Evaluasi risiko pada sistem manajemen keuangan KSP Mulia Prasama Danarta untuk peningkatan perlindungan informasi

Elia Manuel Putri[1], Andre Kurniawan Pamudji[2], Stephani Inggrit Swastini[3], Ridwan Sanjaya[4]
Universitas Katolik Soegijapranata, Jl. Pawiyatan Luhur Sel. IV No. 1, Bendan Duwur, Kec. Gajahmungkur, Kota Semarang[123]
Email: 24n10006@student.unika.ac.id

Received 16 August 2025; Revised 27 August 2025; Accepted for publication 27 August 2025; Published 26 September 2025

Abstract — *This study aims to identify and analyze security vulnerabilities in web-based financial management applications. Testing was conducted using the active scanning method with the help of OWASP ZAP, which allows detection of security vulnerabilities by sending requests directly to the server. The scan results showed six main findings, namely X-Content-Type-Options Missing, No Cache-control Header, X-Frame-Options Not Set, Cookie Without Secure Flag, XSS Protection Not Enabled, Cross-Domain JS Inclusion. Each finding was analyzed and addressed with mitigation recommendations, such as implementing security headers, using the secure HTTPS protocol, and configuring strict content policies. This study is expected to be a reference in improving the security of similar web applications, especially in preventing header-based attacks and malicious code injection.*

Keywords — *web application security, OWASP ZAP, active scanning, security headers, vulnerability mitigation*

Abstrak—Penelitian ini bertujuan untuk mengidentifikasi dan menganalisis kerentanan keamanan pada aplikasi manajemen keuangan berbasis web. Pengujian dilakukan menggunakan metode *active scanning* dengan bantuan OWASP ZAP, yang memungkinkan pendeteksian celah keamanan melalui pengiriman permintaan langsung ke server. Hasil pemindaian menunjukan enam temuan utama, yaitu *X-Content-Type-Options Missing, No Cache-control Header, X-Frame-Options Not Set, Cookie Without Secure Flag, XSS Protection Not Enabled, Cross-Domain JS Inclusion*. Setiap temuan dianalisis dan diatasi dengan rekomendasi mitigasi, seperti penerapan *security headers*, penggunaan protokol aman HTTPS, dan konfigurasi kebijakan konten yang ketat. Penelitian ini diharapkan menjadi acuan dalam meningkatkan keamanan aplikasi web sejenis, khususnya dalam mencegah serangan berbasis header dan injeksi kode berbahaya.

Kata Kunci—keamanan aplikasi web, OWASP ZAP, *active scanning, security headers,* mitigasi kerentanan

PENDAHULUAN

Perkembangan teknologi informasi yang begitu cepat di era digital ini telah membawa perubahan signifikan terhadap berbagai aspek kehidupan manusia, mencakup kehidupan bidang social, ekonomi, budaya bahkan pola interaksi masyarakat secara keseluruhan. Kemajuan ini tidak hanya mempermudah proses komunikasi jarak jauh dan mempercepat transaksi perdagangan, tetapi juga memfasilitasi pengelolaan, penyimpanan, dan distribusi data dalam skala yang semakin besar dan kompleks[1], [2]. Namun, dibalik manfaat yang luar biasa tersebut, tersimpan pula berbagai risiko yang mengancam, memunculkan tantangan dan permasalahan baru yang semakin sulit untuk diantisipasi maupun diberantas[3]. Ancaman ini meliputi penyebaran malware yang merusak sistem, praktik pembajakan perangkat lunak merugikan secara finansial, hingga serangan siber yang menargetkan layanan digital dengan tujuan mengakses atau merusak data[4], [5]. Fenomena kejahatan siber yang melibatkan pelaku dan korban lintas batas negara telah berkembang menjadi bentuk kejahatan internasional yang kompleks, sistematis, dan sulit dilacak[6]. Kondisi ini membuka peluang bagi individu maupun kelompok tertentu untuk melancarkan serangan yang merugikan secara materi, reputasional, maupun operasional terhadap organisasi atau individu[7], [8].

Salah satu target yang paling rentan terhadap serangan siber adalah aplikasi web[9]. Laporan-laporan keamanan global menunjukan bahwa serangan terhadap aplikasi web mengalami peningkatan signifikan setiap tahun, baik dari segi jumlah maupun kompleksitas teknik yang digunakan[10]. Berbagai metode penyerangan, mulai dari eksploitasi celah keamanan sederhana hingga serangan canggih yang memanfaatkan kerentanan tersembunyi, telah menjadi ancaman nyata bagi pengembang dan pemilik sistem[11]. Oleh karena itu, evaluasi keamanan aplikasi web menjadi kebutuhan mendesak, khususnya melalui proses pengujian sistematis, terstruktur, dan berbasis pada standar yang diakui secara internasional.

Salah satu perangkat yang banyak digunakan dalam pengujian ini adalah OWASP ZAP, sebuah

alat uji keamanan yang dirancang untuk mengidentifikasi berbagai untuk mengidentifikasi berbagai kelemahan aplikasi secara otomatis, berdasarkan kerangka kerja OWASP Top[10], [11]. Dengan semakin meningkatnya ancaman *cross-site scripting*, *clickjacking,* dan *session hijacking,* penerapan praktik keamanan yang tepat, konsisten, dan berkelanjutan menjadi prioritas utama dalam proses pengembangan maupun pemeliharaan aplikasi berbasis web[12].

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menilai tingkat keamanan aplikasi web Manajemen Keuangan yang sedang dikembangkan, dengan memanfaatkan OWASP ZAP sebagai metode utama pengujian[13]. Aplikasi Manajemen Keuangan ini dirancang khusus untuk membantu pengguna, khususnya pelaku usaha dan individu dalam mengelola pemasukan, pengeluaran, catatan hutang-piutang, serta menampilkan laporan keuangan otomatis dalam satu platform. Saat ini, aplikasi Manajemen Keuangan telah mencapai tahap penyempurnaan, di mana sebagian besar fitur inti sudah berjalan dengan baik, namun pengujian keamanan menjadi prioritas sebelum dirilis penuh ke public. Fokus penelitian diarahkan pada proses identifikasi kerentanan yang muncul, disertai penyusunan langkah-langkah mitigasi atau solusi yang relevan dan efektif [14].

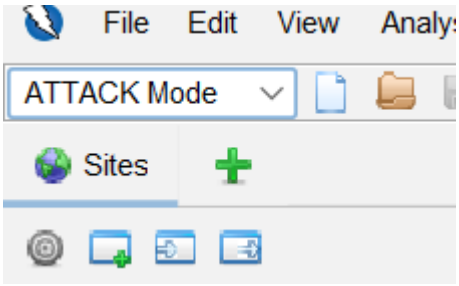
Rumusan masalah yang diangkat meliputi: (1) bagaimana tingkat keamanan aplikasi web diuji secara teknis, (2) kerentanan apa saja yang ditemukan berdasarkan hasil pengujian, serta (3) strategi perbaikan yang selaras dengan standar keamanan modern dan prinsip secure koding . Dengan demikian, penelitian ini tidak hanya berfungsi sebagai dokumentasi hasil pengujian kerentanan, tetapi juga memberikan rekomendasi konkret dalam bentuk penambahan header keamanan, penerapan konfigurasi cookie yang aman, serta pembatasan dan validasi terhadap sumber data eksternal[15]. Implementasi rekomendasi ini diharapkan mampu meningkatkan tingkat keamanan aplikasi secara signifikan, menjaga kerahasiaan dan integritas data keuangan pengguna, serta memperkuat kepercayaan publik terhadap sistem yang didgunakan.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan studi literatur dan pengujian keamanan berbasis OWASP ZAP untuk menganalisis kerentanan aplikasi web. Proses penelitian terdiri dari tiga tahap utama sebagai berikut:

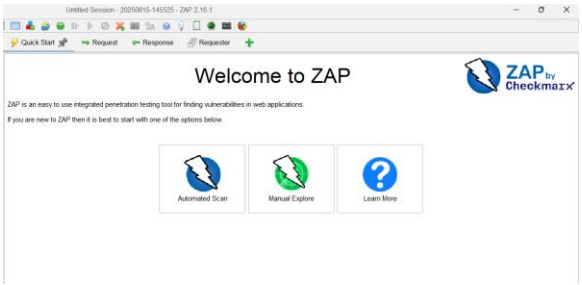
1. Persiapan Pengujian

Pada tahap ini, peneliti menyiapkan lingkungan pengujian menggunakan OWASP ZAP sebagai alat utama untuk mengidentifikasi kerentanan aplikasi. Aplikasi yang diuji adalah sistem manajemen keuangan yang dikembangkan untuk mengelola data transaksi, laporan keuangan, dan perencanaan anggaran. Sebelum pengujian dilakukan, peneliti memastikan bahwa aplikasi berjalan pada server uji untuk menghindari gangguan pada sistem produksi. Terdapat tiga bagian penting dalam pengujian ini. Berikut tiga bagiannya:



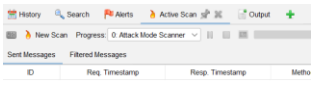
Gambar 1. *ATTACK Mode* untuk *active scan*

Gambar 1 ini merupakan bagian untuk mengatur mode pengecekan atau pengujian kerentanan di aplikasi.



Gambar 2. Bagian pengujian

Gambar 2 merupakan antar muka bagian tempat pengujian, untuk menguji aplikasi.



Gambar 3. Bagian untuk menunjukan hasil *scan*

Gambar 3 ini merupakan bagian untuk menunjukan hasil dari *scan*, semua kerentanan dijelaskan dibagian ini.

Untuk melakukan pemindaian, aplikasi yang akan diuji harus dapat diakses melalui URL yang valid, baik menggunakan localhost untuk pengujian lokal maupun domain publik.

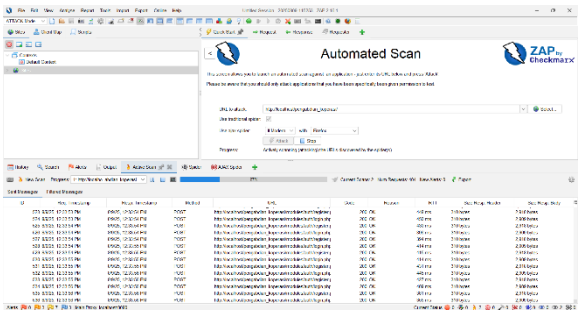
3. Pengujian dan Analisis

Pengujian dilakukan menggunakan metode *active* dibandingkan dengan standar OWASP ZAP Top 10. Setiap kerentanan dianalisis tingkat keparahannya, kemudian diberikan rekomendasi perbaikan berdasarkan prinsip *secure coding.* Proses *scanning* di OWASP ZAP dilakukan dengan dengan memasukkan alamat URL di kolom *URL to Attack* di menu *Quick Start*, lalu memilih opsi *Attack* untuk memulai, kolom itu dibagian yang ada di gambar 2.

HASIL DAN PEMBAHASAN

1. Hasil Pemindaian Keamanan

Gambar 4 menunjukan proses *active scan* yang dilakukan pada aplikasi Manajemen Keuangan yang berjalan di URL http://localhost/pengabdian_koperasi. Proses *scanning* dilakukan untuk memeriksa setiap *endpoint* aplikasi serta menemukan potensi kerentanan yang ada.

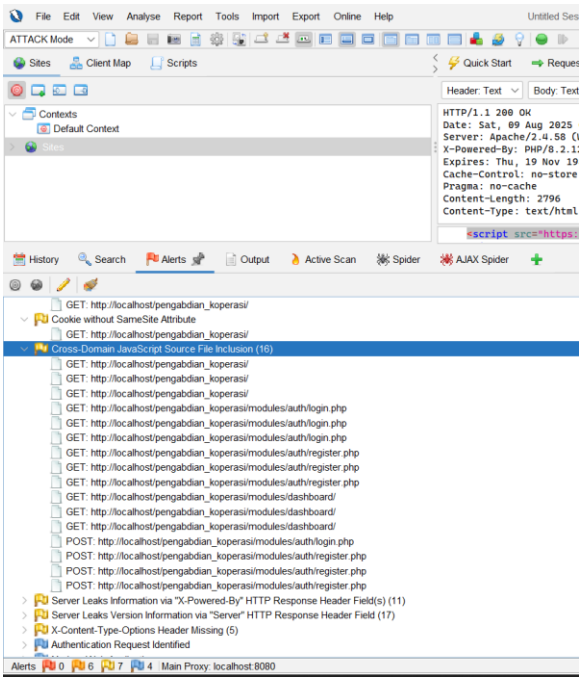


Gambar 4. Proses *scanning* aplikasi Manajemen Keuangan

Setelah proses *active scan* selesai, OWASP ZAP menghasilkan laporan yang memuat daftar kerentanan. Ringkasan temuan ditunjukan pada Tabel 1, sementara tangkapan layar detail hasil pemindaian disajikan pada Gambar 5.

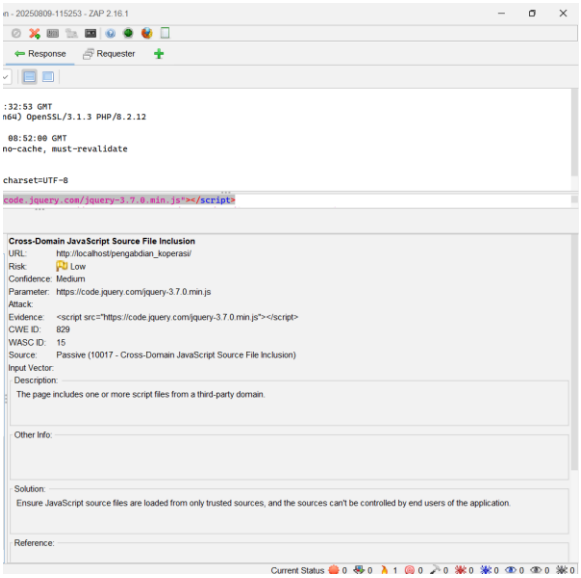
Tabel 1. Ringkasan Hasil Pemindaian Aplikasi Manajemen Keuangan.

Id	Kerentanan	Risiko	Jumlah	URL
10021	*X-Content-Type-Options Missing*	Rendah	6	/auth/login.php
10015	*No Cache-control Header*	Rendah	6	/auth/register.php
10020	*X-Frame-Options Not Set*	Sedang	6	/auth/register.php
10010	*Cookie Without Secure Flag*	Rendah	3	/auth/login.php
10016	*XSS Protection Not Enabled*	Sedang	4	/auth/login.php
10017	*Cross-Domain JS Inclusion*	Tinggi	2	/auth/register.php



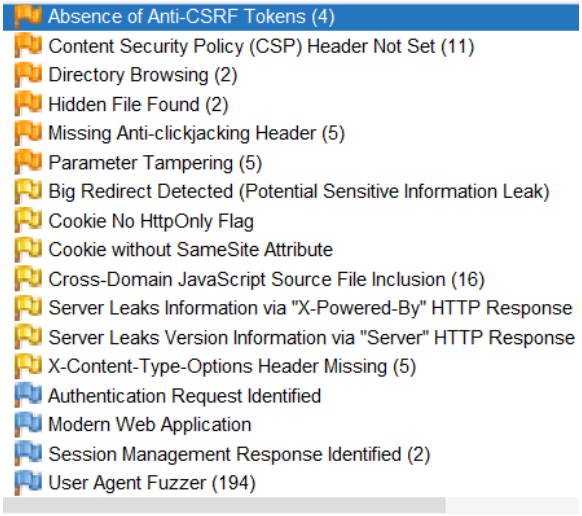
Gambar 5. Hasil *active scan* OWASP ZAP

Gambar 5 merupakan hasil *scan* bagian kiri, berikut gambar 6 yang merupakan hasil scan bagian kanan.



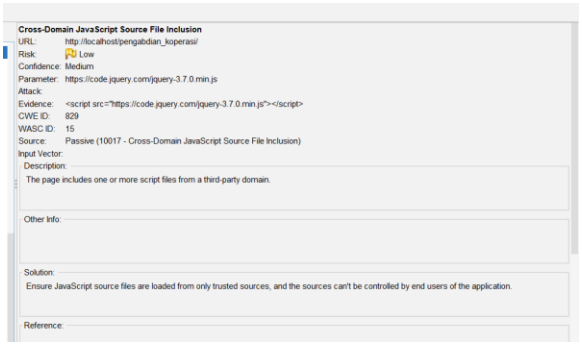
Gambar 6. Hasil *active scan* OWASP ZAP

Untuk gambar yang lebih jelas disajikan pada potongan gambar di gambar 7, 8 dan 9.



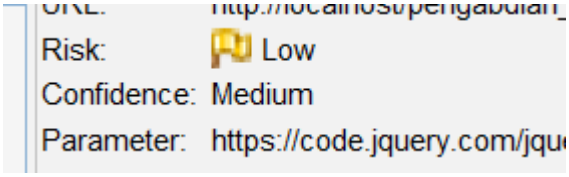
Gambar 6. Kerentanan yang terdeteksi

Gambar 6 menunjukan kerentanan-kerentanan yang terdeteksi dari proses pengujian menggunakan OWASP ZAP ini.



Gambar 7. Penjelasan tiap kerentanan

Gambar 7 merupakan bagian dari penjelasan atau informasi lebih lanjut dari list kerentanan di gambar 6. Pada bagian ini juga terdapat penjelasan tentang tingkat kerentanan yang ada. Di gambar 7 ini, tingkat kerentanannya rendah. Info lebih jelasnya ada di gambar 8



Gambar 8. Detail Jenis Kerentanan

2. Pembahasan

Berdasarkan hasil pada Tabel 1 dan Gambar 5, terdapat beberapa kerentanan pada aplikasi:

a) *X-Content-Type-Options Missing*
Temuan kerentanan ini berarti server tidak mengirimkan header *X-Content-Type-Options: nonsniff*. Tanpa header ini, browser dapat menebak tipe konten, dan dapat dimanfaatkan untuk menyisipkan file berbahaya. Mitigasi yang dilakukan di aplikasi manajemen keuangan ini adalah mengaktifkan header tersebut di konfigurasi server untuk mencegah penebakan tipe konten.

b) *No Cache-control Header*
Kerentanan ini menunjukan bahwa aplikasi belum mengatur kebijakan *caching* pada browser. Hal ini berisiko pada halaman yang menampilkan informasi sensitif seperti data keuangan, karena data dapat tersimpan di *cache* dan terakses oleh pihak lain. Solusi yang tepat yang dilakukan di aplikasi

manajemen keuangan ini adalah menambahkan header *Cache-Control: no store, no-cache, must validate* dan *Pragma: no-cache* untuk memastikan data tidak tersimpan di browser.

c) *X-Frame-Options Not Set*
Kerentanan ini menunjukan tidak adanya proteksi terhadap serangan *clicjacking.* Serangan ini dapat membuat pengguna tanpa sadar mengklik elemen tersembunyi yang merugikan. Mitigasi dari kerentanan ini adalah aplikasi manajemen keuangan ini adalah dengan menambahkan header *X-Frame-Options: DENY* agar halaman tidak bisa di-*embed* oleh situs lain yang berbahaya.

d) *Cookie Without Secure Flag*
Merupakan kerentanan yang dimana *cookie* dikirim tanpa menggunakan atribut *secure*, sehingga berisiko disadap jika data dikirim melalui koneksi non-HTTPS. Solusi yang dilakukan di aplikasi manajemen keuangan ini adalah mengaktifkan atribut *secure* dan *Httponly* pada semua *cookie* sensitif, serta memastikan seluruh komunikasi menggunakan HTTPS.

e) *XSS Protection Not Enabled*
Kerentanan ini aritnya fitur *cross-site scripting* filter bawaan browser tidak diaktifkan. Walaupun bukan satu-satunya pertahanan, fitur ini menambah lapisan proteksi tambahan. Untuk mengatasi hal ini di aplikasi manajemen keuangan mengaktifkan header *X-XSS-Protection: 1; mode=block* untuk menolak halaman jika skrip berbahaya terdeteksi, serta menerapkan sanitasi input yang ketat.

f) *Cross-Domain JS Inclusion*
Kerentanan yang satu ini menunjukan bahwa aplikasi memuat skrip JavaScript dari domain eksternal yang mungkin tidak sepenuhnya terpercaya. Hal ini dapat dimanfaatkan penyerang jika sumber eksternal tersebut di susupi. Solusi yang diterapkan di aplikasi manajemen keuangan ini adalah meminimalkan penggunaan skrip pihak ketiga hanya mengizinkan domain yang benar-benar terpercaya, dan menerapkan *Subresource Integrity (SRI)* untuk memastikan file eksternal tidak dapat diubah oleh pihak berbahaya.

Dengan memperbaiki enam kerentanan ini, keamanan aplikasi manajemen keuangan akan meningkat secara signifikan, mengurangi risiko penyalahgunaan data sensitif pengguna. Selain itu, langkah mitigasi ini akan memberikan kepercayaan lebih kepada pengguna terhadap integritas sistem, meningkatkan kepatuhan terhadap standar keamanan siber yang berlaku, serta memperpanjang umur operasional aplikasi dengan risiko gangguan yang lebih rendah.

KESIMPULAN

Penelitian ini membuktikan bahwa pengujian aplikasi Manajemen Keuangan menggunakan OWASP ZAP efektif untuk mengidentifikasi potensi kerentanan pada aplikasi ini. Hasil evaluasi menunjukan perlunya penerapan langkah mitigasi sesuai prinsip *secure coding* guna memperkuat perlindungan data dan menjaga kepercayaan pengguna. Keamanan aplikasi tidak dapat dicapai sekali saja, melainkan memerlukan pengujian berkala dan pembaruan berkelanjutan seiring berkembangnya teknik serangan siber. Pengembangan selanjutnya disarankan untuk mengintegrasikan pengujian keamanan sejak tahap awal pembuatan aplikasi, memanfaatkan otomatisasi deteksi kerentanan, serta melakukan pelatihan rutin bagi pengembang terkait praktik keamanan terkini.

DAFTAR PUSTAKA

[1] G. Efrianto and N. Tresnawaty, “Pengaruh Privasi, Keamanan, Kepercayaan Dan Pengalaman Terhadap Penggunaan Fintech Di Kalangan Masyarakat Kabupaten Tangerang Banten,” *J. Liabilitas*, vol. 6, no. 1, pp. 53–72, Aug. 2021, doi: 10.54964/liabilitas.v6i1.71.

[2] W. Suryandani, “Pemanfaatan Pembukuan Digital Menggunakan Aplikasi BukuKas untuk Peningkatan Pengelolaan Manajemen Keuangan pada UMKM Kelurahan Sidowayah, Kabupaten Rembang,” *J. Pengabdi. Pada Masy. Indones.*, vol. 1, no. 6, pp. 96–101, 2022, doi: 10.55542/jppmi.v1i6.411.

[3] Y. Daeng *et al.*, “Analisis Penerapan Sistem Keamanan Siber Terhadap Kejahatan Siber Di Indonesia,” *Innov. J. Soc. Sci. Res.*, vol. 3, no. 6, pp. 1135–1145, 2023, [Online]. Available: https://j-innovative.org/index.php/Innovative/article/view/6376

[4] M. Idris, I. Syarif, and I. Winarno, "Web Application Security Education Platform Based on OWASP API Security Project," *Emit. Int. J. Eng. Technol.*, vol. 10, no. 2, pp. 246–261, 2023, doi: 10.24003/emitter.v10i2.705.

[5] M. Aljabri *et al.*, "Testing and Exploiting Tools to Improve OWASP Top Ten Security Vulnerabilities Detection," IEEE, Aug. 2022, pp. 797–803. doi: 10.1109/cicn56167.2022.10008360.

[6] S. Azizah, Z. N. Ula, D. Mutiara, and M. P. Prameswari, "Keamanan siber sebagai fondasi pengembangan aplikasi keuangan mobile: Studi literatur mengenai cybercrime dan mitigasinya," *Akunt. dan Teknol. Inf.*, vol. 17, no. 2, pp. 221–237, 2024, doi: 10.24123/jati.v17i2.6409.

[7] Y. Mulyanto, M. T. A. Zaen, Y. Yuliadi, and S. Sihab, "Analisis Keamanan Website SMA Negeri 2 Sumbawa Besar Menggunakan Metode Penetration Testing (Pentest)," *J. Inf. Syst. Res.*, vol. 4, no. 1, pp. 202–209, Aug. 2022, doi: 10.47065/josh.v4i1.2335.

[8] T. Ariyadi, T. L. Widodo, N. Apriyanti, and F. S. Kirana, "Analisis Kerentanan Keamanan Sistem Informasi Akademik Universitas Bina Darma Menggunakan OWASP," *Techno.Com*, vol. 22, no. 2, pp. 418–429, Aug. 2023, doi: 10.33633/tc.v22i2.7562.

[9] S. K. Lala, A. Kumar, and S. T., "Secure Web development using OWASP Guidelines," IEEE, Aug. 2021. doi: 10.1109/icicccs51141.2021.9432179.

[10] S.-F. Wen and B. Katt, "A quantitative security evaluation and analysis model for web applications based on OWASP application security verification standard," *Comput. & Secur.*, vol. 135, p. 103532, Aug. 2023, doi: 10.1016/j.cose.2023.103532.

[11] A. Jakobsson and I. Häggström, "Study of the techniques used by OWASP ZAP for analysis of vulnerabilities in web applications," 2022. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1675227&dswid=8040

[12] Z. A. Zulfan, S. N. Rahmadiyah, and S. Manullang, "Analisis Keamanan Web Samsat Menggunakan Metode OWASP," *J. Comput. Sci. Informatics Eng.*, vol. 4, no. 1, pp. 21–30, Aug. 2025, doi: 10.55537/cosie.v4i1.987.

[13] G. Pramuja Inngam Fanani, M. A. Mu'min, and N. Tristanti, "Analisis dan Pengujian Kerentanan Website Menggunakan OWASP ZAP," *J. Ris. Sist. dan Teknol. Inf.*, vol. 3, no. 1, pp. 36–50, Aug. 2025, doi: 10.30787/restia.v3i1.1886.

[14] D. Hariyadi and F. E. Nastiti, "Analisis Keamanan Sistem Informasi Menggunakan Sudomy dan OWASP ZAP di Universitas Duta Bangsa Surakarta," *J. Komtika (Komputasi dan Inform.*, vol. 5, no. 1, pp. 35–42, Aug. 2021, doi: 10.31603/komtika.v5i1.5134.

[15] T. R. Arahman, "Evaluasi keamanan manajemen persuratan berbasis website menggunakan framework owasp web security testing guide (WSTG)," 2024. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/76432

PENULIS

Elia Manuel Putri, prodi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata.

Andre Kurniawan Pamudji, prodi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata.

Stephani Inggrit Swastini Dewi, prodi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata.

Ridwan Sanjaya, prodi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata.