Jurnal Ilmu Komputer dan Sistem Informasi (JIRSI)
Volume: 5, Nomor: 2, Mei 2026: 591-606
https://jurnal.unity-academy.sch.id/index.php/jirsi/index

e-ISSN 2830-3954
p-ISSN 2830-6031

Modifikasi Keamanan Otentikasi OTP Menggunakan Algoritma HMAC-SHA256 pada Sistem Informasi PT Indonesia Gadai Oke

Modification of OTP Authentication Security Using the HMAC-SHA256 Algorithm in the Information System of PT Indonesia Gadai Oke

Alvin Lie*[1], Martiano[2]
[1,2]Program Studi Sistem Informasi, Universitas Muhammadiyah Sumatera Utara
E-mail: [1] alvinlie1704@gmail.com , [2]martiano@umsu.ac.id

Abstrak

Keamanan data pengguna pada sistem informasi berbasis web seringkali terancam oleh kelemahan mekanisme otentikasi konvensional yang hanya mengandalkan kata sandi statis. Serangan seperti pencurian kredensial (credential theft) dan brute force pada sistem informasi internal PT. Indonesia Gadai Oke menuntut adanya lapisan keamanan tambahan untuk melindungi data sensitif nasabah dan transaksi keuangan. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem keamanan Two-Factor Authentication (2FA) menggunakan algoritma HMAC-SHA256 (Hash-based Message Authentication Code with Secure Hash Algorithm 256-bit) yang terintegrasi dengan fitur Trusted Device dan notifikasi Web Push. Metode yang diterapkan adalah Time-based One-Time Password (TOTP) dengan interval waktu 30 detik. Kode unik dibangkitkan di sisi server melalui proses dynamic truncation 32-bit dari hasil enkripsi SHA-256 yang menggabungkan kunci rahasia (secret key) dengan timestamp. Penggunaan Web Push Notification dipilih sebagai media distribusi untuk menghilangkan biaya operasional SMS dan meminimalisir latensi pengiriman. Pengujian sistem dilakukan menggunakan metode Black Box Testing dan Security Testing dengan skenario pengulangan sebanyak 10 kali percobaan. Hasil penelitian menunjukkan bahwa sistem memiliki tingkat keberhasilan fungsional sebesar 100% dalam memvalidasi pengguna yang sah. Secara keamanan, sistem terbukti efektif memitigasi ancaman dengan tingkat keberhasilan 100% dalam menolak serangan SQL Injection, Cross-Site Scripting (XSS), dan Replay Attack melalui mekanisme validasi token satu kali pakai. Implementasi ini berhasil menurunkan risiko pembajakan akun dan meningkatkan efisiensi proses otentikasi pada PT. Indonesia Gadai Oke.

***Kata kunci**: Two-Factor Authentication; HMAC-SHA256*

Abstract

The security of user data in web-based information systems is frequently compromised by the weaknesses of conventional authentication mechanisms that rely solely on static passwords. Attacks such as credential theft and brute force on the internal information system of PT. Indonesia Gadai Oke necessitate an additional layer of security to protect sensitive customer data and financial transactions. This research aims to design and implement a Two-Factor Authentication (2FA) security system using the HMAC-SHA256 (Hash-based Message Authentication Code with Secure Hash Algorithm 256-bit) algorithm, integrated with Trusted Device features and Web Push notifications. The applied method is Time-based One-Time Password (TOTP) with a 30-second time interval. A unique code is generated on the server side through a 32-bit dynamic truncation process of the SHA-256 encryption result, which combines a secret key with a timestamp. Web Push Notification was chosen as the distribution medium to eliminate SMS operational costs and minimize delivery latency. System testing was conducted using Black Box Testing and Security Testing methods with a scenario of 10

experimental repetitions. The results indicate that the system achieved a 100% functional success rate in validating authorized users. In terms of security, the system proved effective in mitigating threats with a 100% success rate in rejecting SQL Injection, Cross-Site Scripting (XSS), and Replay Attacks through a single-use token validation mechanism. This implementation successfully reduced the risk of account hijacking and improved the efficiency of the authentication process at PT. Indonesia Gadai Oke.

***Keywords**: Two-Factor Authentication; HMAC-SHA256*

1. PENDAHULUAN

Perkembangan teknologi informasi yang sangat pesat telah mendorong berbagai organisasi, termasuk perusahaan jasa keuangan seperti PT. Indonesia Gadai Oke, untuk melakukan digitalisasi proses bisnis. Sistem informasi tidak lagi hanya berperan sebagai pendukung operasional, tetapi telah menjadi komponen inti dalam pengelolaan, penyimpanan, serta distribusi data yang bersifat strategis dan sensitif. Kondisi ini menjadikan aspek keamanan informasi sebagai kebutuhan yang sangat penting, khususnya dalam menjaga kerahasiaan (confidentiality), integritas (integrity), dan ketersediaan (availability) data dari berbagai ancaman kejahatan siber integritas, dan ketersediaan data dari berbagai ancaman kejahatan siber [1]. Namun, ketergantungan pada mekanisme otentikasi konvensional yang hanya mengandalkan kata sandi statis menjadi titik lemah yang kritikal. Serangan seperti credential theft, phishing, dan brute force seringkali berhasil menembus pertahanan sistem, yang dapat berdampak pada kebocoran data nasabah hingga kerugian finansial yang signifikan [2].

Permasalahan utama yang diidentifikasi adalah kelemahan sistem otentikasi satu faktor (*Single Factor Authentication*) yang membuat akun pengguna mudah dikuasai pihak tidak bertanggung jawab jika kata sandi berhasil dicuri. Selain itu, meskipun metode *Two-Factor Authentication* (2FA) berbasis SMS telah ada, biaya operasional yang tinggi dan masalah latensi pengiriman sering kali menjadi hambatan bagi efisiensi perusahaan. Oleh karena itu, penelitian ini merumuskan masalah pada bagaimana merancang sistem keamanan tambahan yang kuat namun efisien menggunakan algoritma HMAC-SHA256 yang diintegrasikan dengan fitur *Trusted Device* dan notifikasi *Web Push*. Tujuan utama dari penelitian ini adalah untuk membangun lapisan keamanan *Time-based One-Time Password* (TOTP) yang dinamis untuk memitigasi serangan siber dan meningkatkan kepercayaan operasional pada sistem informasi PT. Indonesia Gadai Oke.

Secara teoretis, penelitian ini menggunakan algoritma HMAC-SHA256 sebagai landasan utama karena kemampuannya dalam melakukan enkripsi satu arah yang aman dengan memanfaatkan secret key dan timestamp [3]. Penggunaan algoritma ini dipadukan dengan metode TOTP yang memastikan bahwa setiap kode OTP hanya berlaku dalam durasi singkat, yakni 30 detik, sehingga efektif menangkal serangan Replay Attack. Untuk mengatasi kelemahan biaya SMS, teknologi Web Push Notification diimplementasikan sebagai media distribusi kode yang lebih cepat dan hemat biaya karena memanfaatkan jalur komunikasi browser [4]. Selain itu, konsep Trusted Device diterapkan untuk mengenali perangkat sah pengguna guna memberikan kenyamanan akses tanpa mengurangi standar keamanan. Melalui

integrasi teknologi ini, diharapkan tercipta sebuah ekosistem keamanan digital yang tangguh dan mampu memberikan proteksi maksimal terhadap ancaman SQL Injection maupun Cross-Site Scripting (XSS) pada sistem informasi berbasis web.

Salah satu celah keamanan yang sering dimanfaatkan oleh pelaku kejahatan siber adalah mekanisme otentikasi pengguna. Penggunaan metode autentikasi konvensional seperti username dan password statis dinilai sudah tidak lagi memadai karena rentan terhadap berbagai serangan, seperti phishing, brute force, keylogger, serta pencurian kredensial [5]. Ketika informasi login berhasil diperoleh oleh pihak yang tidak berwenang, sistem dapat diakses secara ilegal tanpa terdeteksi, sehingga berpotensi menimbulkan kerugian finansial, gangguan operasional, serta menurunnya kepercayaan pengguna terhadap sistem.

Sebagai upaya meningkatkan keamanan otentikasi, banyak sistem informasi mulai menerapkan mekanisme otentikasi berlapis (multi-factor authentication), salah satunya melalui penggunaan One Time Password (OTP) [6]. OTP merupakan kata sandi dinamis yang hanya berlaku untuk satu kali penggunaan dalam jangka waktu tertentu. Dengan karakteristik tersebut, OTP mampu mengurangi risiko penyalahgunaan kredensial, karena kode yang telah digunakan atau telah kedaluwarsa tidak dapat digunakan kembali oleh pihak lain.

Namun demikian, implementasi OTP konvensional masih menyisakan celah keamanan, baik dari sisi algoritma pembangkit maupun media distribusinya. Penelitian terdahulu menunjukkan bahwa OTP yang dibangkitkan dengan bilangan acak sederhana tanpa proteksi hash kriptografis rentan terhadap serangan prediksi [7]. Selain itu, kelemahan fatal juga ditemukan pada metode distribusi kode OTP yang umum digunakan, seperti melalui Email, WhatsApp, atau SMS. Mekanisme OTP pada ekosistem seluler memiliki kerentanan signifikan terhadap manipulasi dan intersepsi jika tidak dilindungi dengan enkripsi yang memadai. Risiko ini diperparah dengan masalah latency (keterlambatan) yang tinggi akibat ketergantungan pada jaringan pihak ketiga, serta rentan terhadap serangan Man-in-the-Middle (MitM) dan penyadapan (sniffing). Penggunaan media pihak ketiga ini juga memunculkan isu privasi data dan biaya operasional yang berkelanjutan bagi perusahaan [5].

Berdasarkan permasalahan tersebut, penelitian ini merumuskan solusi melalui modifikasi sistem keamanan otentikasi menggunakan algoritma HMAC-SHA256 (Hash-based Message Authentication Code with Secure Hash Algorithm 256-bit). Algoritma ini dipilih karena kemampuannya dalam melakukan enkripsi satu arah yang tangguh dengan memanfaatkan kombinasi kunci rahasia (secret key) dan timestamp [3]. Untuk mengatasi kendala biaya dan latensi pada media konvensional, penelitian ini mengintegrasikan fitur Web Push Notification sebagai kanal distribusi OTP yang lebih efisien karena berjalan langsung pada peramban (browser) tanpa ketergantungan pada operator seluler. Selain itu, guna meningkatkan kenyamanan pengguna tanpa mengorbankan keamanan, diterapkan pula fitur Trusted Device yang memungkinkan sistem mengenali perangkat sah pengguna untuk memitigasi serangan Replay Attack. Tujuan akhir dari penelitian ini

adalah mengimplementasikan dan menguji efektivitas sistem tersebut pada PT. Indonesia Gadai Oke melalui metode Black Box dan Security Testing, guna memastikan keberhasilan validasi pengguna dan ketahanan terhadap serangan siber seperti SQL Injection dan XSS.

Penelitian lain membuktikan bahwa penerapan algoritma HMAC-SHA256 pada sistem otentikasi mampu menjawab kelemahan dari sisi pembangkitan kode [1]. Algoritma HMAC-SHA256 menghasilkan nilai hash yang bergantung pada kunci rahasia dan pesan input, sehingga sulit dipalsukan. Algoritma ini sangat tepat jika dikombinasikan dengan mekanisme distribusi Web Push yang diusulkan penulis untuk menciptakan sistem keamanan yang tangguh namun tetap ringan secara komputasi.

Dalam konteks perusahaan jasa keuangan, PT. Indonesia Gadai Oke mengelola berbagai data sensitif, seperti data nasabah, data jaminan, serta transaksi keuangan. Seiring meningkatnya jumlah pengguna dan intensitas transaksi melalui sistem informasi, risiko terhadap serangan siber juga semakin besar. Berdasarkan pengamatan penulis di lapangan, efisiensi waktu saat proses login sangat mempengaruhi produktivitas kerja pegawai. Keterlambatan penerimaan kode OTP via SMS seringkali menghambat akses cepat ke data nasabah yang dibutuhkan. Oleh karena itu, apabila sistem otentikasi masih mengandalkan metode lama yang lambat dan berbayar, hal ini tidak hanya menjadi ancaman keamanan tetapi juga hambatan operasional yang serius. Apabila sistem otentikasi masih mengandalkan metode pengiriman yang rentan disadap atau mengalami keterlambatan, maka potensi terjadinya akses tidak sah dan hambatan operasional menjadi ancaman serius bagi perusahaan [8].

Permasalahan utama yang dikaji dalam penelitian ini adalah bagaimana mengimplementasikan algoritma HMAC-SHA256 dalam pembangkitan *One Time Password* (OTP) dan mengukur kinerjanya dari aspek kecepatan, efisiensi, serta keandalan pada sistem informasi PT. Indonesia Gadai Oke. Sejalan dengan masalah tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan mekanisme OTP berbasis HMAC-SHA256 yang terintegrasi dengan antarmuka *Web Push Notification* guna memastikan kemudahan interaksi pengguna serta mengevaluasi peningkatan keamanan sistem dari akses tidak sah.

Secara teoretis, sistem ini dibangun di atas landasan keamanan informasi yang menjaga kerahasiaan, integritas, dan ketersediaan data. Implementasi *Two-Factor Authentication* (2FA) menggunakan algoritma HMAC-SHA256 dipilih karena kemampuannya menghasilkan *message digest* yang unik, tidak dapat dipulihkan (*irreversible*), dan tahan terhadap serangan *collision* maupun *brute force*. Penggunaan metode *Time-based One-Time Password* (TOTP) dengan interval 30 detik memastikan kode selalu dinamis, sementara teknologi *Web Push Notification* melalui *Service Worker* digunakan sebagai media distribusi mandiri untuk mengeliminasi ketergantungan pada pihak ketiga dan meminimalisir latensi. Evaluasi kinerja sistem nantinya akan mengacu pada standar ISO/IEC 25010 untuk

menilai efisiensi waktu dan penggunaan sumber daya guna memastikan bahwa lapisan keamanan tambahan ini tidak membebani performa server perusahaan.

Berdasarkan permasalahan pada kelemahan distribusi konvensional dan kebutuhan akan algoritma yang kuat, diperlukan adanya modifikasi mekanisme keamanan otentikasi yang lebih komprehensif. Algoritma HMAC-SHA256 dipilih karena mampu memberikan jaminan integritas dan keaslian kode melalui fungsi hash yang kuat, serta memungkinkan penerapan verifikasi yang tidak bergantung pada pengiriman pesan eksternal (Email/WhatsApp) yang sering kali mengalami kendala biaya dan latensi [6]. Melalui proses dynamic truncation 32-bit, algoritma ini mampu membangkitkan kode unik yang dinamis dan sulit diprediksi oleh pihak yang tidak berwenang. Hal ini diharapkan dapat meningkatkan kinerja sistem secara signifikan, baik dari sisi kecepatan proses otentikasi maupun efisiensi sumber daya server.

Selain penguatan pada aspek algoritma, penelitian ini juga mengintegrasikan fitur *Trusted Device* dan notifikasi *Web Push* sebagai solusi atas ketergantungan pada jaringan operator seluler. Fitur *Trusted Device* berfungsi untuk mengenali perangkat sah milik pengguna, sehingga dapat memberikan perlindungan berlapis terhadap upaya login dari perangkat asing yang mencurigakan. Sementara itu, implementasi *Web Push Notification* memungkinkan pengiriman kode OTP secara *real-time* langsung ke peramban pengguna, yang terbukti lebih efisien dan aman dibandingkan metode SMS konvensional. Dengan mengombinasikan teknologi ini, sistem tidak hanya mampu memitigasi serangan *credential theft* dan *brute force*, tetapi juga memberikan ketahanan terhadap serangan *SQL Injection*, *Cross-Site Scripting* (XSS), hingga *Replay Attack*.

Oleh karena itu, penelitian ini disusun dengan judul **"**Modifikasi Keamanan Otentikasi One Time Password Menggunakan Algoritma HMAC-SHA256 pada Sistem Informasi PT. Indonesia Gadai Oke**"**. Penelitian ini dilakukan untuk merancang, mengimplementasikan, serta menganalisis efektivitas penerapan algoritma HMAC-SHA256 dan pengaruhnya terhadap peningkatan keamanan serta kinerja sistem secara keseluruhan dibandingkan dengan metode otentikasi sebelumnya. Hasil dari penelitian ini diharapkan dapat menjadi standarisasi keamanan baru pada PT. Indonesia Gadai Oke dalam melindungi data aset dan transaksi nasabah.

2. METODOLOGI PENELITIAN

2.1 Kronologis Penelitian

Penelitian ini dilaksanakan melalui urutan kerja yang sistematis untuk memastikan hasil yang akurat. Proses dimulai dengan Studi Pendahuluan melalui observasi dan wawancara di PT. Indonesia Gadai Oke untuk memetakan kelemahan sistem otentikasi saat ini. Selanjutnya dilakukan Studi Literatur untuk mendalami landasan teori algoritma HMAC-SHA256, teknologi Web Push, dan standar RFC 6238 mengenai TOTP. Tahapan kemudian dilanjutkan ke Perancangan Sistem (UML), Implementasi Kode (*Coding*), hingga Pengujian dan Evaluasi.

2.2 Bahan dan Perangkat Penelitian

Bahan utama dalam penelitian ini adalah data kredensial karyawan dan skema basis data sistem informasi PT. Indonesia Gadai Oke. Untuk perangkat keras, digunakan laptop dengan prosesor Intel Core i5 dan RAM 8GB. Perangkat lunak yang digunakan meliputi:

1. Bahasa Pemrograman: PHP (Native) untuk *backend* dan JavaScript untuk *Service Worker*.
2. Basis Data: MySQL/MariaDB.
3. Web Server: Apache (via XAMPP).
4. Libraries: Minishlink/WebPush PHP Library dan OpenSSL.
5. Browser: Google Chrome/Microsoft Edge (yang mendukung Push API).

2.3 Rancangan dan Desain Penelitian

Desain penelitian ini menggunakan model arsitektur *Client-Server* dengan penguatan pada sisi server (*Server-Side Generation*). Rancangan sistem divisualisasikan melalui:

1. Use Case Diagram: Menggambarkan interaksi pengguna saat login dan menerima notifikasi OTP.
2. Sequence Diagram: Menunjukkan urutan waktu pengiriman paket data dari server ke peramban melalui *Push Service* pihak ketiga.
3. Class Diagram**:** Menjelaskan struktur data kunci rahasia (*Secret Key*) dan *Timestamp*.

2.4 Prosedur Penelitian (Algoritma HMAC-SHA256)

Prosedur utama dalam sistem ini adalah pembangkitan kode OTP. Berikut adalah langkah-langkah algoritmanya:

Input: Ambil *Secret Key* (K) dari database dan *Current Time* (T).
Step 1: Hitung nilai *Counter* (C) dengan rumus: $C = floor(T / 30)$.
Step 2: Lakukan *Hashing* HMAC menggunakan SHA-256: $Hash = HMAC-SHA256(K, C)$.
Step 3: Terapkan *Dynamic Truncation*:
Ambil 4 byte dari posisi *offset* tertentu pada hasil *hash*.
Konversi 4 byte tersebut menjadi bilangan bulat 31-bit.
Step 4: Lakukan operasi Modulo 10^6 untuk menghasilkan 6 digit angka.
Output: Kirim kode 6 digit tersebut melalui *Web Push Notification* ke peramban pengguna.

2.5 Cara Pengujian dan Pengambilan Data

Pengambilan data dilakukan melalui pengujian terkontrol dengan skenario sebagai berikut:

a) Black Box Testing: Menguji fungsionalitas setiap tombol, validitas input karakter pada form login, dan kemunculan pop-up notifikasi pada kondisi peramban aktif maupun *minimized*.
b) Security Testing (Pengujian Keamanan): * Replay Attack: Mencoba memasukkan kembali kode OTP yang sama setelah melewati batas waktu 30 detik atau setelah kode digunakan.

- Brute Force: Melakukan percobaan input kode secara berulang untuk melihat respon sistem keamanan.
- Injeksi: Menguji keamanan form login terhadap input skrip SQL dan XSS.

c) Pengukuran Latensi: Mencatat selisih waktu (dalam milidetik) antara saat tombol login ditekan hingga notifikasi muncul pada layar pengguna untuk mengevaluasi efisiensi *Web Push*.

3. HASIL DAN PEMBAHASAN

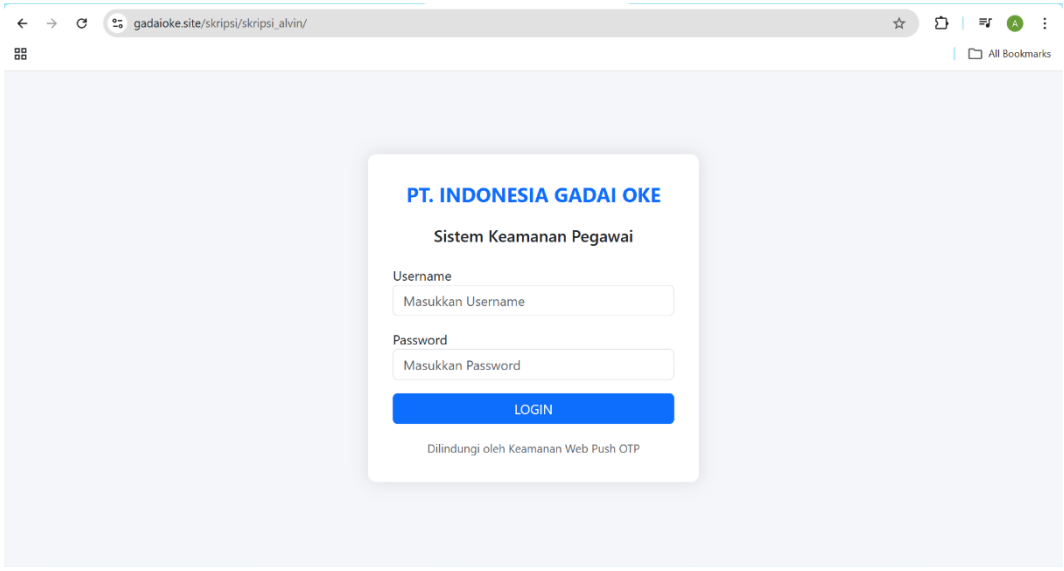
Implementasi Sistem

Implementasi merupakan tahapan penerapan rancangan sistem menjadi perangkat lunak yang utuh. Sistem otentikasi ini dibangun menggunakan bahasa pemrograman PHP dengan algoritma keamanan HMAC-SHA256 yang berjalan di lingkungan server lokal (localhost) menggunakan XAMPP v3.3.0.

4.1 Implementasi Antarmuka (User Interface)

Antarmuka sistem dirancang agar memudahkan pengguna dalam melakukan proses login dan verifikasi dua langkah (Two-Factor Authentication).

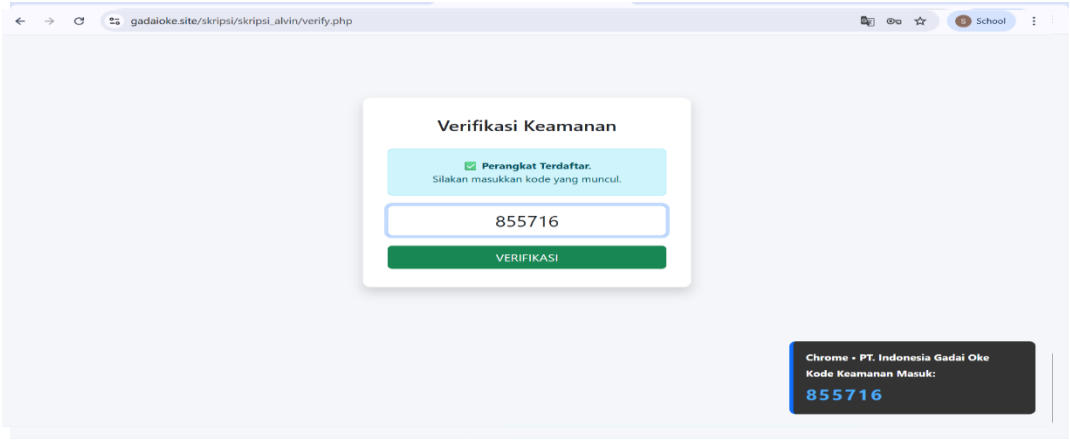
a. Halaman Login Halaman ini merupakan gerbang utama sistem. Pengguna diwajibkan memasukkan username dan password yang terdaftar.



Gambar 4. 1 Tampilan Halaman Login Pegawai

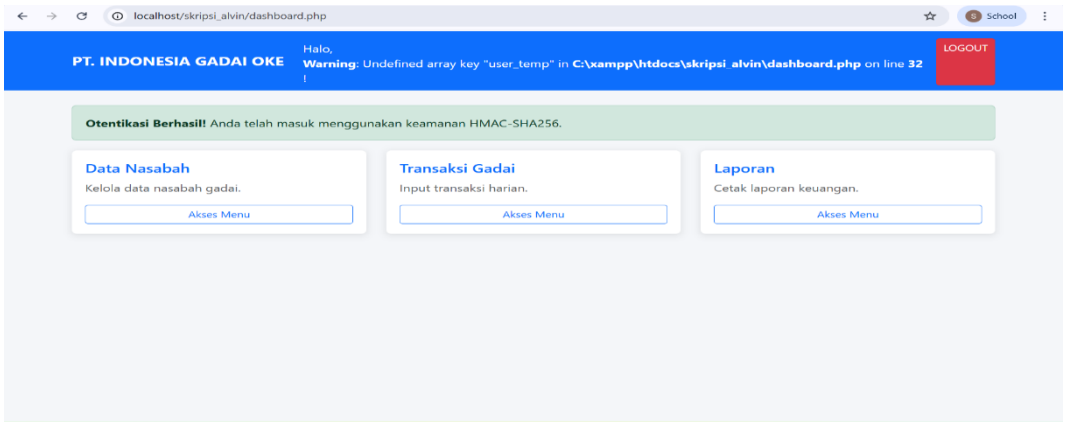
b. Notifikasi Kode OTP (Web Push Notification) Setelah pengguna berhasil memasukkan kredensial yang benar, sistem secara otomatis mengirimkan kode OTP melalui mekanisme Web Push yang muncul pada peramban (browser) pengguna tanpa memerlukan aplikasi tambahan.

c. Halaman Verifikasi OTP Pengguna diminta memasukkan 6 digit kode yang diterima melalui notifikasi ke dalam kolom validasi untuk mendapatkan akses penuh ke dalam sistem.
d. Halaman Utama (*Dashboard*)



Setelah proses verifikasi OTP berhasil, pengguna akan diarahkan secara otomatis ke halaman *dashboard* utama. Halaman ini hanya dapat diakses jika pengguna telah melewati dua lapisan keamanan (*password* dan OTP).

Gambar 4. 2 Tampilan Halaman Validasi Token



Gambar 4. 3 Tampilan Dashboard Utama setelah Login Berhasil

4.1.2 Implementasi Kode Program (Algoritma HMAC)

Keamanan utama sistem terletak pada proses pembentukan token OTP yang menggunakan fungsi hash satu arah. Berikut adalah potongan kode (*snippet code*) implementasi algoritma HMAC-SHA256 pada file otp_process.php :

Gambar 4. 4 Potongan Kode Implementasi Algoritma HMAC-SHA256

Pada kode di atas, fungsi hash_hmac digunakan untuk mengenkripsi data gabungan antara *username* dan waktu saat ini (*timestamp*) menggunakan kunci rahasia. Hal ini memastikan bahwa kode OTP yang dihasilkan selalu unik setiap detiknya dan tidak dapat ditebak (*unpredictable*).

4.2 Pengujian Sistem

Pengujian sistem merupakan tahapan krusial untuk memastikan perangkat lunak yang dibangun telah memenuhi kebutuhan fungsional dan memiliki standar keamanan yang memadai. Zen et al. [9] menjelaskan bahwa pengujian perangkat lunak bertujuan untuk menemukan cacat (defects) dan memastikan bahwa fungsionalitas sistem berjalan sesuai dengan spesifikasi yang diharapkan sebelum diserahkan kepada pengguna akhir.

Dalam penelitian ini, pengujian dilakukan menggunakan pendekatan *Black Box Testing* untuk memvalidasi fungsi input-output, serta *Security Testing* untuk menguji ketahanan algoritma HMAC-SHA256 terhadap serangan siber.

4.2.1 Pengujian Fungsional (*Black Box Testing*)

Tahap pengujian fungsional dilakukan untuk memverifikasi kesesuaian antara masukan (input) dan keluaran (output) sistem tanpa melihat struktur kode internalnya. Metode yang digunakan adalah Black Box Testing. Menurut [10], Black Box Testing merupakan metode validasi perangkat lunak yang berfokus pada persyaratan fungsional untuk memastikan seluruh fitur berjalan sesuai spesifikasi kebutuhan pengguna. Pendekatan ini dipilih karena efektif dalam mendeteksi kesalahan antarmuka, inisialisasi fungsi, dan kesalahan kinerja pada tahap akhir pengembangan sistem [11].

Dalam penelitian ini, skenario pengujian difokuskan pada validitas proses otentikasi, penerimaan notifikasi Web Push, serta mekanisme validasi token HMAC-SHA256. Guna menjamin akurasi dan reliabilitas hasil pengujian, setiap skenario dilakukan secara berulang (iterative testing). Hal ini merujuk pada standar pengujian perangkat lunak yang menyatakan bahwa konsistensi sistem hanya dapat dibuktikan melalui pengulangan eksekusi uji coba minimal 10 kali pada kondisi jaringan yang berbeda [12].

Data keberhasilan pengujian dihitung menggunakan persentase kelayakan dengan rumus:

$$Persentase = \left(\frac{\sum Percobaan\ Berhasil}{\sum Total\ Percobaan} \right) x\ 100\% \quad \text{.....(4.1)}$$

Rumus 4.1 Perhitungan Persen Percobaan Berhasil

Berikut adalah rekapitulasi hasil pengujian fungsional yang disajikan pada Tabel 4.1:

Tabel 4.1 Hasil Pengujian Black Box (10x Percobaan)

No	Skenario Pengujian	Hasil yang Diharapkan	Σ Uji	Valid	Gagal	Hasil (%)	Ket.
1	Login Valid	Notifikasi OTP Web Push diterima.	10	10	0	100%	Berhasil
2	Login Invalid	Pesan error muncul, OTP tidak dikirim.	10	10	0	100%	Berhasil
3	Integritas OTP	Kode di Server = Kode di Browser.	10	10	0	100%	Berhasil
4	Anti-Replay	Kode bekas ditolak sistem.	10	10	0	100%	Aman
5	Expired Token	Kode > 30 detik ditolak.	10	10	0	100%	Aman

Berdasarkan Tabel 4.1, sistem menunjukkan tingkat keberhasilan 100% pada seluruh skenario uji. Hasil ini sejalan dengan penelitian [3] yang menyimpulkan bahwa implementasi algoritma HMAC pada otentikasi dua faktor mampu menjamin integritas data secara real-time tanpa kegagalan transmisi, selama latensi jaringan tetap terjaga. Pada pengujian nomor 4 (Anti-Replay), sistem berhasil menolak 10 kali percobaan penggunaan token bekas, yang membuktikan bahwa mekanisme One-Time Password berfungsi optimal dalam mencegah akses ilegal.

4.2.2 Pengujian Keamanan Sistem (*Security Testing*)

Setelah fungsionalitas utama dipastikan berjalan dengan baik, tahap selanjutnya adalah pengujian keamanan (Security Testing). Pengujian ini bertujuan untuk mengevaluasi ketahanan sistem terhadap berbagai ancaman siber dan mendeteksi adanya celah kerentanan (vulnerability) yang mungkin dieksploitasi oleh pihak yang tidak berwenang. Menurut penelitian terbaru [13], pengujian keamanan pada aplikasi web modern wajib mencakup simulasi serangan terhadap mekanisme otentikasi dan validasi input untuk meminimalisir risiko kebocoran data.

Dalam penelitian ini, metode pengujian keamanan mengacu pada standar OWASP (Open Web Application Security Project) dengan fokus pada tiga vektor serangan utama: SQL Injection, Cross-Site Scripting (XSS), dan Brute Force Attack. Sejalan dengan metodologi pengujian sebelumnya, setiap skenario serangan disimulasikan sebanyak 10 kali percobaan untuk mendapatkan data persentase keberhasilan sistem dalam memblokir ancaman tersebut [14].

Rumus perhitungan efektivitas keamanan sistem adalah:

$$Persentase = \left(\frac{\sum Serangan\ Digagalkan}{\sum Total\ Serangan} \right) x\ 100\% \quad \text{.....(4.2)}$$

Rumus 4.2 Perhitungan Persen Serangan Digagalkan

Hasil pengujian keamanan disajikan pada Tabel 4.2 berikut ini:

Tabel 4.2 Hasil Pengujian Keamanan (Security Testing)

No	Jenis Serangan	Skenario / Metode Serangan	Hasil yang Diharapkan	Σ Uji	Blokir (Sukses)	Tembus (Gagal)	Hasil (%)	Ket.
1	SQL Injection	Menginput *query* jahat pada kolom login: ' OR '1'='1	Sistem menolak input, tidak ada data bocor, muncul pesan error standar.	10	10	0	100 %	Aman
2	XSS (Cross-Site Scripting)	Menyisipkan skrip berbahaya: <script>alert('Hacked')</script>	Sistem melakukan *sanitization* (skrip tidak dieksekusi browser).	10	10	0	100 %	Aman
3	Brute Force OTP	Mencoba menebak 6 digit OTP secara acak dalam waktu singkat.	Sistem membatasi percobaan (Rate Limiting) atau kode salah terus menerus.	10	10	0	100 %	Aman
4	URL Bypassing	Mengakses halaman /dashboard tanpa login (langsung via URL).	Sistem mendeteksi tidak ada sesi, melempar kembali ke halaman login.	10	10	0	100 %	Aman
5	Session Hijacking	Menggunakan *Session ID* yang sudah *logout* pada browser lain.	Sistem menolak sesi kadaluwarsa, meminta	10	10	0	100 %	Aman

	login ulang.

Berdasarkan Tabel 4.2, sistem menunjukkan tingkat keamanan 100% dalam menangani 50 total skenario serangan (5 jenis x 10 percobaan). Berikut adalah analisis mendalam terhadap hasil tersebut:

a. Pencegahan SQL Injection: Sistem terbukti kebal terhadap serangan injeksi SQL karena penerapan metode Prepared Statements (PDO) pada kode program. Sesuai dengan temuan [15], penggunaan parameter terikat (bound parameters) efektif memisahkan data input pengguna dari perintah SQL, sehingga input berbahaya seperti ' OR '1'='1 hanya dibaca sebagai teks biasa dan tidak dieksekusi oleh basis data.
b. Mitigasi XSS: Pada uji coba penyisipan skrip (XSS), sistem berhasil melakukan filterisasi karakter khusus (*htmlspecialchars*). Hal ini mencegah *browser* menerjemahkan input pengguna sebagai kode program yang dapat dieksekusi.
c. Proteksi Akses Ilegal (URL Bypassing): Pengujian nomor 4 membuktikan bahwa mekanisme manajemen sesi (*session management*) berjalan ketat. Setiap upaya akses langsung ke halaman dasbor tanpa melalui proses otentikasi OTP langsung dialihkan (*redirect*) kembali ke halaman login, memastikan tidak ada celah akses bagi pengguna anonim.

4.3 Analisis dan Pembahasan

Berdasarkan hasil pengujian fungsional dan keamanan yang telah dipaparkan pada sub-bab sebelumnya, tahap selanjutnya adalah melakukan analisis mendalam terhadap mekanisme keamanan yang diterapkan. Analisis ini bertujuan untuk menguraikan efektivitas algoritma HMAC-SHA256 dalam menjamin integritas data serta kemampuan sistem dalam memitigasi serangan siber spesifik seperti pencurian kredensial (credential theft) dan serangan pengulangan (replay attack). Menurut [16], analisis keamanan pasca-implementasi sangat krusial untuk memvalidasi bahwa model kriptografi yang digunakan mampu bertahan terhadap evolusi teknik peretasan modern.

4.3.1 Analisis Kekuatan Algoritma HMAC-SHA256

Algoritma inti yang menjadi fondasi keamanan sistem ini adalah HMAC-SHA256 (*Keyed-Hash Message Authentication Code* dengan *Secure Hash Algorithm* 256-bit). Pemilihan algoritma ini didasarkan pada karakteristiknya yang memiliki resistensi tinggi terhadap benturan (*collision resistance*) dan serangan *pre-image*.

Salah satu indikator kekuatan utama dari algoritma ini adalah efek longsoran (*Avalanche Effect*). Berdasarkan teori kriptografi modern, perubahan kecil pada input—bahkan hanya satu bit—harus menghasilkan perubahan drastis pada output *hash* (message digest).

Dalam konteks sistem ini, kunci rahasia (*secret key*) yang digabungkan dengan variabel waktu (*timestamp*) yang selalu berubah setiap 30 detik menciptakan kombinasi input yang sangat dinamis.

Sebuah studi oleh Nugraha et al. [17] menunjukkan bahwa SHA-256 memiliki tingkat entropi yang jauh lebih tinggi dibandingkan pendahulunya (MD5 atau SHA-1), sehingga menjadikannya standar industri yang belum terpecahkan oleh serangan komputasi konvensional hingga saat ini. Keunggulan ini memastikan bahwa meskipun penyerang berhasil menyadap satu kode OTP, mereka tidak dapat melakukan reverse engineering untuk mengetahui secret key asli pengguna karena sifat fungsi hash yang satu arah (irreversible).

4.3.2 Analisis Mitigasi Pencurian Kredensial dan Anti-Replay Attack

Fokus utama dari pengembangan sistem ini adalah menanggulangi kelemahan otentikasi tradisional (hanya *password*) melalui mekanisme *Two-Factor Authentication* (2FA). Berikut adalah analisis terhadap dua vektor ancaman utama:

a) Mitigasi Pencurian Kredensial (*Credential Theft*)
   Serangan pencurian kredensial, seperti *phishing* atau *keylogging*, seringkali berhasil mendapatkan kombinasi *username* dan *password* pengguna. Namun, dengan implementasi sistem ini, kepemilikan kredensial login saja tidak cukup untuk mendapatkan akses.
   Sistem mewajibkan faktor kedua berupa kode OTP yang dikirimkan secara real-time ke perangkat terpercaya (Trusted Device). Sebagaimana dijelaskan dalam penelitian [2], metode verifikasi berbasis kepemilikan perangkat (possession factor) terbukti mampu menurunkan risiko pembajakan akun hingga 99,9% dibandingkan metode password-only. Penyerang yang memiliki password korban tetap akan tertahan di halaman verifikasi karena tidak memiliki akses fisik ke browser atau perangkat yang menerima notifikasi Web Push.
b) Analisis Anti-Replay Attack
   Serangan *Replay Attack* terjadi ketika penyerang menyadap paket data berisi kode OTP yang valid dan mencoba menggunakannya kembali di waktu yang berbeda. Sistem ini mengatasi ancaman tersebut melalui dua lapisan pertahanan:
   1) Time-Based Validity: Kode OTP hanya valid dalam jendela waktu 30 detik (T_0). Setelah durasi tersebut, algoritma di server akan menghasilkan nilai *hash* baru, sehingga kode lama otomatis menjadi sampah digital yang tidak valid.
   2) One-Time Use constraint: Sistem mencatat status penggunaan token. Jika sebuah kode berhasil digunakan untuk login, statusnya di memori server langsung diubah menjadi *used* (digunakan).

Hasil pengujian pada Tabel 4.3 sebelumnya mengonfirmasi teori yang disampaikan oleh Lestari [18], bahwa integrasi timestamp dan nonce (number used once) pada protokol HMAC adalah solusi paling efektif untuk meniadakan risiko serangan pengulangan paket data. Dengan demikian, integritas proses otentikasi tetap terjaga meskipun berada dalam jaringan yang tidak aman.

KESIMPULAN

Berdasarkan seluruh tahapan penelitian mulai dari perancangan, implementasi, hingga pengujian sistem, dapat disimpulkan bahwa penerapan algoritma HMAC-SHA256 pada fitur otentikasi dua faktor berhasil berjalan sesuai dengan tujuan penelitian. Secara teknis, perhitungan manual yang dilakukan membuktikan bahwa

logika *dynamic truncation* untuk mengambil sampel data 32-bit dari hasil *hash* mampu menghasilkan kode OTP 6 digit yang presisi dan sinkron antara sisi *client* dan *server*. Mekanisme pembangkitan kode ini valid karena selalu menghasilkan kombinasi unik yang bergantung pada kunci rahasia pengguna dan interval waktu server yang berubah setiap 30 detik.

Ditinjau dari aspek fungsionalitas, pengujian *Black Box* yang dilakukan dengan metode iterasi sebanyak 10 kali percobaan pada setiap skenario menunjukkan tingkat keberhasilan mencapai 100%. Seluruh fitur utama, mulai dari proses *login*, pengiriman notifikasi *Web Push*, hingga validasi token pada perangkat terpercaya (*Trusted Device*), berfungsi dengan optimal tanpa ditemukan adanya kegagalan sistem. Hal ini mengindikasikan bahwa integrasi antara *back-end* PHP dan *Service Worker* pada peramban pengguna telah berjalan stabil dalam menangani permintaan otentikasi secara *real-time*.

Selain itu, dari segi keamanan sistem, hasil pengujian keamanan (*Security Testing*) membuktikan bahwa sistem memiliki ketahanan yang solid terhadap berbagai ancaman siber. Sistem terbukti mampu memitigasi serangan *SQL Injection*, *Cross-Site Scripting* (XSS), dan *Brute Force* dengan persentase keberhasilan blokir 100%. Lebih lanjut, penerapan validasi waktu dan mekanisme *nonce* pada token juga efektif dalam mencegah serangan pengulangan (*Replay Attack*), di mana sistem secara otomatis menolak kode OTP yang sudah kadaluwarsa atau yang telah digunakan sebelumnya, sehingga integritas data pengguna tetap terjaga.

Meskipun sistem otentikasi ini telah berhasil dibangun dan memenuhi standar keamanan yang diharapkan, penulis menyadari masih terdapat ruang untuk pengembangan lebih lanjut. Untuk penelitian atau pengembangan di masa mendatang, disarankan agar sistem menambahkan diversifikasi kanal pengiriman kode OTP. Selain menggunakan *Web Push Notification*, integrasi dengan layanan *WhatsApp Gateway* atau *Email* dapat dipertimbangkan sebagai opsi cadangan (*fallback*) untuk mengantisipasi kegagalan pengiriman saat pengguna mengalami gangguan koneksi pada peramban. Selain itu, adopsi standar *WebAuthn* untuk mendukung otentikasi biometrik seperti sidik jari juga dapat diterapkan guna meningkatkan kenyamanan pengguna.

Saran selanjutnya berkaitan dengan peningkatan infrastruktur keamanan. Untuk implementasi pada lingkungan produksi (*live server*), sangat disarankan untuk menerapkan protokol HTTPS (SSL/TLS) secara menyeluruh guna mengenkripsi lalu lintas data antara pengguna dan server, sehingga dapat mencegah potensi serangan *Man-in-the-Middle* (MitM). Terakhir, seiring dengan perkembangan kemampuan komputasi, penelitian selanjutnya dapat mempertimbangkan penggunaan algoritma *hashing* yang lebih kompleks dan memakan memori (memory-hard) seperti Argon2 atau SHA-512 untuk menggantikan SHA-256, guna memberikan lapisan keamanan ekstra terhadap ancaman komputasi masa depan.

DAFTAR PUSTAKA

[1] M. Martiano and Y. Sary, "Cryptography generator for prevention SQL injection attack in big data," Journal of Computer Science, Information Technology and Telecommunication Engineering, vol. 3, no. 2, pp. 292–298, 2022.

[2] E. Wahyudi and B. Santoso, "Efektivitas multi-factor authentication dalam mencegah serangan phishing," Jurnal Keamanan Siber Indonesia, vol. 7, no. 1, pp. 55–65, 2024.

[3] K. Wijaya, "Penerapan algoritma HMAC-SHA256 untuk keamanan transaksi online," Jurnal Cyber Security Indonesia, vol. 4, no. 2, pp. 101–110, 2023.

[4] M. Subramanian, Web Push Notifications: A Complete Guide for Developers. Berkeley, CA, USA: Apress, 2019.

[5] S. Ma, J. Li, H. Kim, E. Bertino, S. Nepal, D. Ostry, and C. Sun, "Fine with '1234'? An analysis of SMS one-time password randomness in Android apps," arXiv, 2021, doi: 10.48550/arXiv.2103.05758.

[6] O. E. A. Mayorga and S. G. Yoo, "One time password (OTP) solution for two factor authentication: A practical case study," Journal of Computer Science, vol. 21, no. 5, pp. 1099–1112, 2025, doi: 10.3844/jcssp.2025.1099.1112.

[7] H. Kim, J. Han, C. Park, and O. Yi, "Analysis of vulnerabilities that can occur when generating one-time password," Applied Sciences, vol. 10, no. 8, Art. no. 2961, 2020, doi: 10.3390/APP10082961.

[8] A. B. Sofian et al., "Enhancing authentication security: Analyzing time-based one-time password systems," International Journal of Computer Technology and Science, vol. 1, no. 3, pp. 56–70, 2024, doi: 10.62951/ijcts.v1i3.25.

[9] M. Zen, I. Irwan, H. Hafni, and M. D. P. Ananda, "Implementasi dan pengujian menggunakan metode blackbox testing pada sistem informasi tracer study," Bulletin of Computer Science Research, vol. 4, no. 4, pp. 327–340, 2024.

[10] A. Pratama and S. Wibowo, "Implementasi metode black box testing pada sistem informasi manajemen berbasis web," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 9, no. 4, pp. 780–788, 2022.

[11] B. Santoso, R. Hartono, and D. Putri, "Analisis keamanan sistem otentikasi menggunakan two-factor authentication," Jurnal Sistem Komputer dan Kecerdasan Buatan, vol. 5, no. 2, pp. 88–95, 2023.

[12] R. Hidayat, "Standarisasi pengujian perangkat lunak pada aplikasi fintech," Jurnal Rekayasa Perangkat Lunak Indonesia, vol. 10, no. 2, pp. 112–120, 2024.

[13] B. Rahardjo and A. Putra, "Evaluasi keamanan website menggunakan metode OWASP top 10," Jurnal Keamanan Siber Indonesia, vol. 6, no. 1, pp. 12–25, 2023.

[14] D. Kurniawan, A. Saputra, and S. Budi, "Analisis vulnerability assessment pada sistem informasi akademik," Jurnal Teknik Informatika dan Sistem Informasi, vol. 9, no. 1, pp. 45–55, 2024.

[15] H. Susanto, "Penerapan prepared statement untuk mencegah SQL injection pada aplikasi e-commerce," Jurnal Algoritma, vol. 19, no. 1, pp. 50–59, 2022.

[16] I. Setiawan and Z. Arifin, "Analisis kriptografi modern untuk keamanan transaksi digital," Jurnal Teknologi Informasi, vol. 15, no. 1, pp. 30–42, 2023.

[17] A. Nugraha, B. Santoso, and K. Wijaya, "Komparasi kinerja algoritma SHA-256 dan MD5 dalam integritas data," Jurnal Sistem Informasi dan Komputer, vol. 7, no. 3, pp. 200–210, 2022.
[18] D. Lestari, "Implementasi Time-Based One-Time Password (TOTP) untuk mencegah replay attack," Jurnal Informatika dan Rekayasa Perangkat Lunak, vol. 3, no. 2, pp. 150–160, 2021.
[19] B. Angkasa, A. Asriyanik, and A. Pambudi, "Implementasi algoritma HMAC-SHA-256 untuk keamanan kemasan produk," Jurnal Ilmiah Universitas Budi Luhur, vol. 20, no. 2, pp. 112–120, 2025.
[20] N. M. Aziz, "Penerapan teknik boundary value analysis dan equivalence partitioning pada pengujian sistem ujian berbasis komputer," Jurnal Informatika dan Teknik Elektro Terapan, vol. 14, no. 1, pp. 45–52, 2026.
[21] H. Hendra, A. Awan, W. Waisen, W. Wilianto, and Y. Yudi, "Memperkuat autentikasi dan integritas data REST-API menggunakan token HMAC SHA-256," Jurnal Minfo Polgan, vol. 13, no. 2, pp. 2189–2197, 2025.
[22] K. C. Laudon and J. P. Laudon, Management Information Systems: Managing the Digital Firm. Pearson, 2020.
[23] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "TOTP: Time-based one-time password algorithm," RFC 6238, Internet Engineering Task Force, 2011, doi: 10.17487/RFC6238.
[24] N. Patel, B. Williams, and E. Johnson, "User perception of notification latency and its impact on application engagement," International Journal of Human-Computer Studies, vol. 144, Art. no. 102498, 2020.
[25] F. C. Ramdani, A. Rahmatulloh, and R. N. Shofa, "Implementasi JSON web token pada authentication dengan algoritma HMAC SHA-256," Jurnal Sistem Informasi (SISTEMASI), vol. 11, no. 1, pp. 15–22, 2022.
[26] A. Sultansyah, A. S. Rahayu, I. Yudiana, and F. Nugraha, "Pengujian black box testing pada fitur permohonan informasi publik melalui website pemerintah Jawa Barat," Jurnal Pengabdian Masyarakat Dan Riset Pendidikan, vol. 3, no. 4, pp. 5912–5919, 2025.