JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 14 No. 1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v14i1.8284

EVALUASI KEAMANAN SISTEM INFORMASI KEUANGAN SEKOLAH PAUD BERBASIS LARAVEL FILAMENT 3 MENGGUNAKAN *PENETRATION TESTING*

Dawam Agung Pribadi[1], Wiwin Winarti[2]
[1,2] Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang, Jl. Raya Puspitek, Buaran, Kec. Pamulang, Kota Tangerang Selatan, Banten 15310, Telepon: (021) 7412566
Email: [1]dosen02965@unpam.ac.id, [2]dosen02374@unpam.ac.id

Keywords:
Laravel, Filament 3, Penetration Testing, OWASP Top 10, Keamanan Aplikasi Web.

Corespondent Email:
dosen02965@unpam.ac.id

Abstrak. Dalam konteks digitalisasi pendidikan, keamanan data menjadi faktor kritis, terutama untuk melindungi informasi sensitif seperti transaksi keuangan dan data pengguna. Dengan demikian diperlukannya aspek keamanan aplikasi yang perlu diuji secara mendalam, penelitian ini bertujuan untuk menilai tingkat keamanan sistem melalui penerapan metode *penetration testing* yang mengacu pada standar OWASP Top 10. Pengujian dilakukan menggunakan *OWASP Zed Attack Proxy (ZAP) versi 2.16.1* dengan pendekatan *black-box testing*. Hasil pengujian menunjukkan beberapa kerentanan pada level menengah, seperti *Cookie Without Secure Flag*, *Content Security Policy (CSP) Header Not Set*, dan *Missing Anti-clickjacking Header*. Penelitian ini memberikan rekomendasi mitigasi untuk meningkatkan keamanan aplikasi, termasuk konfigurasi ulang header keamanan, penerapan *Secure* dan *HttpOnly flag*, serta penerapan kebijakan CSP.

Abstract. *In the context of educational digitalization, data security becomes a critical factor, particularly in protecting sensitive information such as financial transactions and user data. Therefore, it is essential to conduct an in-depth evaluation of application security aspects. This study aims to assess the system's security level through the implementation of penetration testing based on the OWASP Top 10 standard. The testing was carried out using OWASP Zed Attack Proxy (ZAP) version 2.16.1 with a black-box testing approach. The results revealed several medium-level vulnerabilities, including Cookie Without Secure Flag, Content Security Policy (CSP) Header Not Set, and Missing Anti-clickjacking Header. This research provides mitigation recommendations to enhance application security, such as reconfiguring security headers, implementing Secure and HttpOnly flags, and applying a Content Security Policy (CSP).*

1. PENDAHULUAN

Informasi merupakan salah satu aset paling vital yang harus dijaga dari ancaman maupun akses oleh pihak yang tidak berwenang [1]. Penerapan digitalisasi dalam proses pelaporan bertujuan untuk mempercepat alur kerja, meminimalkan kesalahan dalam pencatatan, serta mempermudah akses terhadap data yang dibutuhkan [2]. Pengelolaan keuangan adalah elemen kunci bagi kelangsungan usaha kecil, namun banyak pelaku usaha kesulitan dalam mencatat dan memantau arus kas mereka [3]. Sistem Informasi Keuangan Sekolah berbasis Laravel dan Filament 3 telah dikembangkan

pada penelitian sebelumnya sebagai solusi pengelolaan data keuangan yang efisien dan transparan.

Pengujian terhadap keamanan sistem informasi perlu dilakukan secara menyeluruh untuk memastikan sistem terlindungi dari berbagai ancaman siber. Keamanan server web memegang peran yang sangat penting dalam mencegah terjadinya perusakan sistem, pencurian data, manipulasi informasi, serta berbagai bentuk penyalahgunaan lainnya[4]. Proses pengelolaan keuangan yang dahulu membutuhkan tatap muka kini dapat diselesaikan dengan lebih cepat dan efisien melalui perangkat digital. Meskipun memberikan kemudahan, transformasi ini juga menghadirkan tantangan baru berupa potensi ancaman terhadap keamanan sistem digital [5].

Aspek keamanan sistem perlu mendapat perhatian khusus karena potensi ancaman dapat muncul dari berbagai celah kerentanan. Ancaman tersebut meliputi manipulasi data masukan, perubahan kode program, modifikasi berkas secara langsung, pencurian data, serta tindakan sabotase yang dapat mengganggu integritas dan ketersediaan sistem. Bahkan, dalam kasus yang lebih serius, serangan dapat mengarah pada penyalahgunaan maupun pencurian sumber daya informasi [6]. Penilaian terhadap potensi kerentanan sistem secara berkala diperlukan guna memastikan keandalan dan keamanan aplikasi tetap terjaga. Penggunaan alat uji otomatis saja dinilai belum memadai, karena hasil yang diperoleh sering kali memerlukan pembuktian atau validasi lanjutan. Salah satu metode validasi yang umum digunakan adalah penetration testing [7]. *Penetration testing* adalah proses pengujian keamanan dengan mensimulasikan serangan terhadap sistem jaringan untuk menemukan celah keamanan dan mengukur daya tahan sistem [8].

Penelitian ini merupakan studi lanjutan yang bertujuan untuk mengidentifikasi, menganalisis, dan mengevaluasi kerentanan keamanan aplikasi web menggunakan pendekatan *penetration testing*. Pada penelitian terdahulu, fokus utama berada pada perancangan dan implementasi fitur fungsional sistem. Namun, aspek keamanan aplikasi belum diuji secara mendalam. Oleh karena itu, penelitian ini bertujuan untuk melakukan evaluasi keamanan sistem menggunakan metode penetration testing berdasarkan standar *OWASP Top 10*. Standar *OWASP Top* 10 digunakan sebagai acuan utama dalam proses evaluasi. Dengan demikian, hasil penelitian ini diharapkan dapat memberikan kontribusi nyata terhadap peningkatan keamanan sistem informasi keuangan sekolah.

2. TINJAUAN PUSTAKA

2.1. Keamanan Sistem Informasi

Keamanan informasi merupakan upaya untuk melindungi data serta sistem informasi dari akses, penggunaan, pengungkapan, perubahan, atau perusakan oleh pihak yang tidak berwenang, dengan tujuan menjaga kerahasiaan, integritas, dan ketersediaan informasi tersebut [9].

2.2. Framework Laravel dan Filament 3

Laravel merupakan framework PHP yang dirancang untuk mempermudah pengembangan aplikasi web melalui dukungan terhadap pola MVC, pemrograman berorientasi objek, serta integrasi dengan berbagai sistem basis data secara fleksibel dan efisien [10]. Laravel mampu meningkatkan efisiensi proses pengembangan aplikasi dengan meminimalkan jumlah kode yang perlu ditulis serta memudahkan integrasi dengan berbagai layanan pihak ketiga. Selain itu, Laravel menyediakan dokumentasi yang lengkap dan mudah dipahami, serta didukung oleh komunitas pengembang yang aktif, sehingga mempermudah proses pemecahan masalah dan pembaruan teknologi yang dihadapi selama pengembangan aplikasi [11]. Filament 3 merupakan *library* antarmuka administrasi yang mempermudah pengelolaan data dan otorisasi pengguna. Keduanya menawarkan struktur yang fleksibel dan mudah dikembangkan, tetapi tetap membutuhkan konfigurasi keamanan yang tepat agar tidak rentan terhadap eksploitasi.

2.3. *Penetration testing*

Penetration testing merupakan metode yang digunakan untuk mensimulasikan teknik atau pendekatan yang biasanya dilakukan oleh pihak tidak berwenang dalam upaya memperoleh akses ilegal ke dalam sistem [12]. *Penetration testing* berfungsi sebagai langkah pencegahan terhadap ancaman peretasan dengan cara mensimulasikan serangan nyata untuk

menemukan dan memperbaiki celah keamanan sistem[13].

2.4. Keamanan Aplikasi Web dan OWASP Top 10

Untuk memperkuat keamanan pada aplikasi berbasis web, perlu dilakukan serangkaian proses pengujian keamanan guna memastikan aplikasi tersebut terlindungi dari berbagai potensi ancaman [14]. OWASP *(Open Web Application Security Project Top 10)* (2021) mendefinisikan sepuluh risiko keamanan paling kritis, di antaranya *Injection*, *Broken Authentication*, *Sensitive Data Exposure*, dan *Security Misconfiguration* [15]. Framework Laravel secara bawaan telah menyediakan fitur perlindungan seperti *CSRF Token* dan *Input Validation*, namun konfigurasi tambahan tetap diperlukan untuk menghadapi serangan yang lebih kompleks.

3. METODE PENELITIAN

3.1. Tahapan Pengujian

Tahapan pengujian dilakukan sebagai berikut:

1. Perencanaan – Menentukan ruang lingkup dan izin pengujian.
2. Pengumpulan Informasi – Melakukan *mapping* terhadap struktur aplikasi.
3. Pemindaian Kerentanan – Menggunakan *OWASP ZAP* untuk mendeteksi risiko keamanan.
4. Eksploitasi Manual – Memverifikasi hasil pemindaian secara manual.
5. Analisis dan Laporan – Mengelompokkan temuan dan memberikan rekomendasi mitigasi.

3.2. Tools yang Digunakan

3.2.1. OWASP ZAP v2.16.1

Digunakan sebagai alat utama untuk melakukan pemindaian dan pengujian keamanan aplikasi berdasarkan standar OWASP Top 10.

3.2.2. Browser Developer Tools

Digunakan untuk memverifikasi hasil pengujian secara manual, khususnya untuk inspeksi cookie, header HTTP, dan elemen DOM.

3.2.3. Laravel Debug Tools

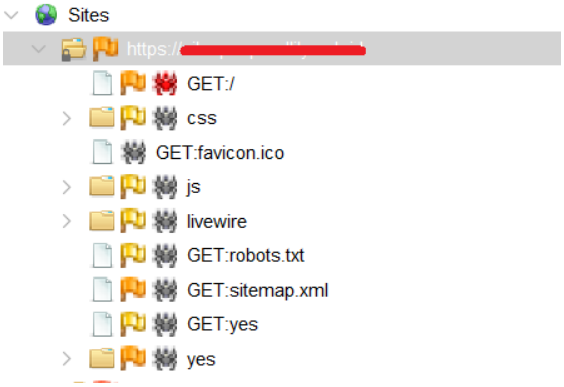
Dimanfaatkan untuk memantau konfigurasi keamanan pada sisi aplikasi selama proses pengujian.

3.3. Skema Pengujian

Berikut flow skema pengujian yang dilakukan

Gambar 1. Merupakan diagram alur proses penelitian yang menggambarkan tahapan-tahapan dalam pengujian kerentanan.

Pada tahap mapping dan spidering, OWASP ZAP melakukan eksplorasi otomatis terhadap seluruh endpoint aplikasi. Dari hasil tersebut, ZAP berhasil menemukan beberapa direktori utama (/css, /js, /livewire) dan file konfigurasi publik (robots.txt, sitemap.xml) yang kemudian digunakan sebagai dasar untuk tahapan vulnerability scanning berikutnya.



Gambar 2. Tampilan hasil spidering OWASP ZAP pada domain pengujian (sumber: hasil uji OWASP ZAP v2.16.1, 2025)

4. HASIL DAN PEMBAHASAN

Berdasarkan hasil pengujian menggunakan OWASP ZAP versi 2.16.1, ditemukan tujuh jenis kerentanan yang termasuk dalam kategori Security Misconfiguration dan Session Management Weakness.

4.1. Ringkasan Hasil Pengujian

Pengujian keamanan dilakukan menggunakan OWASP Zed Attack Proxy (ZAP) v2.16.1 terhadap sistem informasi keuangan sekolah berbasis Laravel Filament 3. Hasil pemindaian menunjukkan adanya tujuh temuan kerentanan yang dikategorikan berdasarkan standar OWASP Top 10. Tabel berikut menyajikan

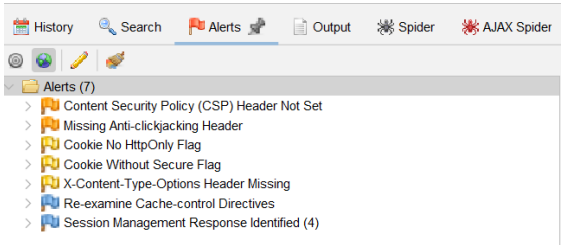
jenis kerentanan, tingkat keparahan (severity), dampak, dan rekomendasi mitigasi yang sesuai.

Tabel 1. Jenis kerentanan, tingkat keparahan (severity), dampak, dan rekomendasi mitigasi.

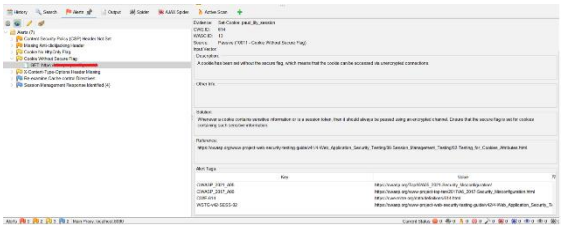
No	Jenis Kerentanan	Kategori OWASP	Severity	Dampak	Rekomendasi
1	**Content Security Policy (CSP) Header Not Set**	A05: Security Misconfiguration	Medium	Potensi *Cross-Site Scripting (XSS)* karena tidak ada pembatasan sumber skrip eksternal	Terapkan header CSP yang ketat dengan whitelist sumber konten terpercaya
2	**Missing Anti-clickjacking Header**	A05: Security Misconfiguration	Medium	Aplikasi dapat ditampilkan dalam frame pihak ketiga, membuka potensi *clickjacking attack*	Tambahkan header X-Frame-Options: DENY atau Content-Security-Policy: frame-ancestors 'none'
3	**Cookie No HttpOnly Flag**	A05: Security Misconfiguration	Medium	Cookie dapat diakses oleh JavaScript, meningkatkan risiko pencurian sesi melalui XSS	Tambahkan flag HttpOnly pada semua cookie sesi
4	**Cookie Without Secure Flag**	A05: Security Misconfiguration	Medium	Cookie dikirim tanpa enkripsi (HTTP), memungkinkan pencurian sesi di jaringan publik	Aktifkan flag Secure dan pastikan semua cookie dikirim melalui HTTPS
5	**X-Content-Type-Options Header Missing**	A05: Security Misconfiguration	Low	Browser dapat menebak tipe konten secara salah, membuka potensi *MIME sniffing attack*	Tambahkan header X-Content-Type-Options: nosniff
6	**Re-examine Cache-Control Directives**	A05: Security Misconfiguration	Low	Data sensitif dapat tersimpan di cache browser	Gunakan Cache-Control: no-store, no-cache, must-revalidate
7	**Session Management Response Identified**	A07: Identification and Authentication Failures	Low	Indikasi adanya sesi yang terbuka tanpa proteksi tambahan	Gunakan rotasi sesi otomatis dan *timeout policy* yang sesuai

4.2. Tampilan Hasil Pengujian OWASP ZAP

Pada gambar 3 memperlihatkan tampilan Alert Summary dari OWASP ZAP yang menampilkan total tujuh kerentanan dengan klasifikasi risiko Medium dan Low. Sedangkan Gambar 4 menunjukkan detail salah satu hasil temuan, yaitu Cookie Without Secure Flag, yang menggambarkan deskripsi kerentanan, solusi, serta referensi dari OWASP.



Gambar 3. Ringkasan hasil pengujian keamanan aplikasi menggunakan OWASP ZAP v2.16.1



Gambar 4. Detail hasil temuan kerentanan "Cookie Without Secure Flag" berdasarkan OWASP ZAP v2.16.1

4.3. Pembahasan

Hasil pengujian menunjukkan bahwa sebagian besar kerentanan berkaitan dengan konfigurasi header keamanan HTTP yang belum optimal. Meskipun tidak ditemukan celah dengan tingkat risiko tinggi (*High Severity*), potensi eksploitasi pada konfigurasi header tetap dapat dimanfaatkan oleh penyerang.

1. Konfigurasi Header Keamanan:
   Kerentanan seperti *Missing Anti-clickjacking Header*, *CSP Header Not Set*, dan *X-Content-Type-Options Header Missing* termasuk dalam kategori *Security*

Misconfiguration. Ini menunjukkan bahwa aplikasi belum mengimplementasikan konfigurasi keamanan HTTP yang direkomendasikan OWASP.

2. Manajemen Cookie:
   Temuan *Cookie Without Secure Flag* dan *Cookie No HttpOnly Flag* menunjukkan bahwa atribut cookie belum diatur untuk mencegah penyalahgunaan sesi. Hal ini dapat diperbaiki dengan menambahkan Secure dan HttpOnly pada cookie yang berisi data sensitif.
3. Cache dan Session Policy:
   Re-examine Cache-Control Directives dan *Session Management Response Identified* memperlihatkan potensi penyimpanan data sensitif di cache serta manajemen sesi yang belum optimal. Mitigasi dapat dilakukan dengan memperbarui konfigurasi Cache-Control dan menambahkan batas waktu sesi (*session timeout*).
4. Perbandingan dengan Penelitian Sebelumnya:
   Pada penelitian sebelumnya, evaluasi keamanan belum dilakukan secara menyeluruh dan hanya mencakup aspek fungsionalitas. Penelitian ini menambahkan dimensi baru berupa evaluasi keamanan sistem dengan hasil konkret berdasarkan alat uji OWASP ZAP, sehingga memberikan gambaran risiko aktual dan langkah mitigasi yang direkomendasikan.

5. KESIMPULAN

Penelitian ini berhasil mengidentifikasi lima kerentanan utama pada sistem informasi keuangan sekolah berbasis Laravel Filament 3, dengan tingkat risiko menengah hingga rendah. Temuan ini menegaskan bahwa meskipun framework Laravel menawarkan keamanan bawaan, konfigurasi tambahan tetap diperlukan. Penelitian lanjutan dapat difokuskan pada implementasi mitigasi dan uji ulang setelah perbaikan.

Saran:

1. Terapkan header keamanan CSP, X-Frame-Options, dan Cache-Control sesuai standar OWASP.
2. Gunakan *flag Secure* dan *HttpOnly* pada semua cookie sesi.
3. Lakukan *penetration testing* berkala setelah pembaruan sistem.
4. Integrasikan keamanan dalam tahap *DevSecOps* pengembangan.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kehadirat Allah SWT atas segala limpahan rahmat, taufik, serta hidayah-Nya, sehingga penulisan karya ini dapat terselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

[1] Abdullah and M. Koprawi, “Analisis Keamanan Website Pada Instansi XYZ Melalui Penetration Testing,” vol. 5, no. 1, pp. 547–555, 2025.

[2] R. A. Putra *et al.*, “IMPLEMENTASI SISTEM PELAPORAN DIGITAL DI BSIP,” vol. 13, no. 2, pp. 871–877, 2025.

[3] Y. F. Baihaqi, T. Sumarni, F. Rusghana, M. Andrie, U. T. Digital, and U. T. Digital, “PERANCANGAN APLIKASI MOBILE KEUANGAN,” vol. 13, no. 3.

[4] A. Fajarino, Y. N. Kunang, H. M. Yudha, E. S. Negara, and N. Rosa, “Evaluasi dan Peningkatan Keamanan Pada Sistem Informasi Akademik Universitas XYZ Palembang,” vol. 7, no. September, pp. 991–1005, 2023.

[5] S. L. Mulyana, “IMPLEMENTASI CYBER SECURITY DALAM SISTEM TRANSAKSI KEUANGAN DIGITAL Aiva,” vol. 2, no. 4, pp. 276–289, 2025.

[6] A. Z. Ifani, N. F. Aspar, A. Dani, and S. Muhammad, “Pengujian Keamanan Sistem Informasi Data Kependudukan Menggunakan Metode Pentetration Testing,” vol. 09, no. 02, pp. 73–78, 2024.

[7] D. F. Priambodo *et al.*, “XYZ Web Penetration Testing Based on OWASP Risk Rating,” vol. 12, no. 1, pp. 33–46, 2023, doi: 10.34148/teknika.v12i1.571.

[8] A. Fatihah and P. Dinarto, “Analisis Keamanan Aplikasi Website Menggunakan Metode Penetration Testing Berdasarkan Framework ISSAF Pada Perusahaan Daerah XYZ,” vol. 4, pp. 4536–4549, 2024.

[9] S. Nurul, S. Anggrainy, and S. Aprelyani, “FAKTOR-FAKTOR YANG MEMPENGARUHI KEAMANAN SISTEM INFORMASI : KEAMANAN INFORMASI , TEKNOLOGI INFORMASI DAN NETWORK (LITERATURE REVIEW SIM),” vol. 3, no. 5, pp. 564–573, 2022.

[10] I. G. Handika and A. Purbasari, “Pemanfaatan Framework Laravel Dalam Pembangunan

Aplikasi E-Travel Berbasis Website," pp. 8–9, 2018.
[11] F. Sinlae, E. Irwanda, Z. Maulana, and V. E. Syahputra, "Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP," vol. 2, no. 2, pp. 119–132, 2024.
[12] A. I. Rafeli, H. B. Seta, and I. W. Widi, "Pengujian Celah Keamanan Menggunakan Metode OWASP Web Security Testing Guide (WSTG) pada Website XYZ," vol. 4221, pp. 97–103, 2022.
[13] S. Hidayatulloh and D. Saptadiaji, "Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP)," pp. 77–86, 2021.
[14] F. Yudha, A. Muhammad, and P. Muryadi, "PERANCANGAN APLIKASI PENGUJIAN CELAH KEAMANAN PADA APLIKASI BERBASIS WEB," vol. 1, no. 1, pp. 1–6, 2018.
[15] OWASP Foundation. (2021). *OWASP Top 10 – The Ten Most Critical Web Application Security Risks (2021 Edition).* https://owasp.org/Top10/2021/