Published online on the page: https://jurnal.mifandimandiri.com/index.php/aicom

A I C O M
Artificial Intelligence and Computing
| ISSN (*Online*) 3064-0814 |

Analisis *Vulnerability Assessment* Menggunakan OWASP ZAP untuk Mengidentifikasi Celah Keamanan Website

Permadi Kusuma[1,*], *Ryan Alghazali Pakkaja*[2]

[1]*Universitas Cokroaminoto Palopo, Palopo,* Indonesia
[2]*Politeknik Dewantara, Palopo,* Indonesia

Informasi Artikel

Sejarah Artikel:
Submit: 04 Maret 2026
Revisi: 16 Maret 2026
Diterima: 25 Maret 2026
Diterbitkan: 30 Maret 2026

Kata Kunci

Keamanan Website, Vulnerability, OWASP ZAP, Cyber Security, OWASP Top 10

Correspondence

E-mail: permadikusumauncp@gmail.com*

A B S T R A K

Website sekolah berperan penting sebagai media penyampaian informasi akademik dan administrasi kepada siswa, guru, serta masyarakat. Namun, banyak website sekolah masih memiliki potensi kerentanan keamanan yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab. Penelitian ini bertujuan untuk menganalisis tingkat keamanan website sekolah melalui proses *vulnerability assessment* guna mengidentifikasi potensi celah keamanan pada aplikasi web. Penelitian ini menggunakan sampel tiga website sekolah, yaitu SMA Negeri 1 Palopo, SMK Negeri 1 Palopo, dan SMP Negeri 1 Tomoni yang berada di wilayah Kota Palopo dan Kabupaten Luwu Timur. Proses analisis kerentanan dilakukan menggunakan perangkat lunak OWASP ZAP (*Open Web Application Security Project Zed Attack Proxy*) yang digunakan untuk melakukan pemindaian terhadap potensi kerentanan keamanan pada website. Metode penelitian meliputi tahap pengumpulan sampel website, proses pemindaian kerentanan, analisis hasil pemindaian, serta pengelompokan tingkat risiko kerentanan yang ditemukan. Hasil penelitian menunjukkan bahwa ketiga website sekolah masih memiliki potensi kerentanan keamanan dengan tingkat risiko yang berbeda, seperti kesalahan konfigurasi keamanan, potensi serangan cross-site scripting, dan paparan informasi sensitif. Klasifikasi tingkat risiko pada penelitian ini mengacu pada standar bawaan OWASP ZAP, yaitu *High*, *Medium*, *Low*, dan *Informational*. Temuan tersebut menunjukkan bahwa keamanan website sekolah masih perlu ditingkatkan melalui penerapan konfigurasi keamanan yang lebih baik, validasi input yang tepat, serta pembaruan sistem secara berkala.

Abstract

School websites play a vital role as a medium for conveying academic and administrative information to students, teachers, and the public. However, many school websites still have potential security vulnerabilities that can be exploited by malicious actors. This study aims to analyze the security level of school websites through a vulnerability assessment process to identify potential security gaps in web applications. This study uses a sample of three school websites, namely SMA Negeri 1 Palopo, SMK Negeri 1 Palopo, and SMP Negeri 1 Tomoni, located in the city of Palopo and East Luwu Regency. The vulnerability analysis process was conducted using OWASP ZAP (Open Web Application Security Project Zed Attack Proxy) software, which was used to scan for potential security vulnerabilities on the websites. The research methodology included the collection of website samples, the vulnerability scanning process, the analysis of scan results, and the categorization of the risk levels of the identified vulnerabilities. The results of the study indicate that all three school websites still have potential security vulnerabilities with varying risk levels, such as security configuration errors, potential cross-site scripting attacks, and exposure of sensitive information. The risk level classification in this study refers to the built-in OWASP ZAP standards, namely High, Medium, Low, and Informational. These findings indicate that school website security still needs to be improved through the implementation of better security configurations, proper input validation, and regular system updates.

Published online on the page: https://jurnal.mifandimandiri.com/index.php/aicom

A I C O M
Artificial Intelligence and Computing
| ISSN (*Online*) 3064-0814 |

1. Pendahuluan

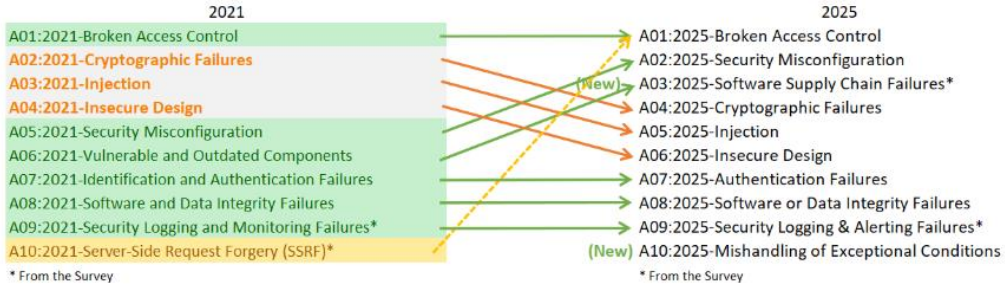
Perkembangan teknologi informasi merupakan faktor penting bagi kemajuan jaman. Ada beberapa bidang menjadi kunci kemajuan teknologi yang mempengaruhi tingkat kemajuan dalam negara tersebut diantaranya bidang Pendidikan, bidang Ekonomi, bidang Kesehatan, bidang Pemerintahan, dan bidang Sosial Budaya [1]. Ancaman nyata yang menyerang sistem informasi seperti kebocoran data, kredensial (kompromi akun), *phishing*, serangan, berbasis web, serangan *malware*, *cracking* (pembajakan), *carding* (transaksi ilegal) dan sebagainya [2]. Perangkat lunak apa pun yang berjalan dalam suatu sistem berpotensi dieksploitasi menggunakan kerentanan dalam perangkat lunak, dari jarak jauh atau lokal. Ini berlaku terutama untuk perangkat lunak yang menghadap ke web, karena lebih terbuka, dan permukaan serangannya jauh lebih besar [3]. Website adalah kumpulan halaman web yang saling terhubung dan seluruh file saling terkait [4]. Dibalik kemudahan layanan yang disediakan oleh setiap website, ternyata terdapat beberapa masalah pada celah keamanan di antaranya: *Cross-Site Scripting*, *information leakage*, *Authentication and Authorization*, *Session management*, *SQL injection, CSRF* dan lain–lain. Serangan *SQL Injection* (SQLi) dilakukan dengan menyisipkan kode berbahaya ke dalam *query database*, sehingga penyerang dapat mengakses, mencuri, atau memodifikasi data tanpa izin [5]. Dengan memanfaatkan celah kemanan ini seseorang dapat melakukan hacking pada website tersebut [6]. Salah satu pendekatan yang digunakan adalah pemanfaatan *web application vulnerability scanner*, yaitu perangkat lunak yang mampu melakukan pengujian penetrasi secara otomatis. Penggunaan alat ini dapat mengurangi waktu, biaya, serta ketergantungan pada keahlian manusia dalam proses pengujian keamanan [7].

Pada penelitian ini dianalisis beberapa celah keamanan pada beberapa website sekolah di kota Palopo dan sekitarnya. Website sekolah sering terhubung dengan berbagai sistem informasi seperti data siswa, guru, nilai akademik, dan informasi administrasi. Jika website memiliki celah keamanan, data tersebut berpotensi dicuri atau dimanipulasi oleh pihak yang tidak bertanggung jawab. Oleh karena itu, analisis kerentanan diperlukan untuk memastikan keamanan data pada sistem website sekolah. Beberapa celah keamanan yang diuji diambil dari hasil pemindaian kerentanan menggunakan aplikasi OWASP ZAP (*Zed Attack Proxy*) yang merupakan sebuah alat *open-socure* yang digunakan untuk melakukan pengujian keamanan apliaksi web. Pemilihan beberapa website sekolah juga dilakukan untuk mengetahui sejauh mana tingkat keamanan aplikasi web yang digunakan oleh institusi pendidikan dalam menyediakan layanan informasi secara daring.

Vulnerability Assessment merupakan kegiatan uji yang memiliki karakteristik yang berkaitan erat dengan penggunaan suatu automation *Vulnerability Scanner*. Di dalamnya, para penguji akan berusaha melakukan validasi terhadap setiap result yang disampaikan dan memberikan tindak lanjut rekomendasi terhadap *issue* yang valid [8]. Melalui pendekatan ini, penelitian dapat memberikan gambaran mengenai kualitas keamanan website sekolah yang ada di Kota Palopo dan sekitarnya. Selain itu, hasil penelitian juga diharapkan dapat menjadi dasar dalam memberikan rekomendasi perbaikan dan peningkatan keamanan sistem, sehingga pengelola website sekolah dapat menerapkan langkah-langkah yang tepat untuk meminimalkan risiko serangan siber serta melindungi data dan informasi yang terdapat dalam sistem tersebut. Melalui penelitian ini, diharapkan dapat diidentifikasi potensi kerentanan keamanan yang ada dalam infrastruktur web dan merekomendasikan langkah-langkah perbaikan yang sesuai untuk meningkatkan keamanan situs web tersebut.

2. Metode Penilitian

Penelitian ini menggunakan metode *Vulnerability Assessment* dengan pendekatan standar keamanan web OWASP Top 10 versi 2025 yang dikembangkan oleh OWASP Foundation. *Open Web Application Security Project* (OWASP) merupakan organisasi non profit berfokus pada peningkatan keamanan perangkat lunak [9]. Daftar ini terus berkembang dan berubah-ubah mengikuti perkembangan teknologi website/aplikasi web yang terus berkembang [10] seperti yang terlihat pada gambar 1 di bawah ini.



Gambar 1. OWASP Top 10 versi 2025

Proses identifikasi kerentanan pada penelitian ini dilakukan menggunakan alat pemindai keamanan web yaitu OWASP ZAP yang digunakan untuk melakukan pemindaian terhadap website target sehingga menemukan potensi celah keamanan seperti kesalahan konfigurasi, kerentanan injeksi, serta kelemahan pada mekanisme autentikasi dan lainnya. Pengujian dalam penelitian ini dilakukan menggunakan perangkat keras dan perangkat lunak, yaitu: 1) Perangkat: Laptop Lenovo LOQ 15IRX9, 2) Sistem Operasi: Windows 11 Home Single Language, 3) RAM: 12 GB, 4) *Tools* Pengujian: OWASP ZAP.

Perangkat tersebut digunakan untuk menjalankan proses *vulnerability assessment* menggunakan OWASP ZAP, termasuk tahapan *spidering* dan *active scanning* terhadap website target. Lingkungan pengujian ini dinilai telah memadai untuk mendukung proses analisis keamanan website secara optimal, serta mampu menangani pemindaian terhadap beberapa objek penelitian secara efisien.

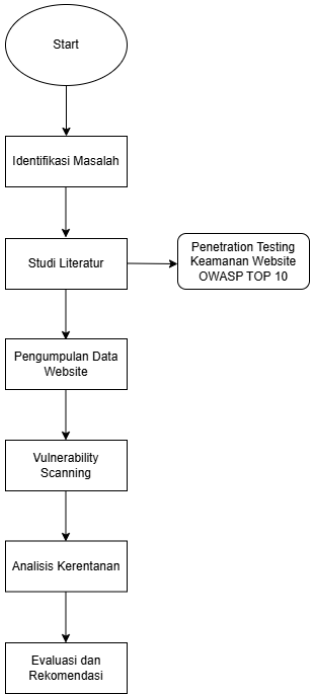
2.1. Tahapan Penelitian

Tahapan penelitian dilakukan melalui beberapa langkah sebagai berikut: 1) Menentukan Objek Penelitian. Penelitian dimulai dengan menentukan objek penelitian berupa beberapa website sekolah yang akan dianalisis tingkat keamanannya. 2) Pengumpulan Data Website. Pada tahap ini dilakukan pengumpulan alamat website yang akan dijadikan sampel penelitian serta pengamatan awal terhadap struktur halaman dan layanan yang tersedia pada website tersebut. 3) Proses Pemindaian Kerentanan (*Vulnerability Scanning*). Website yang menjadi objek penelitian kemudian dipindai menggunakan OWASP ZAP untuk mengidentifikasi potensi kerentanan keamanan yang terdapat pada sistem. 4) Analisis Kerentanan Berdasarkan OWASP Top 10. Hasil pemindaian yang diperoleh selanjutnya dianalisis dan diklasifikasikan berdasarkan kategori kerentanan yang terdapat dalam standar OWASP Top 10, seperti *Injection, Security Misconfiguration, Broken Access Control*, dan kerentanan lainnya. 5) Evaluasi dan Rekomendasi Keamanan. Pada tahap akhir dilakukan evaluasi terhadap tingkat kerentanan yang ditemukan pada setiap website serta memberikan rekomendasi perbaikan keamanan untuk meningkatkan perlindungan sistem terhadap potensi serangan siber.

2.2. Metode Perancangan Sistem

Perancangan sistem dilakukan untuk memberikan gambaran mengenai bagaimana proses analisis keamanan website dilakukan menggunakan standar OWASP Top 10. Perancangan metode penelitian ini bertujuan untuk menggambarkan alur proses analisis kerentanan keamanan pada website yang menjadi objek penelitian. Dalam penelitian ini digunakan tools keamanan web yaitu

OWASP ZAP untuk membantu proses pemindaian kerentanan secara otomatis. *Tools* tersebut digunakan untuk mendeteksi berbagai potensi celah keamanan seperti kesalahan konfigurasi sistem, kerentanan injeksi, serta kelemahan pada mekanisme autentikasi.



Gambar 2. *Use Case Diagram*

Alur tahapan penelitian digambarkan menggunakan *flowchart* agar proses penelitian lebih terstruktur dan mudah dipahami sebelum proses analisis dilakukan. Dengan menggunakan pendekatan standar keamanan OWASP Top 10, penelitian ini diharapkan dapat memberikan gambaran mengenai tingkat keamanan website yang dianalisis serta menghasilkan rekomendasi perbaikan yang dapat meningkatkan keamanan sistem dari potensi serangan siber.

3. Hasil dan Pembahasan

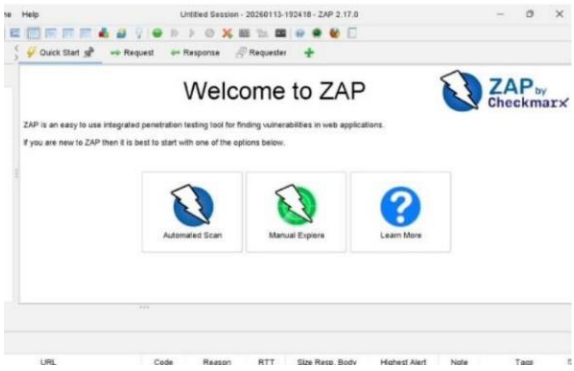
Penelitian ini menggunakan tiga objek penelitian yang merupakan website sekolah yang ada di Kota Palopo dan kabupaten Luwu Timur yaitu SMA Negeri 1 Palopo (https://www.sman1palopo.sch.id/), SMK Negeri 1 Palopo (https://smknegeri1palopo.my.id/) dan SMP Negeri 1 Tomoni (https://smpn1tomoni.sch.id/). Proses pengujian dilakukan menggunakan metode *vulnerability assessment* untuk mengidentifikasi potensi kerentanan keamanan pada website yang menjadi objek penelitian dengan pendekatan standar keamanan web OWASP Top 10 versi 2025.

Gambar 3. Tampilan web SMA Negeri 1 Palopo, SMK Negeri 1 Palopo, dan SMP Negeri 1 Tomoni

3.1. Proses *Scanning*

Proses *scanning* menggunakan OWASP ZAP dilakukan melalui beberapa tahapan untuk mengidentifikasi kerentanan keamanan pada website yang diuji. Tahapan tersebut meliputi proses pengumpulan informasi, pemindaian struktur website, pengujian kerentanan, serta analisis hasil pemindaian. Tahap awal melakukan instalasi dan konfigurasi aplikasi OWASP ZAP pada komputer yang digunakan untuk pengujian. Setelah aplikasi dijalankan, pengguna dapat memilih *mode*

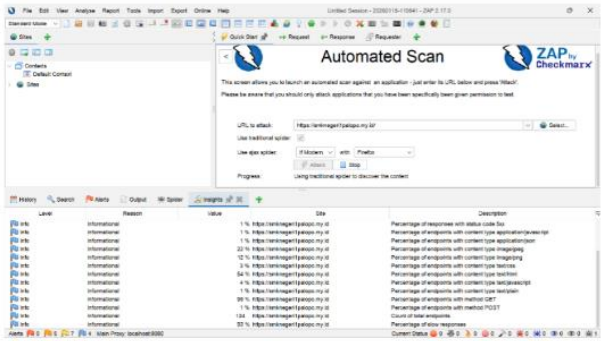
Automated Scan atau *Manual Explore* untuk melakukan proses pemindaian pada website target. Pada tahap ini juga dilakukan pengaturan proxy agar OWASP ZAP dapat memonitor seluruh aktivitas komunikasi antara browser dan website yang diuji.



Gambar 4. Tampilan OWASP ZAP

Setelah konfigurasi selesai, langkah berikutnya adalah memasukkan URL website target yang akan diuji ke dalam fitur *scanning* yang tersedia pada OWASP ZAP. Proses ini bertujuan untuk mengetahui struktur dasar website yang akan dianalisis. OWASP ZAP kemudian akan mulai mengakses halaman website untuk mengidentifikasi berbagai komponen yang terdapat pada website seperti halaman web, *form input*, parameter URL dan file dan direktori.

Tahap selanjutnya menjalankan fitur Spider untuk melakukan crawling terhadap seluruh halaman website yang dapat diakses. Hasil dari proses *spidering* akan menghasilkan daftar seluruh halaman dan endpoint yang terdapat pada website. Spider akan menelusuri setiap *link* yang terdapat pada website untuk menemukan halaman tersembunyi, parameter input dan struktur navigasi website. Setelah proses *spidering* selesai, langkah berikutnya melakukan *Active Scan*. Pada tahap ini OWASP ZAP akan menguji berbagai parameter pada website dengan cara mengirimkan berbagai *payload* untuk mendeteksi potensi kerentanan keamanan. Beberapa jenis kerentanan yang dapat dideteksi yaitu *SQL Injection*, *Cross Site Scripting* (XSS), *Broken Authentication*, *Security Misconfiguration* dan *Information Disclosure*. Salah satu ancaman utama pada layanan berbasis web adalah serangan XSS, yang dapat mengeksploitasi kelemahan aplikasi dengan menyisipkan skrip berbahaya ke dalam halaman HTML yang dihasilkan [11]. Web modern seperti Web Services, API, AJAX, HTML5, dan CSS3 turut memperumit permasalahan XSS. Teknologi tersebut memungkinkan interaksi yang lebih kompleks antara klien dan server, namun di sisi lain juga membuka peluang munculnya jenis kerentanan baru yang lebih sulit dideteksi dan ditangani [12].



Gambar 5. Proses *Scanning* OWASP ZAP

Pada proses *scanning*, OWASP ZAP akan menampilkan hasil analisis berupa daftar kerentanan yang ditemukan pada website target. Setiap kerentanan akan diklasifikasikan berdasarkan tingkat risiko. Selain itu, OWASP ZAP juga menyediakan penjelasan mengenai jenis kerentanan, lokasi kerentanan, serta rekomendasi perbaikan yang dapat dilakukan oleh pengembang website.

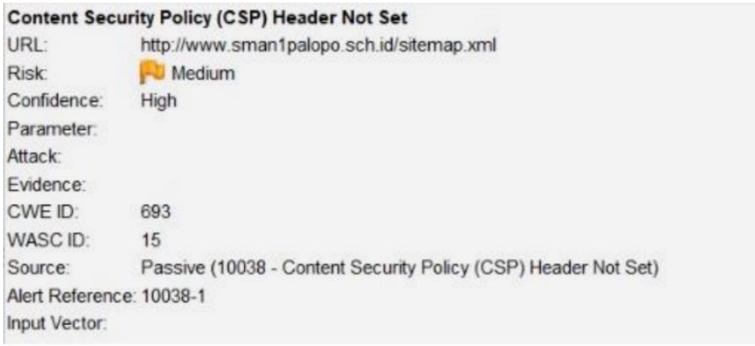
Tabel 1. Tabel Peringatan

Tingkatan Risiko	Keterangan
High	Kerentanan dengan potensi dampak serius terhadap sistem
Medium	Kerentanan dengan tingkat risiko sedang
Low	Kerentanan dengan dampak relatif kecil
Informational	Informasi tambahan terkait konfigurasi sistem

3.2. Hasil *Scanning*

Vulnerability dalam aplikasi web merupakan kelemahan yang dapat dimanfaatkan oleh penyerang untuk mendapatkan akses tidak sah, mencuri data sensitif, atau merusak sistem. Kerentanan ini dapat disebabkan oleh kesalahan desain, kesalahan pemrograman, maupun konfigurasi sistem yang tidak tepat [13]. Hasil *scanning vulnerability* menunjukkan bahwa ketiga website yang diuji masih memiliki beberapa potensi kerentanan keamanan dengan tingkat risiko yang berbeda, yaitu *High Risk, Medium Risk, Low Risk*, dan *Informational*. Kerentanan yang ditemukan sebagian besar berkaitan dengan konfigurasi keamanan server, kurangnya penerapan *security header*, serta potensi kelemahan pada mekanisme pengolahan input.

Pada website SMA Negeri 1 Palopo ditemukan beberapa kerentanan dengan kategori medium dan low risk, seperti tidak diterapkannya *Content Security Policy* (CSP) atau belum dikonfigurasi dengan baik sehingga menunjukkan bahwa website belum memiliki lapisan keamanan tambahan untuk membatasi sumber konten yang boleh dimuat oleh browser. Tanpa CSP, browser akan menerima dan mengeksekusi seluruh konten yang dimuat oleh halaman web, termasuk skrip atau data berbahaya yang disisipkan oleh penyerang. Kondisi ini meningkatkan risiko terjadinya serangan seperti *Cross-Site Scripting* (XSS) dan *data injection*, di mana penyerang dapat menyisipkan kode berbahaya untuk mencuri data pengguna, merusak tampilan website, atau menyebarkan *malware*. Dampak dari serangan *Cross-Site Scripting* (XSS) sangat serius, seperti pencurian data pribadi, pengambilalihan sesi (*session hijacking*), hingga manipulasi konten situs web [14].

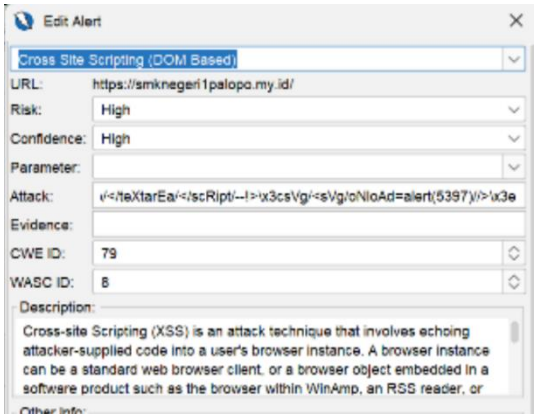


Gambar 6. *Vulnerability Content Security Policy* (CSP)

Selain *Content Security Policy* (CSP). Ditemukan juga missing *integrity* attribute menunjukkan bahwa website memuat file eksternal, seperti JavaScript atau CSS, tanpa menggunakan atribut *Subresource Integrity* (SRI). Kondisi ini menyebabkan browser tidak melakukan verifikasi keaslian file yang diambil dari server pihak ketiga.

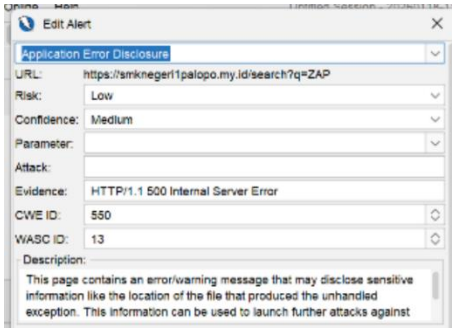
Gambar 7. *Vulnerability Content Security Policy* (CSP)

Sementara itu, hasil *scanning* pada website SMK Negeri 1 Palopo menunjukkan adanya kerentanan yang berkaitan dengan konfigurasi keamanan website, seperti *Cross Site Scripting* (XSS) yang merupakan teknik serangan yang melibatkan penyisipan kode berbahaya yang disediakan oleh penyerang ke dalam instans browser pengguna. Serangan *Cross-site Scripting* pada dasarnya merusak hubungan kepercayaan antara pengguna dan situs web. Aplikasi yang menggunakan instans objek browser yang memuat konten dari sistem file dapat mengeksekusi kode di bawah zona mesin lokal yang memungkinkan kompromi sistem. Ada tiga jenis serangan *Cross-site Scripting*: non-persisten, persisten, dan berbasis DOM.



Gambar 8. *Vulnerability Cross Site Scripting* (XSS)

Terdapat juga *Application Error Disclosure* berisi pesan kesalahan/peringatan yang dapat mengungkapkan informasi sensitif seperti lokasi file yang menghasilkan pengecualian yang tidak tertangani. Informasi ini dapat digunakan untuk meluncurkan serangan lebih lanjut terhadap aplikasi web. Peringatan bisa menjadi positif palsu jika pesan kesalahan ditemukan di dalam halaman dokumentasi.



Gambar 8. *Vulnerability Application Error Disclosure*

Pada website SMP Negeri 1 Tomoni ditemukan *Absence of Anti-CSRF Tokens* terdeteksi melalui aplikasi OWASP-ZAP. Serangan cross site *request* forgery (CSRF) dapat terjadi disebabkan karena tidak adanya mekanisme perlindungan terhadap token keamanan (*request token*) pada sebuah website, sehingga penyerang dapat mengirim suatu *request* (*sumbit* suatu *form*) secara illegal. Hasil pengujian menunjukkan bahwa pada formulir HTML yang dianalisis tidak ditemukan mekanisme Anti-CSRF Token yang merupakan salah satu metode pengamanan penting yang berfungsi untuk memastikan bahwa setiap permintaan (*request*) yang dikirim ke server benar-benar berasal dari pengguna yang sah. Ketiadaan token ini menyebabkan aplikasi web rentan terhadap serangan *Cross Site Request Forgery* (CSRF). Dari sisi dampak, serangan CSRF dapat bervariasi mulai dari tingkat rendah hingga tinggi, tergantung pada fungsi aplikasi yang diserang dan hak akses korban. Dampak ringan dapat berupa pemaksaan logout pengguna, sedangkan dampak yang lebih serius meliputi perubahan data profil, pengambilalihan akun, hingga manipulasi data sensitif [15].

3.3. Rekapitulasi Tingkat Kerentanan

Berikut rekapitulasi jumlah temuan kerentanan pada masing-masing website:

Tabel 2. Perbandingan Tingkat Kerentanan Website

Website	High	Medium	Low	Informational
SMA Negeri 1 Palopo	0	4	8	6
SMK Negeri 1 Palopo	0	5	10	7
SMP Negeri 1 Tomoni	0	5	10	8

Secara keseluruhan, hasil *Vulnerability Assessment* menunjukkan bahwa sebagian besar kerentanan yang ditemukan berkaitan dengan konfigurasi keamanan server dan kurangnya penerapan mekanisme keamanan pada aplikasi web. Hasil pemindaian menunjukkan bahwa ketiga website target rentan terhadap berbagai jenis serangan, dengan kerentanan *Cross-Site Scripting* (XSS) *DOM-Based* sebagai temuan paling kritis yang tergolong dalam kategori risiko tinggi. Selain itu, terdeteksi pula sejumlah kerentanan tingkat sedang seperti absennya token anti-CSRT, tidak diterapkannya *Content Security Policy* (CSP), serta konfigurasi keamanan lintas domain yang kurang ketat, yang secara bersama-sama dapat dimanfaatkan oleh penyerang untuk melancarkan eksploitasi lebih lanjut. Berdasarkan analisis yang dilakukan, rekomendasi perbaikan difokuskan pada penerapan sanitasi input yang ketat, implementasi mekanisme anti-CSRT, penguatan konfigurasi header keamanan, serta pemutakhiran komponen perangkat lunak yang rentan. Diperlukan pula pendekatan keamanan berkelanjutan melalui pemindaian rutin, integrasi pengujian keamanan dalam siklus pengembangan, dan peningkatan kesadaran keamanan bagi pengelola sistem.

Disarankan juga untuk melakukan peningkatan konfigurasi keamanan secara menyeluruh, dimulai dari pembaruan seluruh library, framework, dan komponen pihak ketiga ke versi terbaru guna menghindari eksploitasi celah keamanan yang sudah diketahui. Selain itu, pengelolaan dependensi perlu dilakukan secara berkala agar tidak menggunakan komponen yang sudah usang atau rentan. Selanjutnya, pihak pengelola website perlu menerapkan konfigurasi header keamanan HTTP secara lengkap, seperti mengaktifkan *Content Security Policy* (CSP).

4. Kesimpulan

Berdasarkan hasil analisis keamanan yang telah dilakukan terhadap 3 (tiga) website sekolah, dapat disimpulkan bahwa sistem website masih memiliki berbagai kelemahan pada aspek konfigurasi dan pengelolaan keamanan yang berpotensi meningkatkan risiko serangan siber. Hasil *vulnerability assessment* menggunakan OWASP ZAP menunjukkan bahwa ketiga website target rentan terhadap berbagai jenis serangan, dengan kerentanan *Cross-Site Scripting* (XSS) sebagai temuan paling kritis yang tergolong dalam kategori risiko tinggi. Selain itu, ditemukan pula beberapa kerentanan tingkat sedang seperti absennya mekanisme Anti-CSRF Token, tidak diterapkannya *Content Security Policy* (CSP), serta konfigurasi keamanan lintas domain yang kurang optimal. Kerentanan-kerentanan tersebut saling berkaitan dan berpotensi dimanfaatkan secara bersamaan oleh penyerang untuk melakukan eksploitasi lanjutan. Meskipun sebagian temuan berada pada kategori Low dan Informational, kondisi ini tetap dapat dimanfaatkan sebagai tahap awal dalam proses information gathering oleh penyerang. Secara keseluruhan, kelemahan yang ditemukan berdampak langsung terhadap aspek kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*availability*) data, khususnya data siswa dan informasi akademik yang dikelola oleh sistem website sekolah. Oleh karena itu, diperlukan upaya peningkatan keamanan secara menyeluruh melalui perbaikan konfigurasi sistem, penerapan validasi input yang ketat, serta pembaruan komponen perangkat lunak secara berkala. Secara lebih teknis dan aplikatif, pengelola IT sekolah disarankan untuk mengimplementasikan mekanisme keamanan tambahan seperti penggunaan *Web Application Firewall* (WAF) untuk memfilter dan memitigasi serangan berbasis web, misalnya menggunakan layanan seperti Cloudflare atau solusi *open-source* seperti ModSecurity. Selain itu, penting untuk memastikan CMS, plugin, dan seluruh komponen sistem selalu diperbarui guna menghindari eksploitasi terhadap

celah keamanan yang telah diketahui (*known vulnerabilities*), serta menerapkan *security header* seperti CSP dan mekanisme Anti-CSRF Token pada setiap form penting. Dengan penerapan langkah-langkah tersebut, diharapkan tingkat keamanan website sekolah dapat meningkat secara signifikan serta mampu melindungi data siswa dari potensi kebocoran maupun manipulasi oleh pihak yang tidak bertanggung jawab.

Daftar Pustaka

[1] E. Nurelasari, D. Gumilang, and A. Farabi, "*Analisis Keamanan Sistem Website Menggunakan Metode Open Web Application Security Project (Owasp) Pada Simantep.Id*," 2024.

[2] A. H. Harahap, C. Difa Andani, A. Christie, D. Nurhaliza, and A. Fauzi, "Pentingnya Peranan CIA Triad Dalam Keamanan Informasi dan Data Untuk Pemangku Kepentingan atau Stakholder," *Jurnal Manajemen dan Pemasaran Digital (JMPD)*, vol. 1, pp. 73–83, Apr. 2023.

[3] Hanafi, "*Dasar Cyber Security dan Forensic*," Nov. 2022.

[4] H. Herman, I. Riadi, Y. Kurniawan, and I. A. Rafiq, "Analisis Keamanan Website Menggunakan Information System Security Asessment Framework (ISSAF)," *Jurnal Teknologi Informatika dan Komputer*, vol. 9, no. 1, pp. 126–136, Mar. 2023, doi: 10.37012/jtik.v9i1.1439.

[5] J. R. Tadhani, V. Vekariya, V. Sorathiya, S. Alshathri, and W. El-Shafai, "Securing web applications against XSS and SQLi attacks using a novel deep learning approach," *Sci. Rep.*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-023-48845-4.

[6] L. M. Gultom and M. Harahap, "*Analisis Celah Keamanan Website Instansi Pemerintahan di Sumatera Utara*," 2015. [Online]. Available: www.binjaikota.go.id

[7] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, and N. Crespi, "A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions," *Applied Sciences (Switzerland)*, vol. 12, no. 8, Apr. 2022, doi: 10.3390/app12084077.

[8] H. Y. F. Kho Yoko, "*Bug Hunting 101-Web Application Security Testing*," 2021.

[9] A. W. Kuncoro, F. Rahma, and M. E. Jurusan Informatika, "Analisis Metode *Open Web Application Security Project* (OWASP) pada Pengujian Keamanan Website: Literature Review," *Automata*, vol.03, no. 01, Jan. 2022.

[10] Y. Thurfah Afifa Rosaliah and B. Hananto, Pengujian Celah Keamanan Website Menggunakan Teknik Penetration Testing dan Metode OWASP TOP 10 pada Website SIM xxx. *Jakarta: Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, 2021.

[11] J. Kim and J. Park, "Enhancing Security of Web-Based IoT Services via XSS *Vulnerability* Detection †," *Sensors*, vol. 23, no. 23, Dec. 2023, doi: 10.3390/s23239407.

[12] A. Hannousse, S. Yahiouche, and M. C. Nait-Hamoud, "*Twenty-two years since revealing cross-site scripting attacks: a systematic mapping and a comprehensive survey*," May 2022, doi: 10.1016/j.cosrev.2024.100634.

[13] A. E. Hafez and M. M. Almustafa, "Detecting Security Vulnerabilities in Web Applications: A Proposed System," *International Journal of Safety and Security Engineering*, vol. 14, no. 6, pp. 1933–1940, Dec. 2024, doi: 10.18280/ijsse.140627.

[14] Tenzin Yarphel and Diksha Rani, "*Cross-Site Scripting* (XSS) in Web Applications: A systematic literature review," *International Journal of Science and Research Archive*, vol. 15, no. 2, pp. 1658–1667, May 2025, doi: 10.30574/ijsra.2025.15.2.1521.

[15] C. N. Siahaan, M. Rufisanto, R. Nolasco, S. Achmad, and C. R. P. Siahaan, "Study of Cross-Site Request Forgery on Web-Based Application: Exploitations and Preventions," *in Procedia Computer Science, Elsevier B.V.*, 2023, pp. 92–100. doi: 10.1016/j.procs.2023.10.506.