Department of Digital Business
Journal of Artificial Intelligence and Digital Business (RIGGS)
Homepage: https://journal.ilmudata.co.id/index.php/RIGGS
Vol. 4 No. 4 (2025) pp: 2670-2678
P-ISSN: 2963-9298, e-ISSN: 2963-914X

Pengujian Kerentanan dan Mitigasi Website SIAKAD Fakultas Kedokteran UNESA dengan OWASP ZAP

[1]Mahbubi, [2]Dewangga Alun Kalokajaya, [3]Andika Faliyastats Yunus, [4]Fauzan Ainur Rofiq, [5]Riski Surya Permana, [6]Ahmad Marwan Taufiq, [7]I Gusti Lanang Putra Eka Prismana, [8]Mohammad Wildan Habibi
[1,2,3,4,5,6,7,8]Pendidikan Teknologi Informasi, Fakultas Teknik, Universitas Negeri Surabaya
[1]mahbubi.22098@mhs.unesa.ac.id, [2]dewangga.22099@mhs.unesa.ac.id, [3]andika.22105@mhs.unesa.ac.id, [4]fauzan.22117@mhs.unesa.ac.id, [5]riski.22118@mhs.unesa.ac.id, [6]ahmad.22120@mhs.unesa.ac.id, [7]lanangprismana@unesa.ac.id, [8]mohammadhabibi@unesa.ac.id

Abstrak

Keamanan siber merupakan aspek kritis dalam pengelolaan sistem informasi, terutama untuk platform yang menangani data sensitif seperti Sistem Informasi Akademik (SIAKAD). Penelitian ini bertujuan untuk melakukan evaluasi keamanan secara mendalam pada website SIAKAD Fakultas Kedokteran Universitas Negeri Surabaya dengan memanfaatkan alat OWASP Zed Attack Proxy (ZAP). Metode pengujian diterapkan melalui beberapa tahapan sistematis, dimulai dari pengumpulan informasi awal, dilanjutkan dengan pemindaian kerentanan otomatis, analisis eksploitasi, evaluasi potensi akses ilegal, hingga penyusunan rekomendasi laporan keamanan yang komprehensif. Hasil investigasi mengungkapkan bahwa sistem tersebut mengandung sejumlah kerentanan dengan tingkat risiko yang bervariasi. Secara total, teridentifikasi 21 jenis peringatan, yang diklasifikasikan mulai dari risiko tinggi, sedang, rendah, hingga informasional. Temuan paling kritis bersumber dari komponen JavaScript yang rentan, sementara insiden terbanyak ditemukan pada kerentanan pemanggilan berkas JavaScript lintas domain. Berbagai kelemahan lain yang termasuk dalam kategori OWASP Top 10 juga terdeteksi, seperti kesalahan konfigurasi keamanan, pengungkapan informasi sensitif, mekanisme autentikasi yang lemah, penggunaan komponen usang, serta kurangnya pengelolaan pencatatan dan pemantauan keamanan yang memadai. Sebagai solusi, penelitian ini merekomendasikan serangkaian langkah mitigasi strategis, termasuk perbaikan konfigurasi kebijakan keamanan konten, penguatan pengaturan cookie, pembaruan komponen secara berkala, dan penerapan header keamanan yang wajib. Simpulan dari studi ini membuktikan bahwa OWASP ZAP merupakan instrumen yang efektif untuk mengaudit keamanan website sekaligus memberikan peta jalan yang jelas bagi perbaikan dan peningkatan postur keamanan digital suatu institusi.

***Kata kunci**: Keamanan Web, Kerentanan Aplikasi, OWASP Zed Attack Proxy, Sistem Informasi Akademik, Uji Penetrasi*

1. Latar Belakang

Perkembangan teknologi web dan aplikasi berbasis internet telah menimbulkan tantangan keamanan yang semakin kompleks di level global. Studi terbaru menunjukkan bahwa pelanggaran keamanan yang memanfaatkan aplikasi web sebagai vektor serangan tumbuh pesat, sehingga organisasi harus mengadopsi strategi pengujian kerentanan yang sistematis. Sebagai contoh, laporan (Verizon, 2024) menyebutkan bahwa eksploitasi kerentanan sebagai pintu masuk utama insiden meningkat hingga sekitar 180% dibandingkan tahun sebelumnya, dan vektor yang paling banyak digunakan adalah aplikasi web. Hal serupa juga diungkapkan dalam IBM X-Force Threat Intelligence Index (2024) yang menyoroti meningkatnya serangan berbasis aplikasi web akibat lemahnya kontrol validasi dan autentikasi input pengguna (IBM, 2024).

Sementara itu, survei oleh (Cycognito, 2024) mengemukakan bahwa lebih dari separuh responden (53%) mengalami kesulitan dalam menanggapi hasil pengujian keamanan aplikasi web, dan sekitar 75% hanya melakukan pengujian sebulan sekali atau kurang artinya masih banyak bagian dari permukaan serangan (attack surface) yang luput dari pengujian. Temuan ini sejalan dengan penelitian (Alenezi dan Almustafa, 2021) yang menegaskan bahwa celah antara kemampuan deteksi alat otomatis dan kompleksitas kerentanan nyata di lapangan masih menjadi tantangan utama dalam audit keamanan. Kondisi ini memperlihatkan bahwa meskipun teknik dan alat keamanan makin berkembang, penerapan secara konsisten masih menghadapi hambatan signifikan (Ventura, Franco and Akram, 2023). Dalam sektor pendidikan tinggi, termasuk sistem informasi akademik, risiko tersebut menjadi kian kritikal. Lembaga pendidikan menyimpan data sensitif mahasiswa, dosen, keuangan, dan administrasi yang jika terekspos dapat berdampak besar secara reputasi dan operasional (Lallie and Titis, 2023); (Armis, 2024).

Di Indonesia sendiri, penelitian tentang sistem informasi akademik (SIAKAD) menegaskan bahwa kerentanan pada website akademik dapat menghasilkan sejumlah temuan, termasuk beberapa kerentanan tingkat sedang dan rendah dalam satu pengujian menggunakan OWASP ZAP (Awlarijal, Almaarif and Budiono, 2020). Penelitian oleh (Muzaki *et al.*, 2025) menunjukkan bahwa pengujian keamanan web berbasis OWASP ZAP dapat mendeteksi kerentanan signifikan pada sistem pembelajaran daring universitas, terutama pada komponen login dan form input mahasiswa. Temuan ini memperkuat pentingnya pendekatan automasi pengujian pada sistem akademik. (Akbar *et al.*, 2025) menyoroti bahwa penerapan teknik black-box testing pada sistem informasi akademik (SIAKAD) kampus mampu mengidentifikasi kelemahan logika bisnis dan validasi input yang berpotensi dieksploitasi. Studi ini menunjukkan efektivitas pendekatan pengujian dinamis dalam konteks perguruan tinggi. Dengan demikian, urgensi pengujian kerentanan pada domain akademik tidak hanya berupa isu teknologi, melainkan juga terkait proteksi data, kontinuitas layanan, dan kepercayaan pengguna (Utama, Muhamad and Nurhadi, 2024). Tinjauan teoritis menunjukkan bahwa kerangka acuan utama dalam keamanan aplikasi web sering merujuk pada OWASP Top 10 sebuah standar kesadaran risiko yang diakui secara internasional sebagai titik awal praktik pengamanan aplikasi web (Meucci and Muller, 2014).

Secara historis, teori keamanan aplikasi web berevolusi dari pendekatan mitigasi reaktif menjadi pendekatan shift-left yang mengintegrasikan pengujian kerentanan sejak tahap pengembangan aplikasi. Misalnya, penelitian oleh (Jakobsson and Häggström, 2022) mengkaji teknik yang digunakan ZAP dan alat scanner lainnya dalam mendeteksi dua jenis kerentanan umum dalam aplikasi web. Hal ini memperlihatkan bahwa pendekatan teoretis dalam pengujian keamanan kini semakin berfokus pada automasi, integrasi dengan siklus hidup pengembangan perangkat lunak (SDLC), dan pemilihan alat yang tepat (Maniraj, Ranganathan and Sekar, 2024).

Pada tingkat praktik, kerangka pengujian kerentanan (vulnerability assessment) dan penetration testing telah banyak digunakan dalam literatur. Misalnya, studi oleh (Maniraj, Ranganathan and Sekar, 2024)menguraikan penerapan ZAP sebagai alat utama untuk scanning aplikasi web menggunakan simulasi serangan seperti injeksi SQL dan XSS untuk mengekspos kelemahan aplikasi. Penelitian lainnya menunjukkan bahwa alat OWASP ZAP terbukti efektif untuk pengujian keamanan di berbagai konteks aplikasi, termasuk pendidikan dan pemerintahan (Ventura, Franco and Akram, 2023). Dengan demikian, dalam lingkungan akademik yang menggunakan sistem informasi berbasis web, metodologi seperti black-box scanning, active/passive crawling, dan analisis kerentanan otomatis menjadi bagian penting dari strategi keamanan aplikasi web.

Meskipun demikian, terdapat permasalahan spesifik yang belum banyak ditangani dalam literatur. Khususnya untuk website sistem informasi akademik fakultas kedokteran di perguruan tinggi Indonesia, belum banyak ditemukan studi empiris yang secara sistematis melakukan pengujian kerentanan dan merekomendasikan mitigasi berbasis alat ZAP. Padahal, sistem informasi akademik memiliki karakteristik unik: pengolahan data sensitif (mahasiswa, dosen, keuangan, dan potensi medis) serta beroperasi di lingkungan yang melibatkan mobile computing (akses melalui perangkat seluler). Kesenjangan ini membuka peluang penelitian yang lebih terfokus pada keamanan aplikasi akademik berbasis web dan akses mobile (Putra et al., 2024). (Kadir, Irsan and Putrada, 2025) mengkaji keamanan aplikasi mobile universitas di Indonesia dengan kerangka OWASP Mobile Top 10 dan menemukan bahwa sebagian besar institusi belum menerapkan pengujian keamanan terintegrasi antara aplikasi web dan mobile. Hal ini menandakan adanya kesenjangan dalam perlindungan data akademik lintas platform.

(Sembiring, 2025) menjelaskan bahwa sistem informasi kampus, termasuk pengelolaan surat masuk dan keluar, merupakan bagian penting dari infrastruktur digital universitas yang perlu dijaga keamanannya, karena menjadi pintu akses terhadap data sensitif institusi. Urgensi penelitian ini sangat tinggi: jika kerentanan situs akademik tidak diidentifikasi dan ditangani dengan tepat, potensi akses tidak sah, manipulasi data, atau gangguan layanan (downtime) dapat merusak kepercayaan internal dan eksternal, serta memunculkan implikasi terhadap regulasi perlindungan data (Armis, 2024). Data industri juga menunjukkan bahwa sektor pendidikan merupakan target paling rentan, di mana lebih dari 65% universitas kekurangan konfigurasi keamanan dasar, dan 77% serangan melibatkan enkripsi data atau ransomware (Lallie and Titis, 2023). Dengan demikian, penelitian ini penting baik secara teoretis maupun praktis untuk mendukung penguatan keamanan situs akademik di Indonesia.

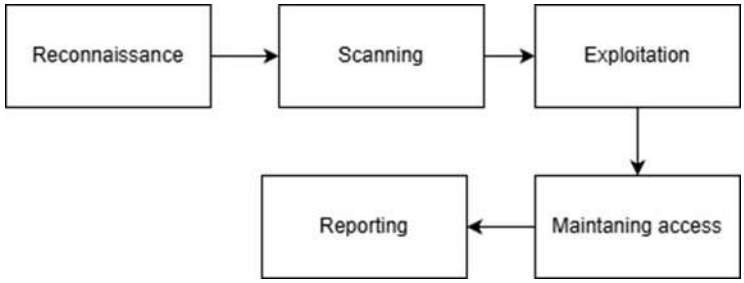
2. Metode Penelitian

2.1. Keranga Kerja

Metode penilaian risiko OWASP merupakan pendekatan yang digunakan untuk mengidentifikasi, menghitung, dan mengevaluasi tingkat kerentanan serta risiko keamanan pada suatu website. Melalui metode ini, peneliti atau pengelola sistem dapat menentukan langkah-langkah yang tepat dalam menangani setiap risiko yang ditemukan. Dengan memahami potensi risiko sejak awal, organisasi dapat menghemat waktu dalam proses mitigasi serta

DOI: https://doi.org/10.31004/riggs.v4i4.3686

mencegah terjadinya ancaman keamanan yang lebih serius di kemudian hari. Berikut ini adalah kerangka kerja OWASP sebagaimana dapat dilihat dari gambar berikut:



Gambar 1. Alur Proses Peneletian menggungakan OWASP ZAP

Setiap tahapan dalam kerangka kerja OWASP terdiri dari lima bagian utama yang saling berhubungan yaitu:

1. Reconnaissance, Merupakan tahap awal yang bertujuan untuk mengumpulkan informasi dan data penting mengenai sistem web target, seperti struktur, teknologi yang digunakan, serta potensi titik lemah yang dapat dimanfaatkan.
3. Scanning, dilakukan untuk melakukan pemindaian lebih mendalam terhadap sistem, guna mengidentifikasi kerentanan dan mengetahui respon website terhadap berbagai bentuk permintaan (request).
4. Exploitation, Tahap ini bertujuan untuk memanfaatkan kerentanan yang telah ditemukan pada tahap sebelumnya, sehingga dapat mengetahui sejauh mana celah keamanan tersebut bisa dieksploitasi.
5. Maintaining Access, Setelah berhasil mengeksploitasi sistem, tahap ini digunakan untuk mempertahankan akses yang telah diperoleh, misalnya dengan menanamkan backdoor atau skrip tertentu melalui celah keamanan yang ada.
6. Reporting, Merupakan tahap akhir yang berfokus pada penyusunan laporan hasil pengujian, yang berisi temuan kerentanan, tingkat risikonya, serta rekomendasi perbaikan dan solusi untuk meningkatkan keamanan sistem.

2.2. Alat dan Bahan

Dalam penelitian ini, digunakan beberapa alat dan bahan yang berfungsi untuk mendukung proses analisis keamanan sistem berbasis web. Pemilihan alat dan bahan disesuaikan dengan kebutuhan pengujian menggunakan kerangka kerja OWASP agar hasil yang diperoleh akurat dan dapat dipertanggungjawabkan. Rincian alat dan bahan yang digunakan ditunjukkan pada Tabel 1.

Tabel 1. Perangkat Lunak dan Perangkat Keras Pendukung

Alat	Peladen
Laptop	Processor: AMD Ryzen 9 5900HX with Radeon Graphics OS: Windows 11 64 bit RAM: 16 GB VGA: AMD Radeon SSD: 500 GB
OWASP ZAP	Versi 2.16.1
Web Browser	Google Chrome

3. Hasil dan Diskusi

Penelitian ini memanfaatkan OWASP ZAP sebagai alat untuk mengoptimalkan keamanan pada website dengan cara mendeteksi serta menanggulangi berbagai celah kerentanan yang berpotensi dimanfaatkan oleh pihak yang tidak berwenang.

3.1. Reconnaissance

Peneliti mengumpulkan data dan informasi terkait sistem web yang akan diuji, dengan melakukan pengujian langsung pada situs https://fk.unesa.ac.id. Pengamatan awal terhadap struktur, komponen, serta potensi titik lemah pada website tersebut. Tahapan ini bertujuan untuk memperoleh gambaran menyeluruh mengenai target pengujian sehingga analisis keamanan dapat dilakukan secara lebih terarah dan efektif.

3.2. Scanning

Pada tahap ini, peneliti menggunakan fitur Automated Scan dari OWASP ZAP untuk melakukan pemindaian otomatis terhadap website target. Proses ini bertujuan untuk mendeteksi berbagai kerentanan keamanan seperti

DOI: https://doi.org/10.31004/riggs.v4i4.3686

SQL Injection, Cross-Site Scripting (XSS), serta konfigurasi server yang tidak aman. Proses ini melibatkan program OWASP ZAP ketika tampilan awal muncul. Setelah itu, pengguna harus mengklik Automatic Scan pada kolom Welcome to OWASP ZAP, sehingga tampilan seperti pada Gambar 2.

Gambar 2. Tampilan Home OWASP ZAP

Tahapan proses scanning dilakukan secara sistematis untuk memastikan setiap komponen pada website dapat teridentifikasi secara menyeluruh. Prosedur ini melibatkan serangkaian langkah yang mendukung proses pendeteksian kerentanan secara otomatis melalui OWASP ZAP. Berikut merupakan langkah-langkah yang dilakukan dalam proses scanning menggunakan metode Automated Scan.

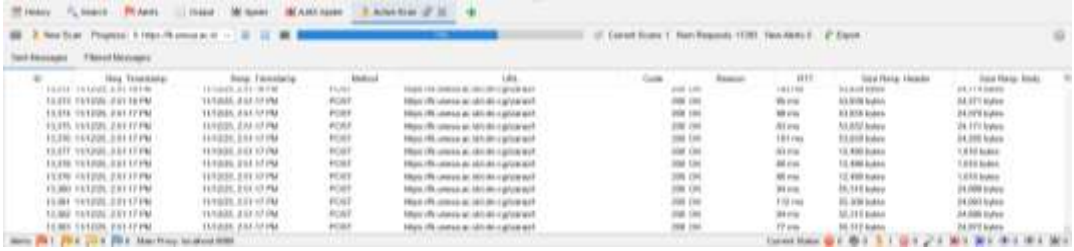
1. Input URL Website

Pada tahap awal, peneliti memasukkan alamat website yang akan diuji ke dalam kolom URL pada OWASP ZAP. Selanjutnya, dilakukan pengaturan dengan memilih opsi Use traditional spider serta Use AJAX spider for scanning assistance untuk membantu proses penelusuran halaman web. Setelah konfigurasi selesai, proses pemindaian otomatis dijalankan dengan menekan tombol Attack. OWASP ZAP kemudian akan memulai analisis terhadap seluruh struktur dan elemen situs berdasarkan URL yang telah dimasukkan. Tampilan proses Automated Scan ditunjukkan pada Gambar 3, yang memperlihatkan antarmuka ZAP saat melakukan pemindaian awal terhadap target website.

Gambar 3. Tampilan Automated Scan pada OWASP ZAP

2. Scanning Website

Setelah proses pemindaian dimulai, OWASP ZAP akan menampilkan progres aktivitas scanning secara real time. Beberapa indikator penting yang diamati meliputi persentase kemajuan pemindaian, daftar URL yang sedang dianalisis, serta metode HTTP yang digunakan pada setiap permintaan. Setiap URL ditandai dengan ikon berwarna hijau menandakan keberhasilan, sedangkan merah menunjukkan adanya kesalahan. Selain itu, URL yang diberi label “Out of Scope” tidak termasuk dalam proses analisis, karena berada di luar batas ruang lingkup pengujian. Bagian message log juga menampilkan notifikasi dan catatan tambahan yang berguna dalam proses debugging serta evaluasi hasil pemindaian. Proses ini dapat dilihat pada Gambar 4, yang memperlihatkan tahapan analisis URL secara detail.



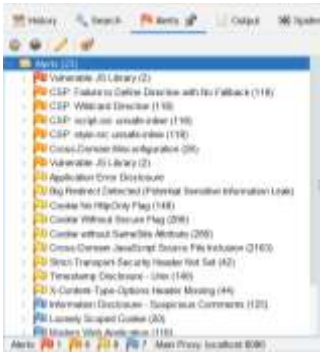
DOI: https://doi.org/10.31004/riggs.v4i4.3686

Gambar 4. Proses Scanning URL Website

Gambar 4 menunjukkan daftar peringatan (alerts) yang terdeteksi setelah proses pemindaian selesai dilakukan. Melalui tampilan ini, pengguna dapat meninjau informasi detail dari setiap peringatan, meliputi jenis kerentanan yang ditemukan, tingkat keparahan, serta rekomendasi tindakan perbaikan yang disarankan oleh sistem. Selain itu, dengan melakukan klik ganda pada salah satu peringatan, pengguna memiliki opsi untuk meninjau lebih lanjut, menambahkan, atau mengedit detail peringatan yang sudah ada guna mendukung analisis keamanan secara lebih mendalam.

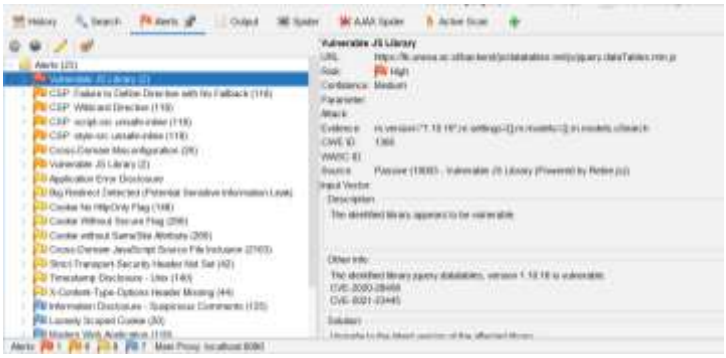
3.3 Exploitation

Tahap ini merupakan proses pengujian atau testing terhadap keamanan situs web untuk mengetahui tingkat kerentanannya. Setelah dilakukan pemindaian, sistem menampilkan beberapa alert yang mengindikasikan adanya potensi ancaman. Gambar 5 menunjukkan hasil ancaman yang ditemukan, di mana terdapat tujuh alert, antara lain Content Security Policy, Missing Anti-Clickjacking Header, Server Leaks, Strict Transport Security, X-Content-Type Options, Modern Web Application, dan Re-examine Cache. Masing-masing alert kemudian dianalisis untuk mendapatkan pemahaman lebih mendalam mengenai jenis kerentanan yang terjadi.



Gambar 5. Hasil Ancaman dari Proses Scanning Website

Setelah menganalisis setiap peringatan, OWASP ZAP menampilkan rincian detail mengenai setiap kerentanan yang ditemukan. Informasi ini meliputi URL yang terdampak, tingkat keyakinan terhadap keberadaan kerentanan, jenis serangan yang mungkin terjadi, serta lokasi input yang rentan terhadap eksploitasi. Rincian ini sangat penting untuk menentukan langkah perbaikan keamanan yang tepat guna mencegah penyalahgunaan lebih lanjut oleh pihak yang tidak bertanggung jawab. Visualisasi hasil detail alert ditunjukkan pada Gambar 6



Gambar 6. Merupakan Rician dari Alert

3.4 Maintaining Access

Tahap Maintaining Access bertujuan untuk mengevaluasi sejauh mana potensi peretas dapat mempertahankan akses terhadap sistem target setelah berhasil melakukan eksploitasi. Biasanya, tahap ini dilakukan dengan memanfaatkan celah keamanan yang belum diperbaiki, seperti Remote Code Execution (RCE) atau SQL Injection, yang memungkinkan penyusupan script berbahaya ke dalam sistem.

Sebagai contoh, serangan dapat diarahkan pada platform CMS yang tidak diperbarui, sehingga membuka peluang bagi penyerang untuk menanamkan malicious file berfungsi sebagai backdoor. File ini memungkinkan penyerang memperoleh akses berulang ke sistem tanpa perlu melakukan autentikasi ulang.

DOI: https://doi.org/10.31004/riggs.v4i4.3686

Selain itu, penggunaan kata sandi bawaan, autentikasi yang lemah, atau konfigurasi yang tidak aman juga dapat dimanfaatkan untuk mempertahankan akses ilegal tersebut. Oleh karena itu, penerapan langkah-langkah keamanan seperti patch management, pemantauan aktivitas sistem, serta penggunaan Web Application Firewall (WAF) menjadi krusial untuk mencegah ancaman berkelanjutan dari serangan jenis ini.

3.5 Reporting

1. Summary of Alerts

Summary of Alerts adalah ringkasan dari ancaman yang terdeteksi melalui pemindaian menggunakan OWASP-ZAP. Ringkasan ini mencakup berbagai jenis kerentanan yang ditemukan, tingkat keparahannya, serta detail terkait potensi risiko keamanan pada sistem atau aplikasi yang diuji. Dengan adanya ringkasan ini, pengguna dapat lebih mudah mengidentifikasi dan mengambil langkah yang diperlukan untuk memperbaiki serta meningkatkan keamanan sistem seperti yang ditunjukkan pada Tabel 2.

Tabel 2. Summary of Alert

Risk Level	Number of Alerts
High	1
Medium	4
Low	9
Informatioanal	7

Alerts menampilkan daftar jenis ancaman beserta tingkat risikonya. Selain itu, fitur ini juga menyajikan jumlah temuan untuk setiap ancaman yang berhasil teridentifikasi selama proses pemindaian menggunakan OWASP ZAP, sebagaimana diperlihatkan pada tabel 3

Tabel 3. Rincian Alert

Alert Type	Risk Level	Count
Vulnerable JS Library	High	1
CSP: Failure to Define Directive with No Fallback	Medium	118
CSP: Wildcard Directive	Medium	118
CSP: script-src unsafe-inline	Medium	118
CSP: style-src unsafe-inline	Medium	118
Application Error Disclosure	Low	1
Big Redirect Detected (Potential Sensitive Information Leak)	Low	1
Cookie No HttpOnly Flag	Low	148
Cookie Without Secure Flag	Low	266
Cookie without SameSite Attribute	Low	266
Cross-Domain JavaScript Source File Inclusion	Low	2163
Strict-Transport-Security Header Not Set	Low	10
Timestamp Disclosure – Unix	Low	121
X-Content-Type-Options Header Missing	Low	18
Information Disclosure - Suspicious Comments	Informational	116
Loosely Scoped Cookie	Informational	20
Modern Web Application	Informational	119
Re-examine Cache-control Directives	Informational	117
Retrieved from Cache	Informational	24
Session Management Response Identified	Informational	134
User Agent Fuzzer	Informational	1399
Total		21

Tabel 3. Menampilkan daftar nama ancaman yang terdeteksi selama pengujian serta jumlah kejadian dari setiap ancaman yang berisiko diretas atau mengalami pencurian data. Hasil pengujian menunjukkan bahwa jumlah kejadian tertinggi mencapai 2.163, dengan ancaman bernama Cross-Domain JavaScript Source File Inclusion. Ancaman ini tergolong dalam kategori Low, yang berarti memiliki tingkat risiko rendah. Berdasarkan rincian tersebut ditemukan kerentanan berbasis OWASP Top 10 seperti A03, A05, A06, A07, dan A09, yang selanjutnya akan dijelaskan melalui tabel 4.

Tabel 4. Rincian Alert

Kerentanan	Alert	Deskripsi	Dampak	Mitigasi
A03 Injection	Application Error Disclosure	Kerentanan ini terjadi ketika aplikasi mengungkapkan pesan kesalahan internal yang	Kebocoran informasi sensitif seperti detail server atau kode kesalahan, yang dapat dieksploitasi	Konfigurasi server untuk tidak menampilkan pesan kesalahan detail; gunakan logging internal dan tampilkan pesan kesalahan umum kepada pengguna.

DOI: https://doi.org/10.31004/riggs.v4i4.3686

		dapat memberikan informasi sensitif kepada penyerang.	untuk serangan lebih lanjut, menyebabkan kerugian finansial dan kerusakan reputasi.	
A05 Security Misconfiguration	CSP: Failure to Define Directive with No Fallback	Kerentanan ini muncul ketika Content Security Policy (CSP) tidak mendefinisikan direktif tertentu tanpa fallback, memungkinkan eksploitasi sumber daya tidak aman.	Peningkatan risiko serangan XSS atau injeksi skrip, yang dapat menyebabkan kebocoran data dan gangguan operasional.	Definisikan direktif CSP secara lengkap dengan fallback; gunakan header CSP yang ketat dan audit konfigurasi secara rutin.
A05 Security Misconfiguration	CSP: Wildcard Directive	Penggunaan wildcard (*) dalam direktif CSP memungkinkan sumber daya dari domain apa pun, yang tidak aman.	Memungkinkan akses tidak sah dari sumber eksternal, meningkatkan risiko injeksi dan kebocoran data sensitif.	Hindari wildcard; tentukan domain spesifik dalam direktif CSP dan lakukan pengujian keamanan berkala.
A05 Security Misconfiguration	CSP: script-src unsafe-inline	Direktif script-src mengizinkan eksekusi skrip inline, yang rentan terhadap XSS.	Potensi serangan XSS yang dapat mencuri data pengguna atau mengubah konten halaman, menyebabkan kerusakan reputasi.	Hilangkan unsafe-inline; gunakan nonce atau hash untuk skrip inline dan migrasikan ke file eksternal.
A05 Security Misconfiguration	CSP: style-src unsafe-inline	Direktif style-src mengizinkan gaya inline, yang dapat dieksploitasi untuk injeksi CSS.	Risiko injeksi CSS yang dapat mengubah tampilan atau mencuri data, mengganggu pengalaman pengguna.	Hindari unsafe-inline untuk style; gunakan nonce atau file eksternal dan audit CSP secara rutin.
A05 Security Misconfiguration	Cookie No HttpOnly Flag	Cookie tanpa flag HttpOnly dapat diakses oleh JavaScript, rentan terhadap XSS.	Kebocoran cookie sesi melalui serangan XSS, menyebabkan pencurian identitas dan akses tidak sah.	Setel flag HttpOnly pada cookie; pastikan konfigurasi server mendukungnya dan lakukan pengujian.
A05 Security Misconfiguration	Cookie Without Secure Flag	Cookie tanpa flag Secure dapat dikirim melalui HTTP, bukan hanya HTTPS.	Risiko pencurian cookie melalui man-in-the-middle attacks, menyebabkan kebocoran data sensitif.	Setel flag Secure pada cookie; paksa penggunaan HTTPS dan audit konfigurasi cookie.
A05 Security Misconfiguration	Cookie without SameSite Attribute	Cookie tanpa atribut SameSite rentan terhadap CSRF.	Potensi serangan CSRF yang dapat menjalankan aksi tanpa izin pengguna, menyebabkan kerugian finansial.	Setel SameSite=Strict atau Lax; uji kompatibilitas browser dan perbarui konfigurasi.
A05 Security Misconfiguration	Strict-Transport-Security Header Not Set	Header HSTS tidak diatur, memungkinkan downgrade ke HTTP.	Risiko man-in-the-middle attacks dan kebocoran data melalui koneksi tidak aman.	Setel header HSTS dengan max-age yang panjang; sertakan subdomains dan preload jika memungkinkan.
A05 Security Misconfiguration	X-Content-Type-Options Header Missing	Header ini tidak diatur, memungkinkan MIME-sniffing.	Risiko eksekusi skrip berbahaya melalui file yang salah ditafsirkan,	Setel header X-Content-Type-Options: nosniff; audit respons server secara rutin.

DOI: https://doi.org/10.31004/riggs.v4i4.3686

			menyebabkan XSS.	
A06 Vulnerable and Outdated Components	Vulnerable JS Library	Library JavaScript yang rentan (misalnya jQuery DataTables 1.10.16) dengan CVE-2020-28458 dan CVE-2021-23445.	Potensi eksploitasi seperti injeksi atau remote code execution, menyebabkan kebocoran data dan kerusakan sistem.	Perbarui library ke versi terbaru (minimal 1.10.23 atau 1.11.3); lakukan scan rutin untuk komponen rentan.
A07 Identification and Authentication Failures	Big Redirect Detected (Potential Sensitive Information Leak)	Redirect besar yang berpotensi membocorkan informasi sensitif.	Kebocoran data sensitif melalui redirect, menyebabkan pencurian identitas.	Validasi dan sanitasi URL redirect; gunakan redirect relatif jika memungkinkan.
A07 Identification and Authentication Failures	Session Management Response Identified	Respons yang mengidentifikasi pengelolaan sesi, berpotensi membocorkan info.	Risiko pencurian sesi atau identitas, menyebabkan akses tidak sah.	Gunakan token sesi aman; terapkan rotasi sesi dan logging akses.
A09 Security Logging and Monitoring Failures	Information Disclosure - Suspicious Comments	Komentar mencurigakan dalam kode sumber yang membocorkan info.	Kebocoran info internal seperti detail server, memudahkan serangan targeted.	Hapus komentar sensitif dari kode produksi; lakukan code review.
A09 Security Logging and Monitoring Failures	Timestamp Disclosure - Unix	Pengungkapan timestamp Unix yang dapat membantu serangan timing.	Membantu penyerang dalam serangan brute-force atau enumerasi.	Sanitasi output untuk menghapus timestamp; gunakan format acak jika diperlukan.
A09 Security Logging and Monitoring Failures	Re-examine Cache-control Directives	Direktif cache-control tidak diatur dengan benar, memungkinkan caching sensitif.	Kebocoran data melalui cache browser atau proxy.	Setel no-cache, no-store untuk konten sensitif; audit header cache.
A09 Security Logging and Monitoring Failures	Retrieved from Cache	Respons diambil dari cache, berpotensi membocorkan data lama.	Risiko akses data kadaluarsa atau sensitif dari cache.	Gunakan header cache-control yang tepat; nonaktifkan caching untuk data sensitif.
A09 Security Logging and Monitoring Failures	Loosely Scoped Cookie	Cookie dengan scope longgar, dapat dibagikan antar subdomain.	Risiko pencurian cookie melalui subdomain berbahaya.	Setel domain cookie secara spesifik; gunakan flag Secure dan HttpOnly.
A09 Security Logging and Monitoring Failures	Cross-Domain JavaScript Source File Inclusion	Inklusi file JS dari domain eksternal, rentan terhadap manipulasi.	Potensi injeksi skrip dari sumber tidak tepercaya, menyebabkan XSS.	Gunakan SRI (Subresource Integrity) untuk file eksternal; batasi domain.
A09 Security Logging and Monitoring Failures	User Agent Fuzzer	Respons terhadap fuzzer User-Agent, menunjukkan potensi identifikasi browser.	Membantu penyerang dalam targeted attacks berdasarkan UA.	Standarisasi respons terlepas dari UA; gunakan logging untuk mendeteksi fuzzer.
A09 Security Logging and Monitoring Failures	Modern Web Application	Aplikasi web modern dengan potensi kerentanan JS.	Risiko serangan client-side seperti XSS atau CSRF.	Terapkan CSP ketat; lakukan audit JS dan update framework.

4. Kesimpulan

Penelitian ini menunjukkan bahwa pengujian kerentanan menggunakan OWASP ZAP pada website Sistem Informasi Akademik Fakultas Kedokteran Universitas Negeri Surabaya berhasil mengidentifikasi berbagai celah keamanan yang berpotensi menimbulkan risiko terhadap kerahasiaan, integritas, dan ketersediaan data. Berdasarkan hasil automated scanning dan analisis lanjutan, ditemukan total 21 jenis alert dengan tingkat risiko

DOI: https://doi.org/10.31004/riggs.v4i4.3686

bervariasi, mulai dari High, Medium, Low hingga Informational. Temuan dengan tingkat risiko tertinggi berada pada kategori Vulnerable JavaScript Library, sedangkan jumlah kejadian terbesar terdapat pada Cross-Domain JavaScript Source File Inclusion dengan 2.163 insiden. Selain itu, beberapa kerentanan yang termasuk dalam OWASP Top 10 seperti A03 Injection, A05 Security Misconfiguration, A06 Vulnerable and Outdated Components, A07 Identification and Authentication Failures, dan A09 Security Logging and Monitoring Failures turut ditemukan pada website yang diuji. Hasil ini menegaskan bahwa konfigurasi keamanan website masih memerlukan perbaikan signifikan, khususnya pada penerapan Content Security Policy (CSP), pengelolaan cookie, pembaruan library, serta pengaturan header keamanan seperti HSTS dan X-Content-Type-Options. Rekomendasi mitigasi yang disampaikan OWASP ZAP dapat dijadikan acuan untuk memperkuat sistem, terutama terkait pencegahan serangan XSS, CSRF, kebocoran informasi, dan potensi eksploitasi komponen usang. Penelitian ini juga membuktikan bahwa OWASP ZAP efektif sebagai alat pendeteksi otomatis yang mampu memberikan gambaran komprehensif mengenai status keamanan aplikasi web. Namun demikian, penelitian ini memiliki keterbatasan karena hanya menggunakan pendekatan black-box scanning tanpa melakukan validasi manual lanjutan atau uji penetrasi tingkat lanjut. Untuk penelitian berikutnya, disarankan mengombinasikan pengujian otomatis dengan analisis manual serta memperluas ruang lingkup pengujian ke modul-modul lain pada sistem akademik. Dengan langkah tersebut, evaluasi keamanan dapat dilakukan secara lebih mendalam sehingga penguatan sistem dapat dilakukan secara berkelanjutan dan lebih menyeluruh.

Referensi

1. Akbar, D. F., Nugraha, L. R. S. R., & Aldi, P. N. (2025). Penerapan Black Box Testing Menggunakan Teknik Equivalence Partitioning pada SIAKAD UKRI. Journal on Pustaka Informatika.
2. Al Muzaki, M., Perdian, R. Z., Fajar, R., Khofifah, S., & Atmaja, S. A. (2025). Analisis Kerentanan Web Menggunakan ZAP oleh Checkmarx pada Situs Kuliah Daring LMS Universitas Kebangsaan Republik Indonesia: Penelitian. Journal on Pustaka Cendekia Informatika, 3(1), 125-132.
3. Armis (2024) The State of Cybersecurity in Education. media.armis.com
4. Awlarijal, A.N., Almaarif, A. and Budiono, A. (2020) 'Vulnerability Assessment for Basic Data of Education Website in Regional Government X – A Black Box Testing Approach', (Andress 2014), pp. 163–168.
5. Cycognito (2024) 'State of Web Application Security Testing. cycognito.com
6. IBM (2024) 'X-Force Threat Intelligence Index'. ibm.com
7. Jakobsson, A. and Häggström, I. (2022) 'Study of the techniques used by OWASP ZAP for analysis of vul- nerabilities in web applications'. diva-portal.org
8. Kadir, F. M., Irsan, M., & Putrada, A. G. (2025). Benchmarking Mobile Apps Security in Universities: An OWASP Mobile Top 10 Framework Perspective. International Journal on ICT.
9. Lallie, H.S. and Titis, E. (2023) 'Understanding Cyber Threats Against the Universities, Colleges, and Schools'. DOI: 10.3390/computers14020049
10. Maniraj, S.P., Ranganathan, C.S. and Sekar, S. (2024) 'SECURING WEB APPLICATIONS WITH OWASP ZAP FOR COMPREHENSIVE SECURITY TESTING', 10(2), pp. 12–23. DOI: 10.29284/ijasis.10.2.2024.12-23.
11. Meucci, M. and Muller, A. (2014) 'OWASP Testing Guide 4.0', (Cc). apriorit.com
12. Putra, F. P. E. ., Ubaidi, U., Hamzah, A. ., Pramadi, W. A. ., & Nuraini, A. . (2024). Systematic Literature Review: Security Gap Detection On Websites Using Owasp Zap . Brilliance: Research of Artificial Intelligence, 4(1), 348–355. 10.4. DOI: 10.47709/brilliance.v4i1.4227
13. Sembiring, E. (2025). Implementation of the Incoming and Outgoing Mail Management Information System Application at STMIK Neumann Indonesia. Jurnal Komputer Teknologi Informasi Sistem.
14. Utama, F.P., Muhamad, R. and Nurhadi, H. (2024) 'Uncovering the Risk of Academic Information System Vulnerability through PTES and OWASP Method', 18(1), pp. 39–51. pdfs.semanticscholar.org
15. Ventura, R., Franco, D.J. and Akram, O.K. (2023) 'A N OVEL VAPT A LGORITHM : E NHANCING W EB A PPLICATION S ECURITY T ROUGH OWASP T OP 10 O PTIMIZATION', pp. 13–27. DOI: 10.5121/csit.2023.132002.
16. Verizon (2024) '2024 Data Breach Investigations Report'. verizon.com

DOI: https://doi.org/10.31004/riggs.v4i4.3686