Hello World
Jurnal Ilmu Komputer
https://jurnal.ilmubersama.com/index.php/hello_world

Klik disini untuk menuliskan kategori naskah

Implementasi Keamanan Website Dari Serangan Cross Site Request Forgery Menggunakan Algoritma HMAC-SHA256 Pada Framework Laravel

Ismi Qontas Lubis [1], *Andi Zulherry* [2*]

[1] *Fakultas Ilmu Komputer dan Teknologi Informasi, Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia*
[2] *Fakultas Ilmu Komputer dan Teknologi Informasi, Sains Data, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia*

INFORMASI ARTIKEL

Diterima Redaksi: 00 Januari 00
Revisi Akhir: 00 Februari 00
Diterbitkan *Online*: 00 Maret 00

KATA KUNCI

Cross-Site Request Forgery; CSRF; HMAC-SHA256; Keamanan Web; Laravel; Defense-in-Depth

KORESPONDENSI

Phone: +6289613531255
E-mail: andizulherry@umsu.ac.id

A B S T R A K

Ancaman keamanan seperti Cross-Site Request Forgery (CSRF) menjadi tantangan serius bagi aplikasi web, bahkan yang dibangun dengan framework modern seperti Laravel yang memiliki proteksi bawaan. Penelitian ini bertujuan untuk merancang, mengimplementasikan, dan menganalisis efektivitas algoritma HMAC-SHA256 sebagai lapisan keamanan tambahan untuk memperkuat pertahanan terhadap serangan CSRF pada framework Laravel. Metode penelitian yang digunakan adalah penelitian terapan dengan pendekatan kuantitatif. Pengujian dilakukan menggunakan metode Black Box Testing melalui empat skenario berbeda untuk mengevaluasi sistem tanpa proteksi, fungsionalitas normal, serta efektivitas pertahanan berlapis dan lapisan HMAC secara mandiri. Hasil pengujian menunjukkan bahwa sistem tanpa proteksi sepenuhnya rentan terhadap serangan. Sebaliknya, sistem dengan pertahanan berlapis berhasil menolak serangan, di mana lapisan pertama (token CSRF Laravel) memblokir permintaan dengan respons error 419. Puncak pengujian membuktikan bahwa lapisan HMAC-SHA256 mampu berfungsi sebagai benteng pertahanan mandiri yang efektif, dengan berhasil memblokir serangan (respons error 403) bahkan ketika proteksi bawaan dinonaktifkan, tanpa mengganggu fungsionalitas normal aplikasi. Penelitian ini menyimpulkan bahwa implementasi strategi pertahanan berlapis (Defense-in-Depth) menggunakan HMAC-SHA256 secara signifikan meningkatkan ketahanan aplikasi web terhadap serangan CSRF dan terbukti menjadi mekanisme pertahanan independen yang andal.

PENDAHULUAN

Perkembangan teknologi informasi mendorong lahirnya berbagai sistem berbasis web yang digunakan secara luas oleh instansi pemerintah, perusahaan, maupun masyarakat umum. Penggunaan Website untuk mendukung aktivitas administratif dan transaksi daring telah menjadi kebutuhan utama dalam digitalisasi layanan (Rizki & Ferico, 2021). Namun, kemajuan ini juga diiringi dengan meningkatnya ancaman terhadap keamanan sistem, salah satunya adalah serangan Cross-Site Request Forgery (CSRF).

CSRF merupakan salah satu jenis serangan yang mengeksploitasi kepercayaan web terhadap pengguna yang telah terautentikasi. Dengan memanfaatkan identitas korban, penyerang dapat mengirimkan permintaan yang tampak sah ke server tanpa sepengetahuan pengguna (Kour, 2020; Rankothge & Randeniya, 2020). Serangan ini memungkinkan penyerang melakukan tindakan seperti mengubah sandi, memindahkan dana, atau memodifikasi pengaturan pengguna dengan menyisipkan skrip berbahaya melalui media sosial, email, atau situs yang tidak terpercaya (Rankothge & Randeniya, 2020; Sajjad et al., 2024).

Framework Laravel telah menyediakan sistem proteksi dasar terhadap serangan CSRF melalui penggunaan middleware dan token keamanan. Namun, penelitian menunjukkan bahwa penggunaan token ini belum sepenuhnya efektif, terutama

https://doi.org/10.12345/xxxxx

jika tidak didukung oleh metode otentikasi tambahan yang kuat. Bahkan dengan middleware aktif sekalipun, sistem masih dapat menjadi target eksploitasi apabila token dapat ditebak atau disalahgunakan (Calzavara et al., 2020).

Salah satu pendekatan yang menjanjikan dalam meningkatkan keamanan terhadap serangan semacam ini adalah dengan menerapkan algoritma HMAC-SHA256 (Hash-based Message Authentication Code). Algoritma ini menggabungkan fungsi hash SHA-256 dengan kunci rahasia untuk menghasilkan tanda tangan digital (Angkasa et al., 2023). HMAC telah terbukti andal dalam menjamin integritas data dan autentikasi pesan dalam berbagai skenario sistem keamanan, termasuk cloud computing dan sistem terdistribusi (Herzberg, 2025; Ranganathan & Srinivasan, 2025).

Dalam penelitian oleh (Suhaili et al., 2024), HMAC-SHA256 dapat diimplementasikan secara efisien bahkan pada sistem berbasis perangkat keras dengan performa tinggi. Dalam konteks web, algoritma ini dapat digunakan untuk menghasilkan token yang unik dan tidak dapat dipalsukan, serta mampu memperkuat validasi setiap permintaan yang dikirimkan oleh klien ke server (Ramdani et al., 2023; Rana et al., 2023).

Studi sebelumnya yang mengimplementasikan pendekatan token sinkronisasi (Kour, 2020) memberikan kontribusi penting terhadap pengembangan sistem keamanan, namun masih menyisakan ruang untuk eksplorasi pendekatan yang lebih kriptografis dan dinamis seperti HMAC-SHA256. Oleh karena itu, penelitian ini mengusulkan penerapan algoritma HMAC-SHA256 sebagai lapisan tambahan keamanan terhadap serangan CSRF pada Framework Laravel.

TINJAUAN PUSTAKA

Pengertian Website

Website adalah sekumpulan halaman yang saling terhubung dan dapat diakses melalui jaringan internet. Website digunakan untuk menyajikan informasi dalam bentuk teks, gambar, animasi, suara, atau gabungan dari semua elemen tersebut. Website berfungsi sebagai sarana komunikasi yang memungkinkan pengguna untuk berbagi informasi secara global melalui jaringan internet (Limbong & Sriadhi, 2021; Rizki & Ferico, 2021).

Secara teknis, Website adalah sebuah aplikasi berbasis web yang dapat diakses melalui URL (Uniform Resource Locator) menggunakan browser. Website dapat bersifat statis atau dinamis, tergantung pada bagaimana konten dikelola dan ditampilkan. Website statis umumnya tidak memerlukan interaksi pengguna, sementara Website dinamis memungkinkan pembaruan konten secara otomatis berdasarkan interaksi pengguna (Limbong & Sriadhi, 2021; Rizki & Ferico, 2021).

Cross Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) adalah jenis serangan di mana penyerang memanfaatkan kepercayaan yang diberikan oleh aplikasi web terhadap browser pengguna untuk mengirimkan permintaan yang tidak sah ke situs yang diserang. Serangan ini biasanya dilakukan dengan menyuntikkan link atau script berbahaya ke dalam halaman yang sah yang diakses oleh korban, yang kemudian mengirimkan permintaan yang tidak diinginkan atas nama korban ke situs web yang menjadi target serangan. Tujuan utama dari serangan ini adalah untuk melakukan tindakan yang merugikan pada situs web yang diserang, seperti mengubah data atau mengeksekusi tindakan yang tidak sah (Ashari et al., 2022; Kour, 2020).

Serangan CSRF dapat dilakukan dengan berbagai cara, tergantung pada jenis serangannya. CSRF umumnya dapat diklasifikasikan berdasarkan cara penyerang mengelabui korban untuk mengirimkan permintaan HTTP palsu dan berdasarkan keadaan autentikasi korban pada aplikasi web tepercaya. Dua jenis serangan CSRF yang umum ditemukan adalah Stored CSRF, di mana kode berbahaya disimpan pada server dan dieksekusi ketika korban mengakses halaman tertentu, dan Reflected CSRF, di mana penyerang mengirimkan link berbahaya yang dieksekusi segera setelah korban mengkliknya (Rankothge & Randeniya, 2020).

Dalam situasi di mana serangan CSRF berhasil, pelaku berhasil menciptakan keadaan di mana korban tanpa sadar melakukan tindakan yang tidak diinginkan, seperti mengganti alamat email atau memasukkan data yang tidak sah. Dampak dari serangan ini dapat sangat beragam, tergantung pada hak akses yang dimiliki korban. Dalam beberapa kasus, penyerang dapat memperoleh akses penuh ke akun pengguna, bahkan jika korban memiliki hak istimewa tertentu dalam aplikasi, yang memungkinkan penyerang untuk mengambil alih kontrol atas seluruh data dan fitur aplikasi tersebut (Sajjad et al., 2024).

Algoritma HMAC-SHA256

Dalam dunia keamanan digital, fungsi hash kriptografi merupakan sebuah prosedur matematis yang krusial, berfungsi untuk mengubah data masukan dengan ukuran acak menjadi sebuah string keluaran berukuran tetap yang dikenal sebagai nilai hash. Hasil keluaran ini sering diibaratkan sebagai "sidik jari digital" yang unik untuk data asli, menjadikannya alat yang sangat efektif untuk memverifikasi integritas data (Uriawan et al., 2024). Salah satu karakteristik utamanya adalah sensitivitasnya yang tinggi, di mana perubahan sekecil apa pun pada data asli akan menghasilkan nilai hash yang sama sekali berbeda (Uriawan et al., 2024).

Agar dapat diandalkan untuk tujuan keamanan, fungsi hash kriptografi yang baik harus memiliki beberapa sifat esensial. Sifat pertama adalah satu arah (Preimage Resistance), yang berarti secara komputasi sangat sulit atau bahkan tidak mungkin untuk mendapatkan kembali data masukan asli hanya dengan mengetahui nilai hash-nya, membuat proses ini tidak dapat dibalik (Dewi, 2023). Sifat kedua adalah Second Preimage Resistance, di mana jika sudah diketahui suatu data masukan dan nilai hash-nya, sulit secara komputasi untuk menemukan data masukan lain yang berbeda namun memiliki nilai hash yang sama (Dewi, 2023; Suhaili et al., 2024).

Sifat ketiga, yang sangat penting untuk menjaga integritas data, adalah tahan tumbukan (Collision Resistance), yang berarti harus sulit untuk menemukan dua data masukan yang berbeda ($m_1 \neq m_2$) yang menghasilkan nilai hash yang sama persis ($H(m_1) = H(m_2)$) (Dewi, 2023; Suhaili et al., 2024). Kegagalan dalam menahan tumbukan inilah yang membuat algoritma lama seperti MD5 dan SHA-1 kini dianggap tidak aman untuk aplikasi modern (Angkasa et al., 2023). Oleh karena itu, standar yang lebih baru dan aman seperti keluarga algoritma SHA-2 lebih direkomendasikan (Angkasa et al., 2023; Uriawan et al., 2024).

Framework Laravel

Framework dalam pengembangan aplikasi web adalah kerangka kerja yang menyediakan fungsi, pustaka, dan alat bantu untuk mempercepat proses pengembangan. Penggunaan framework memungkinkan pengembang untuk memanfaatkan struktur dan fungsi yang telah disediakan, tanpa perlu membangun komponen dari awal (Lubis et al., 2022).

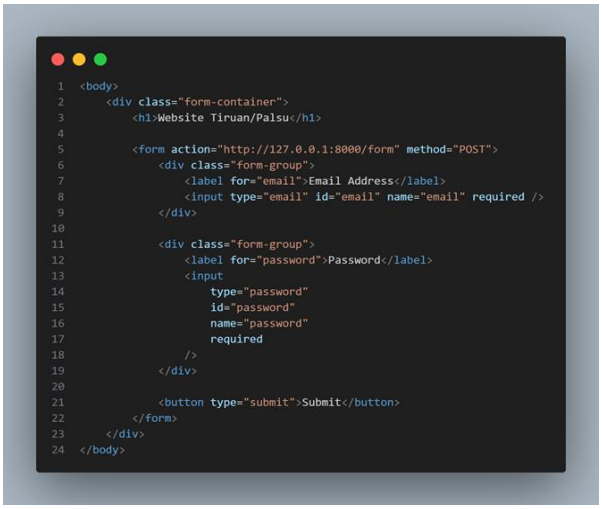
Salah satu bahasa pemrograman yang populer untuk pengembangan aplikasi web sisi server adalah PHP. Untuk mempermudah pengembangan, berbagai framework PHP telah diciptakan, dan salah satu yang paling terkenal adalah Laravel (Jalis et al., 2025; Lubis et al., 2022). Laravel adalah framework open-source yang pertama kali dirilis pada tahun 2011 dan dirancang dengan pola arsitektur Model-View-Controller (MVC), yang memisahkan kode aplikasi menjadi Model (logika data), View (antarmuka pengguna), dan Controller (pengatur alur permintaan) (Jalis et al., 2025; Lubis et al., 2022).

Laravel populer karena sintaksis yang elegan, kemudahan pengelolaan database, fitur keamanan terintegrasi, dan fleksibilitas dalam pengembangan. Framework ini dilengkapi dengan berbagai fitur seperti Eloquent ORM, Blade Templating Engine, routing, middleware, Artisan CLI, dan sistem migrasi database, yang membantu pengembang membangun aplikasi web yang fungsional dan aman (Jalis et al., 2025; Lubis et al., 2022).

METODOLOGI

Halaman Serangan CSRF

Untuk menguji sistem keamanan, dirancang sebuah halaman web statis yang berfungsi sebagai simulasi situs berbahaya. Halaman ini dirancang untuk menipu pengguna dan mencoba mengirimkan permintaan palsu ke aplikasi web asli. Rancangan kode untuk halaman simulasi serangan ini ditunjukkan pada Gambar 1.



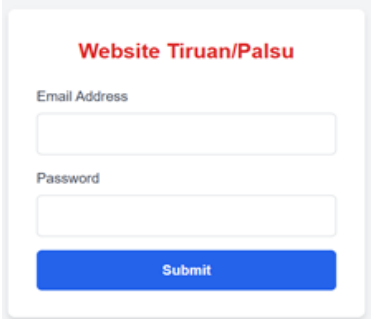
```
<body>
    <div class="form-container">
        <h1>Website Tiruan/Palsu</h1>

        <form action="http://127.0.0.1:8000/form" method="POST">
            <div class="form-group">
                <label for="email">Email Address</label>
                <input type="email" id="email" name="email" required />
            </div>

            <div class="form-group">
                <label for="password">Password</label>
                <input
                    type="password"
                    id="password"
                    name="password"
                    required
                />
            </div>

            <button type="submit">Submit</button>
        </form>
    </div>
</body>
```

Gambar 1. Struktur Halaman Serangan CSRF

Berdasarkan Gambar 1, terdapat beberapa elemen kunci yang sengaja dirancang untuk melancarkan serangan CSRF:

1. Atribut action: Atribut action pada form diarahkan secara langsung ke URL endpoint pemrosesan data pada aplikasi web asli (http://127.0.0.1:8000/form). Hal ini bertujuan untuk mengirimkan permintaan lintas-situs (cross-site).
2. Ketiadaan Token Keamanan: Perbedaan paling krusial adalah halaman ini tidak menyertakan token CSRF bawaan Laravel maupun token kustom HMAC-SHA256. Penyerang tidak memiliki akses ke sesi pengguna yang valid, sehingga tidak dapat menghasilkan token-token tersebut.
3. Antarmuka Penipuan: Tampilan halaman dibuat semirip mungkin dengan halaman input asli untuk menipu pengguna agar secara sukarela memasukkan dan mengirimkan data.

Berdasarkan rancangan kode tersebut, dihasilkan antarmuka pengguna untuk halaman serangan seperti yang ditunjukkan pada Gambar 2.



Gambar 2. Tampilan Halaman Website Tiruan

Controller Dan Function Input Data

Perancangan komponen Controller pada Framework Laravel yang bertanggung jawab untuk menerima dan memproses permintaan yang dikirimkan dari form input, serta mengimplementasikan logika verifikasi keamanan berlapis. Controller berperan sebagai perantara antara View (Halaman Input) dan Model (untuk interaksi database). Perancangan struktur dasar kode Controller yang menunjukkan alur logika penanganan permintaan ini dapat dilihat pada Gambar 3.

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
// Tambahkan 3 baris import di bawah ini
use Illuminate\Support\Facades\Session;
use Illuminate\Support\Facades\Log;
use App\Models\DataInputModel;

class FormController extends Controller
{
    /**
     * Metode untuk menampilkan halaman form.
     */
    public function showForm()
    {
        // 1. Ambil ID Sesi unik pengguna saat ini
        $sessionId = Session::getId();

        // 2. Ambil kunci rahasia dari file config
        $secretKey = config('app.hmac_secret_key');

        // 3. Buat token HMAC-SHA256
        $hmacToken = hash_hmac('sha256', $sessionId, $secretKey);

        // 4. Kirim token tersebut ke view 'form'
        return view('form', ['hmac_token' => $hmacToken]);
    }

    /**
     * Metode untuk memproses data yang dikirim dari form.
     */
    public function processForm(Request $request)
    {
        // --- TAHAP VALIDASI KEAMANAN HMAC-SHA256 ---
        $receivedHmacToken = $request->input('hmac_token');

        // Buat ulang token yang seharusnya di sisi server
        $sessionId = Session::getId();
        $secretKey = config('app.hmac_secret_key');
        $expectedHmacToken = hash_hmac('sha256', $sessionId, $secretKey);

        // Bandingkan token dari form dengan token yang seharusnya
        if (!$receivedHmacToken || !hash_equals($expectedHmacToken, $receivedHmacToken)) {
            // Jika tidak cocok, ini adalah percobaan serangan CSRF. Hentikan!
            Log::warning('CSRF Attack Detected (Invalid HMAC Token)'); // Catat di log
            abort(403, 'Invalid Security Token.');
        }

        // --- JIKA LOLOS, SIMPAN DATA KE DATABASE ---
        try {
            DataInputModel::create([
                'email' => $request->input('email'),
                'password' => bcrypt($request->input('password')) // Selalu hash password
            ]);

            return redirect()->route('form.show')->with('success', 'Data berhasil disimpan dengan aman!');
        } catch (\Exception $e) {
            Log::error('Gagal menyimpan data: ' . $e->getMessage());
            return redirect()->route('form.show')->with('error', 'Terjadi kesalahan pada server.');
        }
    }
}
```

Gambar 3. Struktur Kode Controller

Berdasarkan Gambar 3, FormController memiliki dua metode utama dengan fungsi sebagai berikut:

1. Metode showForm(): Metode ini bertanggung jawab untuk menangani permintaan GET dari pengguna untuk menampilkan halaman form. Di dalam metode ini, sistem secara dinamis menghasilkan token HMAC-SHA256 yang unik berdasarkan ID Sesi pengguna. Token tersebut kemudian dikirimkan ke view untuk disisipkan ke dalam form.
2. Metode processForm(): Metode ini menangani permintaan POST saat pengguna menekan tombol submit. Logika di dalamnya dirancang untuk melakukan validasi keamanan berlapis. Pertama, ia akan memvalidasi token HMAC-SHA256 yang diterima. Jika token valid, proses akan dilanjutkan ke blok try untuk menyimpan data ke database. Jika token tidak ada atau tidak valid, permintaan akan dihentikan dengan respons error 403, yang menandakan adanya percobaan serangan CSRF.

HASIL DAN PEMBAHASAN

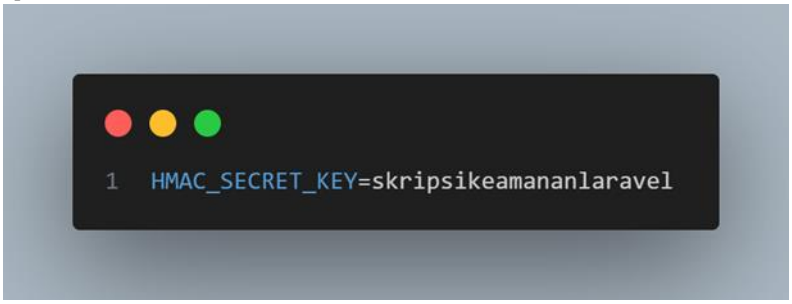
Implementasi Mekanisme Keamanan HMAC-SHA256

Tahapan ini menjelaskan implementasi teknis dari algoritma HMAC-SHA256 sebagai lapisan keamanan tambahan di dalam aplikasi Laravel. Implementasi ini mencakup empat langkah utama: konfigurasi kunci rahasia, pembangkitan token, penyisipan token ke dalam form, dan validasi token di sisi server.

Konfigurasi Kunci Rahasia

Langkah pertama dalam implementasi HMAC adalah mendefinisikan sebuah kunci rahasia (secret key) yang hanya diketahui oleh server. Kunci ini merupakan komponen krusial yang digunakan untuk menghasilkan dan memvalidasi token. Sesuai dengan praktik keamanan modern, kunci ini tidak disimpan langsung di dalam kode, melainkan didefinisikan sebagai variabel lingkungan di dalam file .env untuk mencegah kebocoran pada source code repository.

Baris berikut ditambahkan pada file .env:



Gambar 4. Konfigurasi Kunci Rahasia

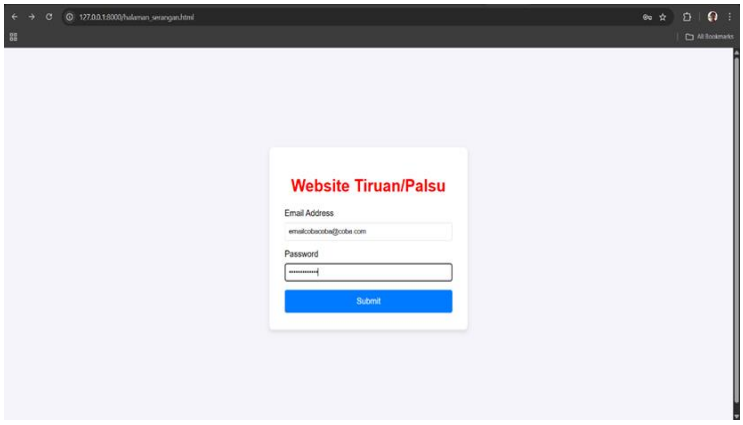
Selanjutnya, untuk memudahkan pemanggilan di dalam aplikasi, kunci ini didaftarkan pada file konfigurasi config/app.php.

Skenario dan Prosedur Pengujian

Pengujian sistem dilakukan untuk mengukur efektivitas implementasi keamanan HMAC-SHA256 dalam menangkal serangan Cross-Site Request Forgery. Metode pengujian yang digunakan adalah Black Box Testing, di mana pengujian berfokus pada evaluasi fungsionalitas sistem (input-output) tanpa memperhatikan struktur kode internalnya. Prosedur pengujian dibagi menjadi empat skenario utama untuk mengevaluasi sistem secara komprehensif, mulai dari kondisi tanpa proteksi hingga kondisi dengan proteksi keamanan berlapis penuh.

Skenario Uji 1 : Serangan Pada Sistem Tanpa Proteksi (Baseline)

Skenario pertama ini bertujuan untuk menetapkan kondisi dasar (baseline) dengan membuktikan bahwa aplikasi web rentan terhadap serangan CSRF jika tidak ada mekanisme keamanan yang aktif. Pada pengujian ini, proteksi CSRF bawaan Laravel dan lapisan keamanan HMAC-SHA256 dinonaktifkan. Serangan kemudian dilancarkan dengan melakukan submit form dari website tiruan, seperti yang ditunjukkan pada Gambar 5.



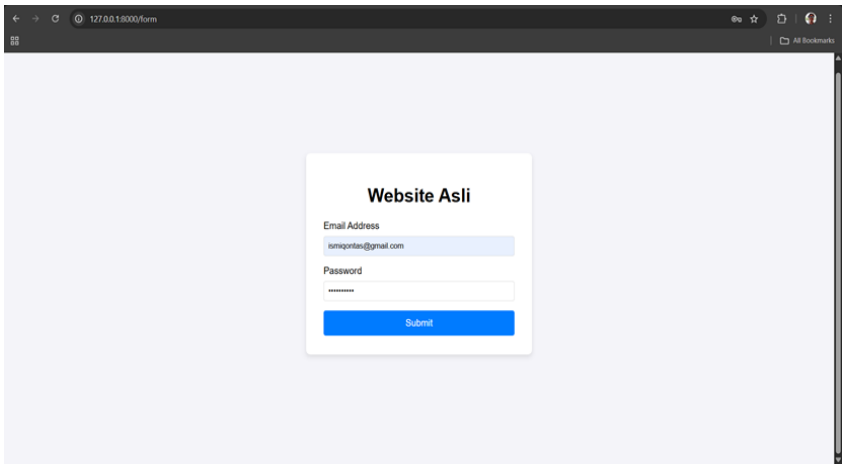
Gambar 5. Tampilan Input Pada Website Tiruan

Setelah form tersebut di-submit, permintaan berhasil diproses oleh server tanpa ada penolakan. Hasilnya, data dari website tiruan tersebut berhasil tersimpan di dalam database.

Hasil dari Skenario Uji 1 secara empiris membuktikan adanya kerentanan kritis pada aplikasi web jika tidak dilengkapi dengan mekanisme proteksi CSRF. Keberhasilan serangan dalam skenario ini menunjukkan bahwa server secara inheren akan memproses permintaan apa pun yang membawa cookie sesi yang valid, terlepas dari mana permintaan itu berasal. Hal ini sejalan dengan konsep dasar serangan CSRF, di mana penyerang mengeksploitasi kepercayaan implisit antara server dan browser pengguna.

Skenario Uji 2 : Fungsionalitas Normal Dengan Proteksi Penuh

Skenario kedua bertujuan untuk memastikan bahwa implementasi keamanan yang ditambahkan tidak mengganggu fungsionalitas normal bagi pengguna yang sah. Pada pengujian ini, seluruh lapisan keamanan, yaitu proteksi CSRF bawaan Laravel dan validasi token HMAC-SHA256, diaktifkan. Pengujian dilakukan dengan mengirimkan data melalui form input pada aplikasi web asli, seperti yang terlihat pada Gambar 6.



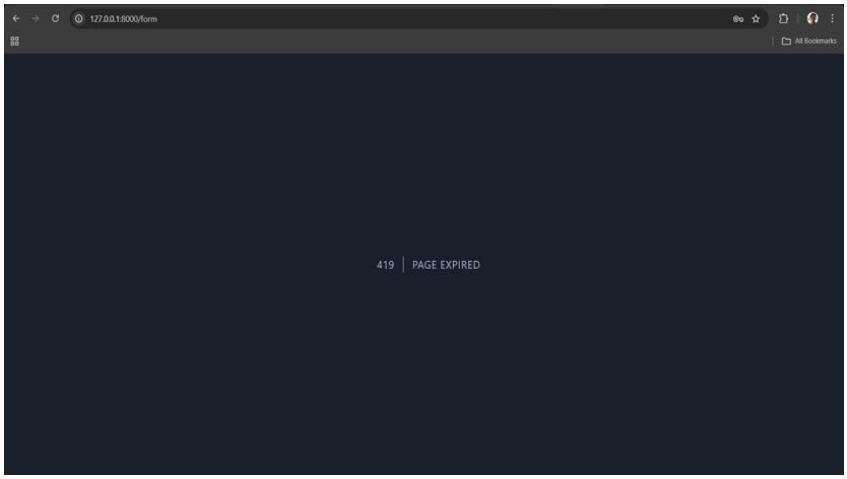
Gambar 6. Tampilan Input Pada Aplikasi Web Asli

Hasil pengujian pada skenario ini menunjukkan bahwa implementasi lapisan keamanan tambahan tidak memberikan dampak negatif pada alur kerja normal aplikasi. Pengguna yang sah tetap dapat mengirimkan data dengan lancar, yang menandakan bahwa sistem keamanan berjalan sesuai yang diharapkan tanpa mengorbankan fungsionalitas utama.

Skenario Uji 2 mengonfirmasi bahwa implementasi lapisan keamanan tambahan tidak mengganggu fungsionalitas normal aplikasi. Data yang dikirim dari pengguna sah melalui form asli berhasil diproses, menandakan bahwa mekanisme keamanan yang dibangun telah terintegrasi dengan baik tanpa menimbulkan dampak negatif pada pengalaman pengguna.

Skenario Uji 3 : Serangan Dengan Deteksi Penuh (Defense-in-Depth)

Skenario ketiga bertujuan untuk menguji efektivitas dari strategi pertahanan berlapis (Defense-in-Depth) saat semua mekanisme keamanan aktif. Pada pengujian ini, proteksi CSRF bawaan Laravel dan validasi token HMAC-SHA256 keduanya diaktifkan. Serangan dilancarkan dengan melakukan submit form dari website tiruan. Hasil dari pengujian ini ditunjukkan pada Gambar 7.



Gambar 7. Hasil Pengujian Skenario 3

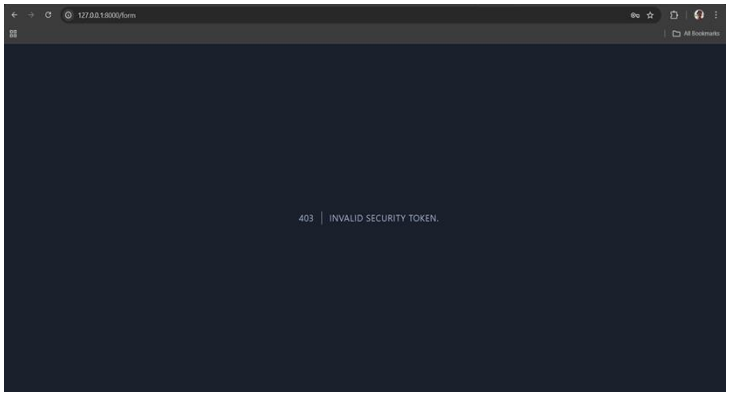
Gambar 7. menunjukkan bahwa permintaan dari website tiruan gagal dan diblokir oleh sistem. Halaman error 419 (Page Expired) yang ditampilkan merupakan respons standar dari framework Laravel ketika token CSRF bawaannya tidak valid atau tidak ada. Hal ini membuktikan bahwa lapisan keamanan pertama dari sistem berhasil mengidentifikasi dan menolak permintaan yang tidak sah tersebut sebelum sempat diproses lebih lanjut oleh lapisan keamanan kedua.

Selanjutnya, Skenario Uji 3 menunjukkan keberhasilan dari strategi pertahanan berlapis (Defense-in-Depth). Permintaan dari situs tiruan berhasil digagalkan oleh lapisan keamanan pertama, yaitu proteksi CSRF bawaan Laravel. Munculnya error 419 (Page Expired)

adalah bukti bahwa mekanisme Synchronizer Token Pattern yang digunakan oleh Laravel efektif dalam mengidentifikasi permintaan tanpa token yang valid.

Skenario Uji 4 : Serangan Dengan Proteksi Lapisan Kedua (HMAC)

Skenario keempat bertujuan untuk mengisolasi dan membuktikan efektivitas lapisan keamanan HMAC-SHA256 secara mandiri. Prosedur pada skenario ini adalah dengan menonaktifkan proteksi CSRF bawaan Laravel, namun tetap mengaktifkan validasi token HMAC yang telah diimplementasikan. Untuk menjalankan Skenario Uji 4, proteksi CSRF bawaan Laravel dinonaktifkan terlebih dahulu melalui konfigurasi middleware pada file bootstrap/app.php. Serangan kembali dilancarkan dengan melakukan submit form dari website tiruan. Hasil dari pengujian ini ditunjukkan pada Gambar 8.



Gambar 8. Hasil Pengujian Skenario 4

Gambar 8 menunjukkan bahwa meskipun lapisan keamanan pertama dinonaktifkan, permintaan dari website tiruan tetap gagal dan berhasil diblokir. Halaman error 403 dengan pesan "Invalid Security Token" yang ditampilkan merupakan respons yang telah didefinisikan secara kustom di dalam FormController. Hal ini membuktikan bahwa lapisan keamanan kedua, yaitu validasi token HMAC-SHA256, mampu berfungsi sebagai pertahanan yang independen dan efektif dalam mengidentifikasi serta menolak permintaan yang dipalsukan.

Puncak dari penelitian ini dibuktikan pada Skenario Uji 4. Dengan menonaktifkan proteksi bawaan Laravel, serangan berhasil melewati lapisan pertahanan pertama. Namun, permintaan tersebut tetap gagal karena berhasil diidentifikasi dan diblokir oleh lapisan keamanan kedua, yaitu validasi token HMAC-SHA256, yang ditandai dengan munculnya error 403. Hasil ini secara langsung menjawab rumusan masalah penelitian, membuktikan bahwa implementasi HMAC-SHA256 tidak hanya berfungsi sebagai lapisan tambahan, tetapi juga sebagai benteng pertahanan mandiri yang efektif. Keberhasilan ini sejalan dengan teori HMAC yang menjamin integritas dan autentikasi permintaan melalui penggunaan kunci rahasia yang tidak dapat dipalsukan oleh penyerang.

Hasil dari Skenario 3 dan 4 secara kuantitatif menunjukkan tingkat keberhasilan 100% dalam memblokir serangan, yang dibuktikan dengan gagalnya permintaan dari situs tiruan pada kedua skenario pengujian tersebut. Secara keseluruhan, hasil pengujian ini menegaskan bahwa penerapan strategi pertahanan berlapis dengan mengombinasikan proteksi CSRF bawaan framework dan mekanisme token HMAC-SHA256 kustom secara signifikan meningkatkan ketahanan aplikasi web terhadap serangan Cross-Site Request Forgery.

Hasil Pengujian

Berdasarkan empat skenario pengujian yang telah dilakukan, hasil dari setiap pengujian dirangkum dalam tabel berikut. Penyajian hasil dalam format tabel ini bertujuan untuk memberikan gambaran yang jelas mengenai efektivitas sistem keamanan yang diimplementasikan dalam berbagai kondisi. Pendekatan ini serupa dengan metode penyajian hasil pengujian yang dilakukan pada penelitian relevan sebelumnya.

Tabel 1. Ringkasan Hasil Pengujian Keamanan

No	Skenario Uji	Keamanan Laravel	Keamanan HMAC	Hasil	Keterangan

1	Serangan (Baseline)	Nonaktif	Nonaktif	Berhasil	Data dari situs tiruan berhasil masuk ke *database*.
2	Penggunaan Normal	Aktif	Aktif	Berhasil	Data dari situs asli berhasil disimpan.
3	Serangan (Lengkap)	Aktif	Aktif	Gagal	Permintaan diblokir oleh Laravel (*Error* 419).
4	Serangan (Lapis 1 Gagal)	Nonaktif	Aktif	Gagal	Permintaan diblokir oleh validasi HMAC (*Error* 403).

Tabel 1. di atas secara ringkas menunjukkan bahwa implementasi keamanan HMAC-SHA256 berhasil memenuhi tujuannya, yaitu memblokir serangan CSRF tanpa mengganggu fungsionalitas normal aplikasi web, dan mampu berfungsi sebagai lapisan pertahanan mandiri.

Berdasarkan hasil pengujian yang telah disajikan, dapat dilakukan analisis mendalam mengenai efektivitas implementasi keamanan HMAC-SHA256. Setiap skenario pengujian memberikan wawasan spesifik terhadap postur keamanan aplikasi web yang dibangun.

KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut: Implementasi algoritma HMAC-SHA256 sebagai lapisan keamanan tambahan pada aplikasi web berbasis Laravel dapat dilakukan secara efektif dengan memanfaatkan ID Sesi pengguna dan kunci rahasia yang tersimpan di server. Hasil pengujian membuktikan bahwa aplikasi web tanpa mekanisme proteksi CSRF yang memadai berada dalam kondisi rentan dan dapat dieksploitasi oleh permintaan yang berasal dari situs eksternal. Penerapan token HMAC-SHA256 terbukti berhasil menjadi lapisan pertahanan mandiri yang mampu mengidentifikasi dan memblokir serangan CSRF, bahkan ketika proteksi bawaan framework dinonaktifkan. Secara keseluruhan, penerapan strategi pertahanan berlapis (Defense-in-Depth) dengan mengombinasikan proteksi CSRF bawaan Laravel dan mekanisme HMAC-SHA256 secara signifikan meningkatkan ketahanan aplikasi web terhadap serangan Cross-Site Request Forgery.

DAFTAR PUSTAKA

Buku

[1] Indah Purnama Sari. Algoritma dan Pemrograman. Medan: UMSU Press, 2023, pp. 290.
[2] Janner Simarmata Arsan Kumala Jaya, Syarifah Fitrah Ramadhani, Niel Ananto, Abdul Karim, Betrisandi, Muhammad Ilham Alhari, Cucut Susanto, Suardinata, Indah Purnama Sari, Edson Yahuda Putra. Komputer dan Masyarakat. Medan: Yayasan Kita Menulis, 2024, pp.162.
[3] Mahdianta Pandia, Indah Purnama Sari, Alexander Wirapraja Fergie Joanda Kaunang, Syarifah Fitrah Ramadhani Stenly Richard Pungus, Sudirman, Suardinata Jimmy Herawan Moedjahedy, Elly Warni, Debby Erce Sondakh. Pengantar Bahasa Pemrograman Python. Medan : Yayasan Kita Menulis, 2024, pp.180
[4] Zelvi Gustiana Arif Dwinanto, Indah Purnama Sari, Janner Simarmata Mahdianta Pandia, Supriadi Syam, Semmy Wellem Taju Fitrah Eka Susilawati, Asmah Akhriana, Rolly Junius Lontaan Fergie Joanda Kaunang. Perkembangan Teknologi Informatika. Medan: Yayasan Kita Menulis, 2024, pp.158
[5] Indah Purnama Sari. Buku Ajar Pemrograman Internet Dasar. Medan: UMSU Press, 2022, pp. 300.
[6] Indah Purnama Sari. Buku Ajar Rekayasa Perangkat Lunak. Medan: UMSU Press, 2021, pp. 228.

Jurnal

[7] Angkasa, B., Asriyanik, & Pambudi, A. (2023). Implementasi Algoritma Hmac-Sha-256 Untuk Keamanan Kemasan Produk Implementation of Hmac-Sha-256 Algorithm for Product Packaging Security. 20(2), 1693–9166.

[8] Ashari, I. F., Oktarina, V., Sadewo, R. G., & Damanhuri, S. (2022). Analysis of Cross Site Request Forgery (CSRF) Attacks on West Lampung Regency Websites Using OWASP ZAP Tools. Jurnal Sisfokom (Sistem Informasi Dan Komputer), 11(2), 276–281. https://doi.org/10.32736/sisfokom.v11i2.1393

[9] Budiman, A., Ahdan, S., & Aziz, M. (2021). Analisis Celah Keamanan Aplikasi Web E-Learning Universitas Abc Dengan Vulnerability Assesment. Jurnal Komputasi, 9(2), 1–10. https://jurnal.fmipa.unila.ac.id/komputasi/article/view/2800

[10] Calzavara, S., Conti, M., Rabitti, R. F. and A., & Tolomei, G. (2020). Machine Learning for Web Vulnerability Detection: The Case of Cross-Site Request Forgery. IEEE Security and Privacy, 18(3), 8–16. https://doi.org/10.1109/MSEC.2019.2961649

[11] Sari, I.P., Jannah, A., Meuraxa, A.M., Syahfitri, A., & Omar, R. (2022). Perancangan Sistem Informasi Penginputan Database Mahasiswa Berbasis Web. Hello World Jurnal Ilmu Komputer 1 (2), 106-110

[12] Satria, A., Ramadhani, F., & Sari, I.P. (2023). Rancang Bangun Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Sekolah Menengah Kejuruan Telkom 2 Medan Menggunakan Codeigniter. Wahana Jurnal Pengabdian kepada Masyarakat 2 (1), 23-31

[13] Sari, I.P., Azzahrah, A., Qathrunada, I.F., Lubis, N., & Anggraini, T. (2022). Perancangan sistem absensi pegawai kantoran secara online pada website berbasis HTML dan CSS. Blend sains jurnal teknik 1 (1), 8-15

[14] Hariani, P.P., Sari, I.P., & Batubara, I.H. (2021). Android-Based Financial Statement Presentation Model. JURNAL TARBIYAH 28 (2), 1-16

[15] Sari, I.P., Syahputra, A., Zaky, N., Sibuea, R.U., & Zakhir, Z. (2022). Perancangan sistem aplikasi penjualan dan layanan jasa laundry sepatu berbasis website. Blend sains jurnal teknik 1 (1), 31-37

[16] Sari, I.P., Al-Khowarizmi, A., & Batubara, I.H. (2021). Cluster Analysis Using K-Means Algorithm and Fuzzy C-Means Clustering For Grouping Students' Abilities In Online Learning Process. Journal of Computer Science, Information Technology and Telecommunication Engineering 2 (1), 139-144

[17] Hutasuhut, B.K., Sari, I.P., & Al-Khowarizmi, A. (2023). Analysis the Effect of Digitalization and Technology on Web-Based Entrepreneurship. Journal of Computer Science, Information Technology and Telecommunication Engineering 4 (1), 350-354

[18] Sari, I.P., Batubara, I. H., & Al-Khowarizmi, A. (2021). Sensitivity Of Obtaining Errors In The Combination Of Fuzzy And Neural Networks For Conducting Student Assessment On E-Learning. International Journal of Economic, Technology and Social Sciences (Injects) 2 (1), 331-338

[19] Sari, I.P., Fahroza, M.F., Mufit, M.I., & Qathrunad, I.F. (2021). Implementation of Dijkstra's Algorithm to Determine the Shortest Route in a City. Journal of Computer Science, Information Technology and Telecommunication Engineering 2 (1), 134-138

[20] Manurung, A.A., Nasution, M.D., & Sari, I.P. (2023). Implementation of Fuzzy K-Nearest Neighbor Method in Dengue Disease Classification. 2023 11th International Conference on Cyber and IT Service Management (CITSM), 1-4

[21] Sari, I.P., Batubara, I.H., Al-Khowarizmi, A., & Hariani, P.P. (2022). Perancangan Sistem Informasi Pengelolaan Arsip Digital Berbasis Web untuk Mengatur Sistem Kearsipan di SMK Tri Karya. Wahana Jurnal Pengabdian kepada Masyarakat 1 (1), 18-24

[22] Sari, I.P., & Batubara, I.H. (2021). Perancangan Sistem Informasi Laporan Keuangan Pada Apotek Menggunakan Algoritma K-NN. Seminar Nasional Teknologi Edukasi dan Humaniora (SiNTESa) (1).

[23] Ramadhani, F., Satria, A., & Sari, I.P. (2023). Implementasi Metode Fuzzy K-Nearest Neighbor dalam Klasifikasi Penyakit Demam Berdarah. Hello World Jurnal Ilmu Komputer 2 (2), 58-62

[24] Sari, I.P., Batubara, I.H., & Basri, M. (2022). Implementasi Internet of Things Berbasis Website dalam Pemesanan Jasa Rumah Service Teknisi Komputer dan Jaringan Komputer. Blend Sains Jurnal Teknik 1 (2), 157-163

[25] Sari, I.P., & Ramadhani, F. (2021). Pengaruh Teknologi Informasi Terhadap Kewirausahaan Pada Aplikasi Perancangan Jual Beli Jamu Berbasis WEB. Prosiding Seminar Nasional Kewirausahaan 2 (1), 874-878

[26] Sari, I.P., Al-Khowarizmi, A., Ramadhani, F., & Sulaiman, O.K. (2023). Implementation of the Selection Sort Algorithm to Sort Data in PHP Programming Language. Journal of Computer Science, Information Technology and Telecommunication Engineering 4 (1), 377-381

[27] Ichsan, A., Al-Khowarizmi, A., & Azhari, M. (2024). Implementation of The Sales and Purchase Program Application Using the Rapid Application Development Model Web Based. Tsabit Journal of Computer Science 1 (1), 27-34

[28] Sari, I.P., & Batubara, I.H. (2021). User Interface Information System for Using Account Services (Joint Account) WEB-Based. International Journal of Economic, Technology and Social Sciences (Injects) 2 (2), 462-469

[29] Ramadhani, F., & Sari, I.P. (2021). Pemanfaatan Aplikasi Online dalam Digitalisasi Pasar Tradisional di Medan. Prosiding Seminar Nasional Kewirausahaan 2 (1), 806-811
[30] Sari, I.P., & Alfarisi, F. (2024). Perancangan Sistem Aplikasi Pendataan Membership Gym Menggunakan Metode Unified Software Development Process (USDP) Berbasis Web. Hello World Jurnal Ilmu Komputer 3 (1), 37-48
[31] Sari, I.P. (2020). Implementasi Pembayaran SPP Berbasis WEB Pada Sekolah Menengah Pertama (SMP) Muhammadiyah Kota Medan. Jurnal Pengabdian Barelang 2 (03), 11-14
[32] Habib, T.A., Azly, R., Irza, M.A., & Prasetya, I. (2024). User Interface Design for the Orca Music Player Mobile Application. Tsabit Journal of Computer Science 1 (1), 18-26
[33] Sari, I.P., Batubara, I.H., Ramadhani, F., & Wardani, S. (2022). Perancangan Sistem Antrian pada Wahana Hiburan dengan Metode First In First Out (FIFO). Sudo Jurnal Teknik Informatika 1 (3), 116-123
[34] Ramadhani, F., Satria, A., & Sari, I.P. (2022). Aplikasi internet berbasis website sebagai E-Commerce penjualan komponen sport car. Blend Sains Jurnal Teknik 1 (2), 69-75
[35] Sari, I.P., Ramadhani, F., Satria, A., Apdilah, D., & Basri, M. (2023). Rancangan UI/UX Aplikasi Analytics pada Toko Online Wao Sneakers Menggunakan Figma Berbasis Mobile. Factory Jurnal Industri, Manajemen dan Rekayasa Sistem Industri 1 (3), 93-101
[36] Sari, I.P., Al-Khowarizmi, A., & Batubara, I.H. (2021). Implementasi Aplikasi Mobile Learning Sistem Manajemen Soal dan Ujian Berbasis Web Pada Platform Android. IHSAN: JURNAL PENGABDIAN MASYARAKAT 3 (2), 178-183
[37] Sari, I.P., & Ramadhani, F. (2021). User Interface Prototype Using User Centered System Design Method in Motorvice Information System. 2021 International Conference on Computer Science and Engineering (IC2SE) 1, 1-6
[38] Ramadhani, F., Sari, I.P., & Satria, A. (2024). Perancangan UI/UX Surat Keterangan Waris dalam Pengembalian Dana Haji Berbasis Web. Blend Sains Jurnal Teknik 2 (3), 198-203
[39] Sari, I.P., Hariani, P.P., Satria, A., & Manurung, A.A. (2023). Rancang Bangun Sistem Informasi Pengelolaan Arsip Materi Ajar Berbasis Web untuk Guru MAS Darul Falah. Wahana Jurnal Pengabdian kepada Masyarakat 2 (2), 59-65
[40] Sari, I.P., Syafii, R., Lubis, D.F.,, Setyadi, A., & Nasution, P. (2022). Pemanfaatan fasilitas google dalam perkuliahan di fakultas teknologi informasi. Blend Sains Jurnal Teknik 1 (2), 107-113
[41] Ramadhani, F., & Sari, I.P. (2021). Improving the Performance of Naïve Bayes Algorithm by Reducing the Attributes of Dataset Using Gain Ratio and Adaboost. 2021 International Conference on Computer Science and Engineering (IC2SE) 1, 1-5
[42] Sari, I.P., Sulaiman, O.K., Al-Khowarizmi, A., & Azhari, M. (2023). Perancangan Sistem Informasi Pelayanan Masyarakat pada Kelurahan Sipagimbar dengan Metode Prototype Berbasis Web. Blend Sains Jurnal Teknik 2 (2), 125-134
[43] Sitompul, D.N., Rahmatika, A., & Sari, I.P. (2023). Application of The Sales and Purchase Program Using The Rapid Application Development Model. Al'adzkiya International of Computer Science and Information Technology (AIoCSIT) Journal 4 (1), 6-16
[44] Sari, I.P., Ramadhani, F., Satria, A., & Apdilah, D. (2023). Implementasi Pengolahan Citra Digital dalam Pengenalan Wajah menggunakan Algoritma PCA dan Viola Jones. Hello World Jurnal Ilmu Komputer 2 (3), 146-157
[45] Sari, I.P., Sulaiman, O.K., Ramadhani, F., & Satria, A. (2023). Perancangan Sistem Manajemen Surat Berbasis Web Pada Kantor Camat Tano Tombangan Angkola. INCODING: Journal of Informatics and Computer Science Engineering 3 (2), 61-76
[46] Guntur, S., Ichsan, A., & Sari, I.P. (2024). Designing a Web-Based Mail Management System at the Beringin Helvetia Sub-district Office. Altafani: Jurnal Pengabdian Masyarakat 1 (1)
[47] Sari, I.P., Al-Khowarizmi, A., Jannah, A., Meuraxa, A.M., & Tanjung, M.I. (2023). Web-Based Offline Game Suit Design: A Model Overview. Journal of Computer Science, Information Technology and Telecommunication Engineering 4 (2), 389-394
[48] Sari, I.P., Al-Khowarizmi, A., Sulaiman, O.K., & Apdilah, D. (2024). System Design for Ordering and Digitizing Website-Based Bus Tickets. Journal of Computer Science, Information Technology and Telecommunication Engineering 5 (1), 543-549
[49] Bisono, A.T., & Zulherry, A. (2025). Analisis Sentimen Game Genshin Impact Untuk Mengetahui Reaksi Dan Harapan Pemain Menggunakan Metode Naïve Bayes. Sudo Jurnal Teknik Informatika 4(2), 183-193
[50] Zulherry, A., Gunawan, T.S., & Wanayumini, W. (2021). Analisis Hasil Pendukung Keputusan Mendapatkan Rumah Dinas Perusahaan Menggunakan Metode Analytical Hierarcy Process (AHP) Dan Teknik For Order Referenci By Similarity (Topsis). Media Informatika Budi Darma 5(2), 695-704

[51] Dewi, D. A. M. K. P. (2023). Analisis Penggunaan HMAC-SHA256 pada Keamanan Aplikasi Chatting. 18220084. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-

[52] Hasanah, F. N., & Untari, R. S. (2020). Buku Ajar Rekayasa Perangkat Lunak. In Buku Ajar Rekayasa Perangkat Lunak. https://doi.org/10.21070/2020/978-623-6833-89-6

[53] Herzberg, A. (2025). Applied Introduction to Cryptography and CyberSecurity. Applied Introduction to Cryptography and CyberSecurity, November 2023. https://doi.org/10.1142/14014

[54] Jalis, Auliana, S., & Permana, B. R. S. (2025). Penerapan Framework Laravel Pada Aplikasi Nilai Siswa Berbasis Web Di Sdn Terumbu Kota Serang. 9(2), 3338–3342.

[55] Kour, P. (2020). A Study On Cross-Site Request Forgery Attack And Its Prevention Measures. InterNational Journal of Advanced Networking and Applications, 12(02), 4561–4566. https://doi.org/10.35444/ijana.2020.12204

[56] Limbong, T., & Sriadhi. (2021). Pemrograman Web Dasar. https://books.google.co.id/books/about/Pemrograman_Web_Dasar.html?id=0pxLDwAAQBAJ&redir_esc=y

[57] Lubis, M. M. M., Tommy, Handoko, D., & Wulan, N. (2022). Analisis Implementasi Laravel 9 Pada Website E-Book Dalam Mengatasi N+1 Problem Serta Penyerangan Csrf dan Xss. Jurnal Ilmu Komputer Dan Sistem Informasi (JIRSI), 2023(2), 173–187. https://jurnal.unityacademy.sch.id/index.php/jirsi/index%0Ahttp://creativecommons.org/licenses/by-sa/4.0/

[58] Ramdani, F. C., Rahmatulloh, A., & Shofa, R. N. (2023). Implementasi JSON Web Token pada Authentication dengan Algoritma HMAC SHA-256. SISTEMASI: Jurnal Sistem Informasi, 12(1), 194–205. http://sistemasi.ftik.unisi.ac.id

[59] Rana, M., Pandey, A., Mishra, A., & Kandu, V. (2023). Enhancing Data Security: A Comprehensive Study on the Efficacy of JSON Web Token (JWT) and HMAC SHA-256 Algorithm for Web Application Security. InterNational Journal on Recent and Innovation Trends in Computing and Communication, 11(9), 4409–4416. https://doi.org/10.17762/ijritcc.v11i9.9930

[60] Ranganathan, C. S., & Srinivasan, C. (2025). Enhanced Cloud Data Security by Employing HMAC for Advanced Cryptographic Protection. 1(1), 1–10.