PROSIDING Seminar Nasional SISFOTEK ISSN 2597-3584 (Media Online)
Sistem Informasi dan Teknologi
website : www.seminar.iaii.or.id | email : seminar@iaii.or.id

Kajian Kerentanan Keamanan Website Universitas XYZ Menggunakan Framework Open Web Application Security Project (OWASP)

Teguh Reskya Rahmadan[1], Riska Amelia[2], Budi Sunayo[3], Zulfadli[4], Arnita[5]
[1,2,3,4,5] Teknologi Rekayasa Komputer Jaringan, Fakultas Teknologi Industri Universitas Bung Hatta
riskaamelia@bunghatta.ac.id

Abstract

Website security is a critical priority for educational institutions, including XYZ University, to safeguard their data and services from cyber threats. This study aims to assess the security vulnerabilities of XYZ University's website using the Open Web Application Security Project (OWASP) framework. The analysis revealed that the website has high-security risks with multiple vulnerabilities, such as Injection, Cross-Site Scripting (XSS), and Security Misconfiguration. The study employs an OWASP-based approach, encompassing risk identification, vulnerability analysis, and mitigation recommendations. The findings indicate that the primary weaknesses stem from inadequate input data protection, delayed software updates, and weak access controls. This research provides strategic recommendations to mitigate security risks, including strengthening input validation, updating systems, and implementing robust data encryption mechanisms. The results are expected to enhance the security posture of XYZ University's website and serve as a reference for proactive measures against cyber threats.

Keywords*: Security, Website, OWASP.*

Abstrak

Keamanan website merupakan prioritas penting bagi institusi pendidikan, termasuk Universitas XYZ, dalam melindungi data dan layanan dari ancaman siber. Penelitian ini bertujuan untuk mengkaji kerentanan keamanan pada website Universitas XYZ menggunakan framework Open Web Application Security Project (OWASP). Berdasarkan hasil analisis, ditemukan bahwa website tersebut memiliki risiko keamanan yang tinggi dengan berbagai kerentanan, seperti Injection, Cross-Site Scripting (XSS), dan Security Misconfiguration. Kajian dilakukan dengan pendekatan berbasis OWASP, yang mencakup identifikasi risiko, analisis kerentanan, dan rekomendasi mitigasi. Hasil penelitian menunjukkan bahwa kelemahan utama disebabkan oleh kurangnya pengamanan input data, pembaruan perangkat lunak yang tertunda, dan kontrol akses yang lemah. Penelitian ini menghasilkan rekomendasi strategis untuk mengurangi risiko keamanan, termasuk penguatan validasi input, pembaruan sistem, dan penerapan mekanisme enkripsi data. Hasil kajian diharapkan dapat meningkatkan keamanan website Universitas XYZ dan menjadi referensi untuk langkah proaktif dalam menghadapi ancaman siber. Kata kunci: Keamanan, *Website, OWASP.*.

Kata kunci: Security, Website, OWASP.

1. Pendahuluan

Universitas dan Perguruan Tinggi memainkan peran penting dalam mendukung penelitian dan pengembangan serta menyediakan pendidikan berkualitas di Indonesia. Seiring dengan berkembangnya layanan digital, Universitas secara bertahap mengadopsi teknologi website untuk meningkatkan sarana vital yakni menyediakan informasi akademik, mengelola data akademik dan mendukung interaksi antara mahasiswa, dosen, serta pihak administrasi. Namun, dalam proses digitalisasi ini, ancaman keamanan siber juga semakin kompleks dan beragam yang dapat mengakibatkan terjadinya kebocoran dan manipulasi data, dan gangguan layanan yang merugikan berbagai pihak. Institusi akademik menjadi target utama bagi penjahat siber, kegiatan spionase, dan aktivis peretas (hacktivist) karena memiliki sejumlah besar data pribadi yang bersifat sensitif dan penelitian penting [1]. Oleh karena itu, Organisasi perlu melakukan evaluasi dan pengujian keamanan sistem secara berkala guna memastikan dan mendeteksi keretanan dan memahami resiko yang dihadapi [2].

Sektor akademik menempati posisi teratas dalam kasus serangan siber jenis web defacement di Indonesia, dengan kontribusi sebesar 34% dari total 9.749 kasus yang tercatat pada tahun 2020. Posisi ini tetap bertahan pada tahun 2021, di mana sektor ini menyumbang 37% dari total 5.940 serangan web defacement, meskipun terjadi penurunan dalam jumlah kasus secara keseluruhan[2]. Menurut Departemen Teknologi Informasi dan Komunikasi (DICT), Filipina mengalami sekitar 3.000 serangan siber tingkat tinggi dari rentang tahun 2020 hingga 2022, di bulan Juni 2020, tiga puluh (30) sekolah mengalami peretasan [3]. Selain filipina pada Agustus 2024, Singapura juga menghadapi berbagai insiden serangan keamanan siber yang

melibatkan Aplikasi Manajemen Perangkat Mobile Guardian milik kementrian pendidikan singapura dan mengakibatkan terganggunya perangkat pembelajaran pribadi siswa (Personal Learning Devices), setelah sebelumnya pada april 2024 juga terjadi pencurian data pribadi dari 127 sekolah di Singapura [4].

Berdasarkan informasi yang di dapat dari Pusat Teknologi Informasi dan Komunikasi (Pustikom) Universitas XYZ, di bulan Oktober 2024, Aplikasi berbasis Web (biasa disebut portal akademik) juga pernah menjadi target serangan siber. Penyerang berhasil melakukan defacement dengan mengubah tampilan utama portal dan menampilkan pesan yang tidak sesuai. Kejadian ini berdampak negatif terhadap citra profesional universitas dan mengganggu layanan online yang disediakan untuk sivitas akademika.

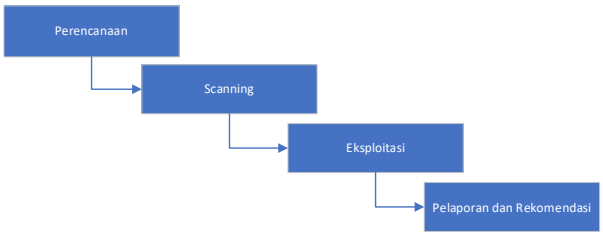
Penyerang sering memanfaatkan kekurangan yang terjadi dalam pengembangan kode program saat membuat aplikasi berbasis web. Dalam kasus ini, kerentanan seperti SQL Injection, Authentication, dan Cross-Site Scripting (XSS) sering dieksploitasi oleh penyerang [5]. XSS terjadi ketika peretas menyuntikkan skrip berbahaya ke dalam laman web yang digunakan oleh pengguna browser. Skrip ini mencakup JavaScript tetapi juga dapat melibatkan bahasa lain seperti HTML atau Flash, yang dapat merusak data pengguna, mencuri informasi sensitif, atau mengendalikan sesi pengguna [6][7][8].

Salah satu metode yang dapat digunakan untuk mengidentifikasi dan mengatasi kerentanan keamanan pada website adalah Open Web Application Security Project (OWASP). OWASP adalah framework open-source yang dirancang untuk meningkatkan keamanan aplikasi berbasis web [9]. Framework ini menawarkan berbagai panduan, alat, dan metodologi yang membantu pengembang dan administrator sistem dalam mengidentifikasi serta memperbaiki kerentanan keamanan pada aplikasi web mereka. Keunggulan OWASP terletak pada sifatnya yang terbuka, fleksibel, dan didukung oleh komunitas global yang terus memperbarui standar keamanan berdasarkan tren ancaman terbaru. Beberapa penelitian menyebutkan bahwa aplikasi OWASP yaitu ZAP membantu mengidentifikasi dan mengatasi kerentanan keamanan secara efektif [10][11].

Oleh karena itu, untuk mengantisipasi terjadinya serangan ulang pada portal akademik Universitas XYZ dan mengingat begitu pentingnya penelitian ini maka peneliti mengangkat judul "Kajian Kerentanan Keamanan Website Universitas XYZ Menggunakan Framework Open Web Application Security Project (OWASP)". Pembahasan ini diharapkan dapat memberikan wawasan dalam meningkatkan keamanan aplikasi web pendidikan.

2. Metode Penelitian

Dalam penelitian ini, Framework yang digunakan adalah Owasp Zap Open Web Application Security Project Zed Attack Proxy (Owasp Zap) dengan Metode Pentest. OWASP ZAP merupakan sebuah alat pengujian yang dapat digunakan untuk menentukan apakah suatu sistem memiliki kerentanan keamanan [12]. OWASP ZAP menawarkan beberapa fitur seperti pemeliharaan sistem, pengujian keamanan, pencarian data ataupun informasi, dan pencarian domain.



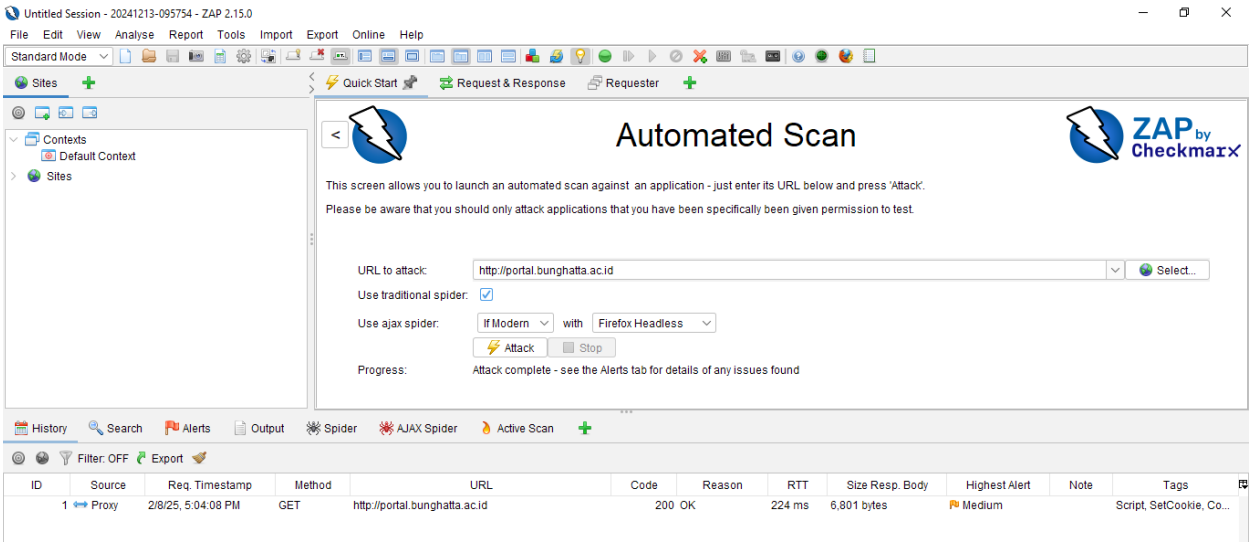
Gambar 1. Tahapan Penelitian

Pelaksanaan peneliti ini memiliki beberapa tahapan yaitu:

1. Tahap Perencanaan,
   Fase isi berisikan langkah – langkah awal yang dilakukan seperti menentukan ruang lingkup dan tujuan pengujian, Mengumpulkan informasi awal website target (URL, domain, fungsionalitas sistem), Mempersiapkan tools dan lingkungan pengujian.
2. Tahap Scanning
   Pada Tahap ini kegiatan yang dilakukan yaitu Melakukan proses scanning / pengujian menggunakan aplikasi OWASP ZAP, Mengidentifikasi celah keamanan yang krusial, dan mencatat semua alert dan warning yang ditemukan.
3. Tahap Eksploitasi
   Pada tahap ini, peneliti melakukan verifikasi terhadap kerentanan yang ditemukan, mengkategorikan tingkat risiko (High, Medium, Low, Informational), dan menganalisis dampak potensial dari setiap kerentanan..
4. Tahap Pelaporan dan Rekomendasi
   Tahap terakhir adalah dokumentasi temuan kerentanan dan penyusunan rekomendasi perbaikan untuk meningkatkan keamanan website

3. Hasil dan Pembahasan

Penelitian ini bertujuan untuk mengkaji kerentanan keamanan website Universitas XYZ menggunakan framework OWASP yaitu dengan menggunakan aplikasi ZAP 2.1.5.0 seperti pada Gambar 2.



Gambar 2. Tampilan Aplikasi Pengujian ZAP

Melalui pendekatan sistematis, penelitian ini berhasil mengidentifikasi berbagai celah keamanan yang berpotensi membahayakan integritas, kerahasiaan, dan ketersediaan sistem website. Hasil scanning dengan menggunakan OWASP ZAP pada Tabel 1.

Tabel 1 merupakan pengelompokkan hasil scanning pada portal akademik Universitas XYZ, dari pengujian tersebut terdapat 19 yang teridentifikasi, dengan tingkat risiko yang bervariasi, yaitu Low, Medium, High, dan Informational.

Tabel 2 menunjukkan distribusi tingkat risiko dari kerentanan yang ditemukan. Dari total 19 kerentanan, 8 kerentanan (42%) memiliki risiko Medium, 7 kerentanan (37%) memiliki risiko Low, dan 4 kerentanan (21%) bersifat Informational. Tidak ditemukan kerentanan dengan risiko High.

Tabel 1. Jenis Scanning

Category	Alert Type	Kerentanan	Count
A05	wildcard directive(content security policy)	Medium	1
A03	script src unsafe inline	Medium	2
A03	style src unsafe inline	Medium	3
A05	content security policy header not set	Medium	4
A01	cross domain misconfigurtion	Medium	5
A01	hidden file found	Medium	7
A02	secure pages include mixed content(including script)	Medium	8
A06	vulnerable JS Library	Medium	1
A02	Cookie No HttpOnly Flag	Low	2
A02	Cookie without Secure Flag	Low	3
A02	Cookie without SameSite Attribute	Low	4
A03	Cross-Domain JavaScript Source File Inclusion	Low	5
A05	Server Leaks Information via "X-Powered-By" HTTP Response Header Field	Low	6
A05	Server Leaks Version Information via "Server" HTTP Response Header Field	Low	7
A02	Strict-Transport-Security Header Not Set	Low	8
A08	Information Disclosure - Suspicious Comments	Informational	1
A04	Modern Web Application	Informational	2
A04	Session Management Response Identified	Informational	3
A10	User Agent Fuzzer	Informational	4

Tabel 2. Alert Chart

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**	**0 (0%)**
	Medium	**0 (0%)**	**4 (21%)**	**3 (16%)**	**1 (5%)**	**8 (42%)**
	Low	**0 (0%)**	**2 (11%)**	**5 (26%)**	**0 (0%)**	**7 (37%)**
	Informational	**0 (0%)**	**0 (0%)**	**4 (21%)**	**0 (0%)**	**4 (21%)**
	Total	**0 0%**	**6 32%**	**12 63%**	**1 5%**	**19 100%**

Tabel 3. Rekomendasi Solusi

Kategori	**Alert**	**Solusi**
A	1.Wildcard directive (content security policy) 2. Script src unsafe inline 3. Style src unsafe inline 4. Content security policy header not set	Content-Security-Policy merupakan header keamanan yang penting untuk mencegah serangan XSS (Cross-Site Scripting) mengontrol sumber daya mana yang diizinkan dimuat oleh browser, membatasi akses ke domain eksternal yang tidak terpercaya, Lakukan konfigurasi ulang pada web server (seperti Apache, Nginx), Application Server (Tomcat, Node.Js), Load balancer (HAProxy, Nginx sebagai load balancer).
B	cross domain misconfigurtion	1. Pastikan bahwa data sensitif tidak tersedia dalam keadaan tanpa autentikasi (misalnya dengan menggunakan white-listing alamat IP). 2. Konfigurasikan header HTTP 'Access-Control-Allow-Origin' ke kumpulan domain yang lebih ketat, atau hapus semua header CORS sepenuhnya, untuk memungkinkan browser web menerapkan Same Origin Policy (SOP) dengan cara yang lebih ketat."
C	Hidden file found	Evaluasi Kebutuhan: 1. Periksa apakah komponen benar-benar dibutuhkan di server produksi Jika tidak dibutuhkan, sebaiknya dimatikan untuk mengurangi risiko keamanan. 2. Jika Komponen Dibutuhkan: a. Implementasikan sistem autentikasi yang tepat b.Terapkan sistem otorisasi untuk mengontrol akses Atau batasi akses dengan cara: - Hanya bisa diakses dari sistem internal - Hanya bisa diakses dari IP tertentu yang sudah diizinkan - Metode pembatasan akses lainnya yang sesuai Tujuan dari rekomendasi ini adalah untuk mengurangi permukaan serangan (attack surface) dengan menghilangkan atau mengamankan komponen yang tidak diperlukan.
D	Secure pages include mixed content(including script)	1. Halaman HTTPS harus: a. Semua konten harus menggunakan protokol HTTPS b. Tidak boleh ada mixed content (campuran HTTPS dan HTTP) c. Termasuk gambar, script, CSS, dan resource lainnya 2. Larangan konten HTTP: a. Tidak boleh ada konten yang dimuat menggunakan HTTP biasa b. Semua resource harus menggunakan HTTPS c. Resource dari domain eksternal juga harus menggunakan HTTPS Contoh: CDN, analytics, font, iklan, dll

Kategori	Alert	Solusi
		Tujuannya adalah untuk memastikan keamanan penuh dalam transmisi data dan mencegah potensi serangan man-in-the-middle. CopyRetryClaude can make mistakes. Please double-check responses.
E	Vulnerable JS Library	Lakukan Update Jquery versi terbaru
F	Cookie No HttpOnly Flag	Pastikan flag HttpOnly diatur untuk semua cookie. Flag HttpOnly adalah pengaturan keamanan untuk cookie yang: a. Mencegah akses cookie melalui JavaScript, b. Melindungi dari serangan XSS (Cross-Site Scripting), c. Hanya memperbolehkan cookie dikirim melalui HTTP request. Tujuan pengaturan ini adalah untuk meningkatkan keamanan cookie dengan mencegah akses tidak sah melalui client-side script.
G	Cookie without Secure Flag	Pastikan flag 'secure' diatur untuk cookie yang berisi informasi sensitif tersebut.
H	Cookie without SameSite Attribute	Pastikan atribut SameSite diatur ke 'lax' atau idealnya 'strict' untuk semua cookie.
I	Cross-Domain JavaScript Source File Inclusion	Pastikan file sumber JavaScript yang dimuat hanya dari sumber-sumber terpercaya, dan sumber-sumber tersebut tidak dapat dikendalikan oleh pengguna akhir aplikasi.
J	Server Leaks Information via "X-Powered-By" HTTP Response Header Field	Server Leaks Information via "X-Powered-By" HTTP Response Header Field
K	Server Leaks Version Information via "Server" HTTP Response Header Field	Server Leaks Version Information via "Server" HTTP Response Header Field
L	Strict-Transport-Security Header Not Set	Pastikan bahwa komponen-komponen server (server web, server aplikasi, load balancer, dll) dikonfigurasi dengan menerapkan Strict-Transport-Security(HSTS). HSTS adalah mekanisme keamanan web yang mengharuskan browser untuk selalu menggunakan koneksi HTTPS yang aman ketika mengakses website. Manfaat mengaktifkan HSTS: a. Mencegah serangan "man-in-the-middle", b. melindungi dari downgrade attacks yang mencoba menurunkan koneksi HTTPS ke HTTP c. Memastikan semua komunikasi antara browser dan server terenkripsi
M	Information Disclosure - Suspicious Comments	Menghapus komentar yang dapat membocorkan informasi dan memperbaiki masalah yang berasal dari komentar
N	Modern Web Application	This is an informational alert and so no changes are required.
O	Session Management Response Identified	Menghapus komentar yang dapat membocorkan informasi dan memperbaiki masalah yang dirujuk dari komentar
P	User Agent Fuzzer	1. Temukan potensi kerentanan keamanan jika aplikasi memberikan respons yang sangat berbeda berdasarkan User-Agent, 2. identifikasi apakah ada konten sensitif yang mungkin bocor ketika menggunakan User-Agent tertentu, 3. Pastikan kontrol akses berfungsi dengan konsisten terlepas dari User-Agent yang digunakan. Jika ditemukan perbedaan respons yang signifikan, ini bisa menjadi indikasi masalah keamanan yang perlu diselidiki lebih lanjut, terutama jika perbedaan tersebut membuka akses ke informasi atau fungsi yang seharusnya dibatasi.

4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa website Universitas XYZ memiliki beberapa kerentanan keamanan yang perlu segera ditangani. Meskipun tidak ditemukan kerentanan dengan risiko High, terdapat 8 kerentanan dengan risiko Medium yang berpotensi membahayakan integritas dan kerahasiaan data. Kerentanan tersebut terutama terkait dengan Content Security Policy (CSP), Cross-Domain Misconfiguration, dan Mixed Content.

Rekomendasi yang diberikan, seperti memperbarui library JavaScript, menerapkan flag keamanan pada cookie, dan mengonfigurasi ulang server, diharapkan dapat meningkatkan keamanan website Universitas XYZ. Penelitian ini juga menunjukkan pentingnya penggunaan framework OWASP dan alat seperti OWASP ZAP dalam mengidentifikasi dan memitigasi kerentanan keamanan pada aplikasi web.

Untuk penelitian selanjutnya, disarankan untuk melakukan pengujian keamanan secara berkala dan memantau perkembangan ancaman siber yang terus berkembang. Selain itu, pelatihan dan kesadaran keamanan bagi pengembang dan administrator sistem

juga perlu ditingkatkan untuk memastikan implementasi keamanan yang lebih baik dan mencegah potensi serangan siber di masa mendatang

Daftar Rujukan

[1] N. Siphambili, "Exploring Cybersecurity Implications in Higher Education," *Eur. Conf. Cyber Warf. Secur.*, vol. 23, no. 1, pp. 526–531, 2024, doi: 10.34190/eccws.23.1.2306.

[2] Y. K. Heong and Y. S. Teng, "Digital Security Risk Disclosure and Investment Process," vol. 14, no. 3, pp. 504–523, 2024, doi: 10.6007/IJARAFMS/v14-i3/22643.

[3] Carlos P. Flores Jr., "Evaluation of Common Security Vulnerabilities of State Universities and Colleges Websites Based on OWASP," *J. Electr. Syst.*, vol. 20, no. 5s, pp. 1396–1404, 2024, doi: 10.52783/jes.2471.

[4] S. Watini, G. Davies, and N. Andersen, "Cybersecurity in Learning Systems : Data protection and privacy in educational information systems and digital learning environments," vol. 3, no. 1, pp. 26–35, 2024.

[5] O. Access, "Understanding the Threat: Exploring SQL Injection Attacks and Prevention Strategies," *Int. Res. J. Mod. Eng. Technol. Sci.*, no. 05, pp. 7112–7116, 2024, doi: 10.56726/irjmets57129.

[6] R. M. Wibowo and A. Sulaksono, "Web Vulnerability Through Cross Site Scripting (XSS) Detection with OWASP Security Shepherd," *Indones. J. Inf. Syst.*, vol. 3, no. 2, pp. 149–159, 2021, doi: 10.24002/ijis.v3i2.4192.

[7] G. Kusuma, "Implementasi Owasp Zap Untuk Pengujian Keamanan Sistem Informasi Akademik," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 16, no. 2, pp. 178–186, 2022, doi: 10.47111/jti.v16i2.3995.

[8] O. C. Abikoye, A. Abubakar, A. H. Dokoro, O. N. Akande, and A. A. Kayode, "A novel technique to prevent SQL injection and cross-site scripting attacks using Knuth-Morris-Pratt string match algorithm," *Eurasip J. Inf. Secur.*, vol. 2020, no. 1, 2020, doi: 10.1186/s13635-020-00113-y.

[9] H. Hermanto and H. Haeruddin, "Peningkatan Sistem Keamanan Website Menggunakan Metode OWASP," *J. Ilmu Komput. dan Bisnis*, vol. 13, no. 1, pp. 94–104, 2022, doi: 10.47927/jikb.v13i1.277.

[10] E. Nurelasari and D. Gumilang Al Farabi, "Analisis Keamanan Sistem Website Menggunakan Metode Open Web Application Security Project (Owasp) Pada Simantep.Id," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3049–3054, 2024, doi: 10.36040/jati.v8i3.9314.

[11] D. Supriadi, E. Suryadi, R. Muslim, L. D. Samsumar, and U. T. Mataram, "IMPLEMENTASI VULNERABILITY ASSESSMENT OWASP (OPEN WEB APPLICATION SECURITY PROJECT) PADA WEBSITE," vol. 1, no. 4, pp. 232–240, 2024.

[12] G. Angga Septiawan, K. W. S. Irawan, I. Mayasari, and I. M. E. Listartha, "Analisis Kerentanan XSS dan Rate Limiting Pada Website SMAN 8 Denpasar Menggunakan Framework OWASP ZAP," *J. Inform. Upgris*, vol. 8, no. 1, pp. 6–8, 2022, doi: 10.26877/jiu.v8i1.10271.