security Proofs of Multi-factor Authentication for Mobile Devices," 2022.
[13] R. A. Nugroho, M. Sidqon, S. Si, and M. Si, "RANCANG BANGUN SISTEM INFORMASI MANAJEMEN SURAT INTERNAL BERBASIS WEB DENGAN MULTI-FACTOR AUTHENTICATION (MFA) PADA PT. PELABUHAN INDONESIA III (PERSERO)," 2021.
[14] W. Badeges and M. N. Fauzi, "IMPLEMENTASI MULTI FACTOR AUTHENTICATION PADA PHPMYADMIN," *Jurnal Pendidikan Teknologi Informasi (TRIPLE A)*, vol. 2, no. 1, 2023.
[15] A. S. Permadi and A. Prihanto, "Simulasi Monitoring Jaringan Menggunakan Aplikasi The Dude Dengan Notifikasi Whatsapp," *Journal of Informatics and Computer Science*, vol. 05, 2023.