JATIKOM: Jurnal Aplikasi dan Teori Ilmu Komputer
ISSN: 2615-577X (Online)

Analisis Kerentanan Menggunakan Dynamic Application Security Testing (DAST) Berdasarkan Pedoman OWASP Pada Situs Website Eprints Universitas Muhammadiyah Malang

Muhammad Zakiy Alfaris[#1]

Teknik Informatika
Universitas Muhammadiyah Malang
Indonesia

[1] zakiy.alfaris22@gmail.com

Abstract*—The security of information systems is very important in digital data management, especially on repository websites such as Eprints UMM. This research aims to identify security vulnerabilities in Eprints UMM in terms of Data Validation Testing and Error Handling using the testing tools Zed Attack Proxy (ZAP) and Arachni. The testing results reveal weaknesses in input validation that are susceptible to SQL Injection and Cross-Site Scripting (XSS) attacks, as well as error handling that may expose sensitive information. To improve security, it is recommended that the Eprints UMM administrators enhance input validation, secure error handling mechanisms, and regularly update both software and hardware.

***Key Words*— OWASP, DAST Methods, ZAP, Arachni, Mitigation**

Abstrak— **Keamanan sistem informasi sangat penting dalam pengelolaan data digital, terutama pada website repositori seperti Eprints UMM. Penelitian ini bertujuan mengidentifikasi kerentanan keamanan pada Eprints UMM dalam aspek Data Validation Testing dan Error Handling menggunakan alat uji Zed Attack Proxy (ZAP) dan Arachni. Hasil pengujian menunjukkan kelemahan dalam validasi input yang berisiko terhadap serangan SQL Injection dan Cross-Site Scripting (XSS), serta penanganan pesan error yang dapat mengungkap informasi sensitif. Untuk meningkatkan keamanan, disarankan pengelola Eprints UMM memperbaiki validasi input, mengamankan mekanisme penanganan kesalahan, serta rutin memperbarui perangkat lunak dan perangkat keras. Dengan langkah ini, risiko serangan siber dapat diminimalkan, dan keandalan sistem dapat meningkat.**

***Kata kunci*— OWASP, Metode DAST, ZAP, Arachni, Mitigasi**

I. PENDAHULUAN

Perkembangan teknologi informasi telah mengakselerasi transformasi perpustakaan dari format fisik ke digital, sehingga memungkinkan akses informasi yang lebih luas dan efisien. Berbagai institusi pendidikan, termasuk Universitas Muhammadiyah Malang (UMM), telah mengadopsi sistem repositori digital seperti Eprints guna mengelola publikasi ilmiah [1]. Namun, revolusi digital ini juga membawa tantangan tersendiri, terutama dalam aspek keamanan informasi. Peningkatan serangan siber yang menargetkan server berperan sebagai penggerak utama dalam akses situs web dan basis data menimbulkan risiko yang signifikan, apabila serangan tersebut tidak segera ditangani [2][3].

Observasi awal terhadap situs Eprints UMM mengungkap sejumlah kendala, seperti kehilangan data akibat pemadaman listrik dan penggunaan server yang belum diperbarui. Kondisi tersebut menekankan perlunya perbaikan tata kelola Teknologi Informasi dan Komunikasi (TIK) khususnya dalam pemenuhan aspek keamanan (security), integritas (integrity), dan ketersediaan (availability) [4]. Upaya penguatan keamanan ini menjadi penting mengingat peran vital sistem digital dalam mendukung proses akademis dan publikasi ilmiah.

Untuk mengidentifikasi dan mengevaluasi tingkat keamanan situs web Eprints UMM, penelitian ini menerapkan metode Dynamic Application Security Testing (DAST). Metode DAST memungkinkan pengujian dilakukan secara eksternal terhadap sistem yang sedang berjalan, tanpa perlu mengakses kode sumber secara langsung [5]. Pendekatan ini berbeda dengan metode SAST (Static Application Security Testing) dan IAST (Interactive Application Security Testing) yang memiliki keunggulan dan keterbatasan masing-masing [6][7][8].

Agar pengujian keamanan lebih sistematis dan komprehensif, penelitian ini mengacu pada pedoman OWASP Top 10 dalam menilai risiko serta kerentanan aplikasi web [9][10][11]. Dengan menerapkan pengujian berbasis DAST sesuai standar OWASP Top 10, diharapkan UMM dapat meningkatkan keamanan situs Eprints, menjaga integritas data, dan memberikan akses yang lebih aman serta andal bagi pengguna

.

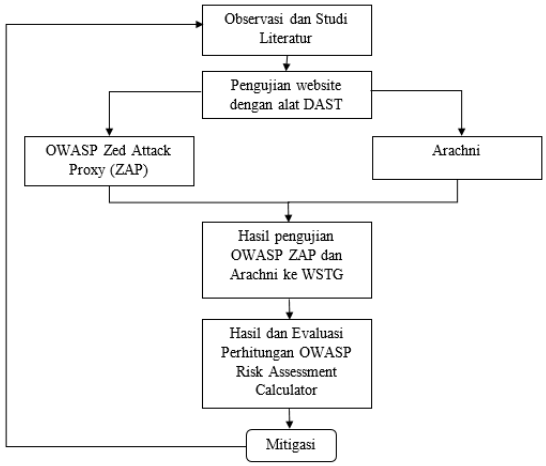
TABEL 1. ALAT UJI, LANGKAH PENGUJIAN, DAN PERANGKAT

Alat Pengujian	Deskripsi	Langkah Pengujian	Perangkat Keras	Perangkat Lunak
OWASP ZAP	OWASP ZAP merupakan alat pengujian keamanan web sumber terbuka yang mampu mendeteksi kerentanan secara otomatis [4][7] [12].	Mengunduh ZAP, memilih *automated scan*, memasukkan URL target, mengaktifkan Ajax Spider, dan menjalankan serangan.	Laptop dengan RAM 16 GB, ROM 512 GB SSD, dan prosesor Intel Core i5 Gen 10.	*Virtual Machine* Kali Linux.
Arachni	Arachni adalah alat uji DAST yang dapat mengidentifikasi celah keamanan seperti SQL Injection, XSS, dan Command Injection [4].	Mengunduh Arachni, masuk ke sistem, membuat sesi pemindaian baru, memasukkan URL target, dan menjalankan pemindaian.	Laptop dengan RAM 16 GB, ROM 512 GB SSD, dan prosesor Intel Core i5 Gen 10.	*Virtual Machine* Kali Linux.

II. METODE

Penelitian ini dilaksanakan melalui empat tahapan utama, yaitu: (i) Observasi dan studi literatur, (ii) Pengujian website menggunakan DAST dengan alat OWASP ZAP dan Arachni, (iii) Analisis hasil pengujian berdasarkan pedoman Web Security Testing Guide (WSTG) dan perhitungan risiko menggunakan OWASP Risk Assessment Calculator, serta (iv)Mitigasi.

Seluruh tahapan tersebut digambarkan secara struktural pada Gambar 1.



Gambar 1. Struktur Penelitian

A. Observasi dan Studi Literatur

Penelitian diawali dengan observasi mendalam terhadap website Eprints Universitas Muhammadiyah Malang (UMM) dan wawancara dengan kepala pengembang sebagai upaya untuk mengonfirmasi temuan awal. Berdasarkan studi literatur terdahulu, terdapat kebutuhan mendesak untuk menyempurnakan proses pengujian keamanan web, terutama dalam hal penyediaan rekomendasi perbaikan, konsistensi penilaian risiko, dan penjelasan rinci mengenai alat uji.

Beberapa penelitian terkait telah mengemukakan bahwa kombinasi penggunaan Fortify, Arachni, dan OWASP ZAP menghasilkan tingkat false positif yang rendah [7]. Di sisi lain, integrasi metode SAST, DAST, dan IAST belum sepenuhnya terjangkau, seperti terlihat pada studi Setiawan et al. [8], yang meskipun mampu mengidentifikasi sejumlah risiko, tidak menyertakan langkah mitigasi. Studi lainnya, seperti yang dilakukan oleh Sunardi et al. [13] dan Vibhandik et al. [14], menyoroti berbagai kerentanan dan kekurangan rekomendasi perbaikan, sedangkan Shahid et al. [15] menekankan keunggulan OWASP ZAP dalam hal deteksi.

B. Identifikasi Website Eprints

Repositori institusional Eprints UMM berfungsi sebagai platform untuk menyimpan dan mendistribusikan publikasi akademik dalam format web yang dapat diakses melalui protokol HTTP atau HTTPS [16]. Baru-baru ini, situs ini mengalami gangguan berupa kehilangan data mahasiswa yang diakibatkan oleh pemadaman listrik, sehingga mengganggu akses terhadap tugas akhir dan karya ilmiah. Selain itu, penggunaan server dengan versi lama meningkatkan potensi gangguan serta risiko keamanan yang signifikan.

C. Pengujian Website dengan Alat DAST (Dynamic Application Security Testing)

Alat uji kerentanan yang digunakan dalam penelitian ini adalah OWASP ZAP dan Arachni. Kedua alat tersebut telah terbukti memiliki tingkat deteksi yang baik berdasarkan studi literatur. (i) OWASP ZAP: Dikembangkan oleh OWASP, merupakan alat open-source yang mampu mendeteksi kerentanan secara otomatis, seperti SQL Injection, Cross-Site Scripting (XSS), dan lain-lain [17][18][19][20]. (ii) Arachni: Diciptakan oleh Tasos Laskos, alat ini didesain untuk mengidentifikasi berbagai celah keamanan sesuai dengan standar OWASP Top Ten.

Pengujian dilakukan secara sistematis pada website eprints.umm.ac.id dengan mempertimbangkan dua tolak ukur dari OWASP versi 4, yaitu: (i) Data Validation Testing dan (ii) Error Handling.

Proses pengujian dimulai dari pemindaian, dilanjutkan dengan analisis, dan diakhiri dengan pelaporan hasil guna memastikan website aman dari potensi eksploitasi. Rincian langkah pengujian dan perangkat yang digunakan dapat dilihat pada Tabel 1.

D. Skenario Pengujian OWASP ZAP dan Arachni ke WSTG

OWASP merupakan organisasi nirlaba yang menyediakan framework pengujian keamanan secara gratis, salah satunya adalah Web Security Testing Guide (WSTG). WSTG versi terbaru pada tahun 2021 mencakup 12 tolak ukur yang meliputi berbagai aspek, seperti: (i) Information Gathering, (ii) Authentication & Authorization Testing, (iii) Session Management, (iv) Data Validation, (v) Error Handling, (vi) API Testing [17][18].

Dalam penelitian ini, hasil pemindaian dari Zed Attack Proxy (ZAP) dan Arachni dianalisis berdasarkan pedoman WSTG, dengan fokus pada dua aspek utama:

(a) A07:2021 - Data Validation Testing
- Meliputi 20 kriteria pengujian.
- Menganalisis kerentanan SQL Injection dan Cross-Site Scripting (XSS) pada halaman detail setelah login.
- Hasil pengujian dikategorikan sebagai Pass (aman), Issue (ditemukan celah), atau N/A (tidak diuji) [19].

(b) A08:2021 - Error Handling
- Terdapat 2 kriteria pengujian.
- Menguji apakah aplikasi mampu menangani kesalahan dengan aman tanpa mengungkapkan informasi sensitif.
- Hasil pengujian dikategorikan sebagai Pass (aman), Issue (ditemukan celah), atau N/A (tidak diuji) [19].

E. Analisa Perhitungan WSTG OWASP Risk Assessment Calculator

Pada tahap ini, setiap celah keamanan yang diidentifikasi dinilai menggunakan Risk Assessment Calculator dari WSTG. Penilaian dilakukan berdasarkan dua faktor utama, yaitu:

(a) Likelihood Factors (Faktor Kemungkinan)
- Threat Agent Factors: Menilai kemampuan, motivasi, akses, dan jumlah penyerang potensial.
- Vulnerability Factors: Mengukur kemudahan penemuan, eksploitasi, kesadaran, dan deteksi terhadap celah.

(b) Impact Factors (Faktor Dampak)
- Technical Impact: Menilai dampak terhadap kerahasiaan, integritas, ketersediaan, dan akuntabilitas data.
- Business Impact: Mengukur potensi kerugian finansial, reputasi, kepatuhan, dan pelanggaran privasi.

3	**Likelihood factors**			**Impact factors**			
4	***Threat Agent Factors***			***Technical Impact Factors***			
5	Skills required	Network and programming skills [6]		Loss of confidentiality	Extensive critical data disclosed [7]	7	
6	Motive	Low or no reward [1]		Loss of Integrity	Minimal slightly corrupt data [1]	1	
7	Opportunity	Some access or resources required [7]		Loss of Availability	Minimal primary services interrupted [5]	5	
8	Population Size	System Administrators [2]		Loss of Accountability	Attack possibly traceable to individual [7]	7	
9							
10	***Vulnerability Factors***			***Business Impact Factors***			
11	Easy of Discovery	Difficult [3]		Financial damage	Damage costs less than to fix the issue [1]	1	
12	Ease of Exploit	Difficult [3]		Reputation damage	Not Applicable [0]	0	
13	Awareness	Public knowledge [9]		Non-Compliance	Minor violation [2]	2	
14	Intrusion Detection	Active detection in application [1]		Privacy violation	Hundreds of people [5]	5	
15							
16	**Likelihood score:**	**4**		**Impact score:**	**3.5**		

Gambar 2. Perhitungan Rata-Rata Faktor Kemungkinan dan Faktor Dampak

Setiap faktor dinilai dalam rentang 0-9, di mana nilai rata-rata untuk likelihood dan impact dihitung, kemudian dijumlahkan dan dibagi dua untuk mendapatkan skor rata-rata keseluruhan. Hasil perhitungan tersebut digunakan untuk menentukan tingkat kerentanan, seperti yang ditunjukkan pada Gambar 3.

Likelihood and Impact Levels	
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

Gambar 3. Tingkat Kerentanan Faktor Kemungkinan Dan Faktor Dampak

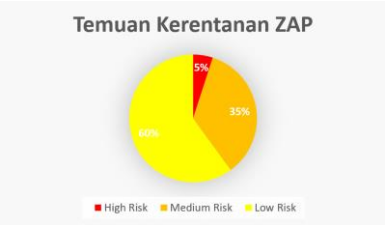
F. Mitigasi

Tahap mitigasi difokuskan pada penanganan ancaman dengan risiko tinggi, khususnya terkait dengan SQL Injection dan XSS. Perbaikan prioritas diarahkan pada celah-celah keamanan dengan skor risiko tertinggi guna meminimalkan potensi eksploitasi. Langkah mitigasi dilakukan dengan menerapkan solusi teknis dan administratif untuk memastikan perlindungan yang optimal terhadap data serta menjaga kelangsungan akses bagi pengguna.

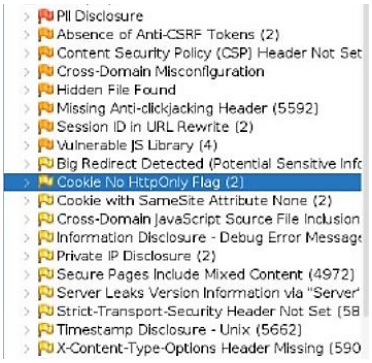
III. HASIL DAN PEMBAHASAN

A. Hasil Pengujian ZAP

Pengujian menggunakan metode Spider, Ajax Spider, dan Active Scan pada aplikasi mengidentifikasi 1 kerentanan dengan tingkat risiko tinggi (5%), 7 kerentanan dengan risiko sedang (35%), serta 12 kerentanan dengan risiko rendah (60%). Gambar 4 menggambarkan persentase temuan, sedangkan Gambar 5 menampilkan detail masing-masing kerentanan dengan indikator yang menggunakan warna merah (tinggi), oranye (sedang), dan kuning (rendah).



Gambar 4. Persentase Temuan Kerentanan Menggunakan ZAP



Gambar 5. Kerentanan Temuan ZAP

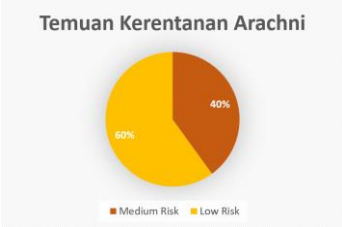
TABEL 2. HASIL UJI DATA VALIDATION TESTING ZAP DAN ARACHNI

Kode	Pengujian	Deskripsi	Alat	Bagian Terdampak
WSTG-INPV-01	Reflected XSS	Menguji kemungkinan XSS	ZAP, Firefox	CSP header not set, X-Content-Type-Header Missing
WSTG-INPV-04	HTTP Parameter Pollution	Menguji polusi parameter HTTP	ZAP, Firefox, Arachni	Big Redirect Detected, common sensitive file
WSTG-INPV-05	SQL Injection	Menguji keberadaan injeksi SQL	ZAP, Firefox	PII disclosure, CSP header not set
WSTG-INPV-11	Code Injection	Menguji kemungkinan injeksi kode	ZAP, Firefox	Vulnerable JS Library
WSTG-INPV-14	Incubated Vulnerability	Memeriksa incubated vulnerability	ZAP, Firefox	Hidden File Finder
WSTG-INPV-19	SSRF	Menguji kemungkinan SSRF	ZAP, Firefox	Private IP Disclosure

B. Hasil Pengujian Arachni

Pemindaian menggunakan Arachni pada halaman sebelum login Eprints UMM menemukan 2 kerentanan yang dikategorikan berisiko sedang dan 3 kerentanan berisiko rendah. Temuan tersebut mencakup masalah pada konfigurasi keamanan, antara lain: (i) HTTP Strict Transport Security (HSTS) tidak diterapkan, (ii) Direktori yang tidak terpakai namun dapat diakses, (iii) Fitur pengisian otomatis sandi yang aktif, (iv) Tidak adanya header keamanan X-Frame-Options.

Detail persentase dan temuan lebih lanjut ditampilkan pada Gambar 6 dan Gambar 7, dengan indikator warna coklat untuk risiko sedang dan oranye untuk risiko rendah guna memudahkan identifikasi.



Gambar 6. Persentase Temuan Kerentanan Menggunakan Arachni

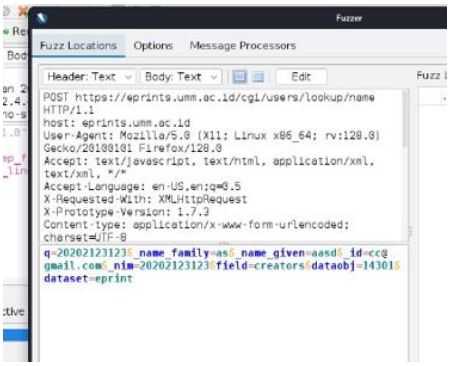
Gambar 7. Kerentanan Temuan Arachni

C. Hasil Pengujian ZAP dan Arachni ke WSTG

(a) Data Validation Testing

Pengujian, baik secara otomatis maupun manual, dengan memanfaatkan ZAP dan Arachni, mengungkapkan berbagai kerentanan terkait validasi input. Di antaranya terdapat celah yang mengindikasikan kemungkinan terjadinya XSS, SQL Injection, dan SSRF. Tabel 2 menyajikan hasil rinci pengujian ini.

Pengujian dengan memasukkan karakter acak pada input (POST) mengungkapkan adanya parameter yang rentan terhadap SQL Injection dan XSS. Hal ini berpotensi menimbulkan kebocoran data pengguna. Gambar 8, Gambar 9, dan Gambar 10 menunjukkan: (i) Gambar 8: Parameter input yang terdampak, (ii) Gambar 9: Indikasi terjadinya SQL Injection, (iii) Gambar 10: Indikasi terjadinya Cross-Site Scripting (XSS).



Gambar 8. Parameter Input

Gambar 9. SQL Injection

Gambar 10. Cross-Cite Scripting (XSS)

Dari pengujian tersebut, Gambar 8 menunjukkan parameter yang rentan, sementara Gambar 9 dan Gambar 10 mengungkap potensi kebocoran informasi, seperti nama lengkap dan alamat email pengguna yang seharusnya tetap terlindungi.

(b) Error Handling

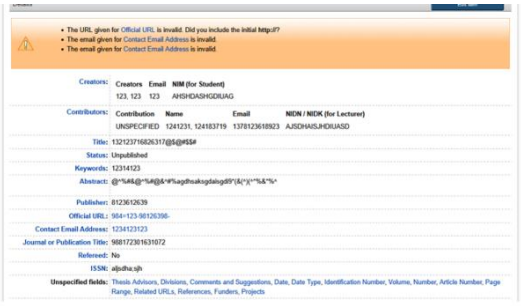
Pengujian mengenai Error Handling pada platform Eprints UMM dilakukan dengan menggunakan ZAP, Arachni, serta metode manual. Tujuannya adalah mengevaluasi respons sistem terhadap error untuk mencegah terjadinya kebocoran informasi sensitif. Tabel 3 menyajikan hasil pengujian tersebut.

TABEL 3. HASIL UJI ERROR HANDLING

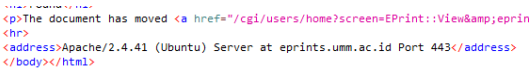
Kode	Pengujian	Deskripsi	Alat	Bagian Terdampak
WSTG-ERRH-01	Improper Error Handling	Menganalisis respons error	ZAP, Firefox	Debug error messages

Pengujian manual dilakukan dengan mengirimkan script injeksi SQL dan XSS melalui halaman login (Gambar 11). Hasil pengujian menunjukkan bahwa sistem mampu menangani error dengan baik. Selanjutnya, pada halaman input setelah login (Gambar 12), meskipun seluruh karakter diterima tanpa penyaringan, tidak terjadi pengungkapan informasi sensitif. Namun, server tetap menampilkan informasi jenis server, yakni Apache/2.4.41 (Ubuntu) (Gambar 13), yang seharusnya disembunyikan untuk menghindari potensi eksploitasi.

Gambar 11. Pengujian Error Handling



Gambar 12. Input Karakter Unik

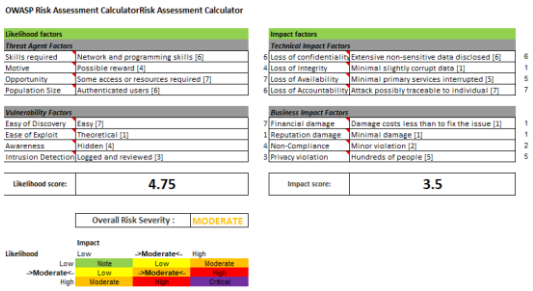


Gambar 13. Respon Jenis Server

D. Hasil dan Evaluasi Perhitungan Kalkulator Kerentanan OWASP WSTG

(a) Perhitungan Kalkulator Kerentanan Hasil Pengujian ZAP

Analisis perhitungan risiko menggunakan OWASP Risk Assessment Calculator menunjukkan bahwa rata-rata tingkat kerentanan berdasarkan pengujian dengan ZAP berada pada kategori MODERATE (menengah). Hal ini diindikasikan oleh nilai "ease of discovery" yang tinggi, mengingat kerentanan seperti kebocoran data (nama lengkap dan email) dapat dengan mudah diidentifikasi menggunakan ZAP.



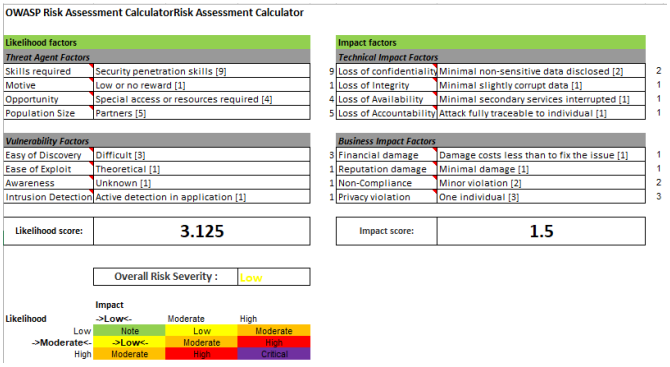
Gambar 14. Hasil Perhitungan ZAP

(b) Perhitungan Kalkulator Kerentanan Hasil Pengujian Arachni

Sementara itu, perhitungan risiko berdasarkan hasil pengujian Arachni menunjukkan rata-rata tingkat kerentanan berada pada kategori LOW (rendah). Hal ini terjadi karena tingkat kerusakan reputasi (reputation damage) relatif minimal, mengingat website tidak mengelola data sensitif seperti informasi keuangan atau medis.

TABEL 4. MITIGASI DATA VALIDATION TESTING

No	Jenis Kerentanan	Mitigasi	Level
1	**PII Disclosure**	Sembunyikan data pribadi, gunakan enkripsi, kontrol akses ketat	High
2	**CSP Header Not Set**	Aktifkan dan terapkan CSP header	Medium
3	**Vulnerable JS Library**	Perbarui atau hapus pustaka rentan	Medium
4	**Hidden File Finder**	Hapus atau atur izin file sensitif	Medium
5	**Common Directory**	Batasi akses dengan firewall atau .htaccess	Medium
6	**Big Redirect Detected**	Hindari redirect tidak perlu, gunakan enkripsi	Low
7	**Cross-Domain JS Inclusion**	Kontrol sumber daya eksternal, batasi domain di CSP	Low
8	**X-Content-Type-Options Missing**	Tambahkan header *nosniff* pada server	Low
9	**Common Sensitive File**	Hapus atau amankan file konfigurasi	Low


OWASP Risk Assessment CalculatorRisk Assessment Calculator

Likelihood factors			Impact factors		
Threat Agent Factors			*Technical Impact Factors*		
Skills required	Security penetration skills [9]	9	Loss of confidentiality	Minimal non-sensitive data disclosed [2]	2
Motive	Low or no reward [1]	1	Loss of Integrity	Minimal slightly corrupt data [1]	1
Opportunity	Special access or resources required [4]	4	Loss of Availability	Minimal secondary services interrupted [1]	1
Population Size	Partners [5]	5	Loss of Accountability	Attack fully traceable to individual [1]	1
Vulnerability Factors			*Business Impact Factors*		
Easy of Discovery	Difficult [3]	3	Financial damage	Damage costs less than to fix the issue [1]	1
Ease of Exploit	Theoretical [1]	1	Reputation damage	Minimal damage [1]	1
Awareness	Unknown [1]	1	Non-Compliance	Minor violation [2]	2
Intrusion Detection	Active detection in application [1]	1	Privacy violation	One individual [3]	3
Likelihood score:	3.125		Impact score:	1.5	

Overall Risk Severity : Low

Likelihood	Impact ->Low<-	Moderate	High
Low	Note	Low	Moderate
->Moderate<-	->Low<-	Moderate	High
High	Moderate	High	Critical

Gambar 15. Hasil Perhitungan Arachni

E. Mitigasi

Langkah mitigasi difokuskan pada perbaikan kerentanan yang teridentifikasi pada aspek Data Validation Testing dan Error Handling, dengan prioritas utama diberikan pada temuan dengan tingkat risiko tinggi guna mencegah potensi eksploitasi selanjutnya.

(a) Mitigasi Data Validation Testing

Mitigasi dilakukan melalui penerapan validasi input yang ketat dan perlindungan terhadap serangan seperti SQL Injection, XSS, serta pencegahan kebocoran data. Tabel 4 menyajikan rekomendasi mitigasi pada masing-masing tingkat risiko: (i) Kerentanan tingkat tinggi harus segera diperbaiki untuk mencegah pencurian data, (ii) Kerentanan tingkat sedang perlu mendapatkan perhatian untuk menghindari risiko serangan di masa depan, (iii) Kerentanan tingkat rendah, meskipun dampaknya lebih kecil, tetap harus dilakukan perbaikan.

(b) Mitigasi Error Handling

Tindakan mitigasi pada aspek Error Handling dilakukan untuk mengurangi potensi kebocoran informasi melalui pesan error yang tidak dikelola dengan baik. Informasi lebih lanjut mengenai rekomendasi mitigasi dapat dilihat pada Tabel 5.

TABEL 5. MITIGASI ERROR HANDLING

Jenis Kerentanan	Mitigasi	Level
Information Disclosure - Debug Error Messages	Nonaktifkan debug di produksi, tampilkan pesan umum, simpan log error dengan aman	Low

Langkah tambahan untuk meningkatkan keamanan sistem, diantaranya: (i) Validasi Input: Terapkan validasi pada sisi klien dan sisi server, gunakan ekspresi reguler, dan lakukan penyaringan serta sanitasi input secara menyeluruh. (ii) Menyembunyikan Informasi Server: Konfigurasikan sistem agar tidak menampilkan versi server (misalnya, versi Apache) dan implementasikan Web Application Firewall (WAF) untuk perlindungan tambahan.

IV. KESIMPULAN

Evaluasi keamanan situs Eprints UMM menggunakan OWASP ZAP dan Arachni mengungkapkan celah signifikan dalam aspek Data Validation Testing dan Error Handling. Ketidakmampuan sistem dalam melakukan validasi input secara ketat menyebabkan kerentanan terhadap serangan injeksi, seperti SQL Injection dan Cross-Site Scripting (XSS), yang berpotensi mengakibatkan kebocoran data sensitif. Di samping itu, pesan kesalahan yang terlalu rinci memberikan informasi yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab untuk merancang serangan lebih lanjut.

Untuk meningkatkan keamanan sistem, direkomendasikan penerapan validasi input yang lebih menyeluruh pada tingkat klien dan server serta pengembangan mekanisme error handling yang tidak menampilkan informasi sensitif. Langkah-langkah perbaikan teknis juga mencakup penataan ulang konfigurasi keamanan server, dengan mengimplementasikan firewall, enkripsi data yang lebih canggih, dan melakukan pembaruan sistem secara berkala.

Implementasi rekomendasi mitigasi tersebut diharapkan dapat mengurangi risiko serangan siber, sehingga sistem menjadi lebih andal dan aman untuk mendukung aktivitas akademis dan publikasi ilmiah.

REFERENSI

[1] G. Nur Pramudyo, Z. Sintia Putri, I. Alim Prayogi, A. Mukti Sari, S. Widianah, and Y. Trisnawati, "Penerapan EPrint sebagai repositori institusi pada Perpustakaan Universitas Muhammadiyah Malang," *Khizanah al-Hikmah : Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan*, vol. 6, no. 1, pp. 12–19, 2018, doi: 10.24252/kah.v6i1a2.

[2] S. S.Kom, M.Kom, W. bayu Ahadin, and Z. ST, MT, "Visualisasi data attacker activity log portable modern honey network," *Jurnal Repositor*, vol. 4, no. 1, pp. 95–102, 2022, doi: 10.22219/repositor.v4i1.1446.

[3] I. A. Romadhan, S. Syaifudin, and D. R. Akbi, "Implementasi Multiple Honeypot pada Raspberry Pi dan Visualisasi Log Honeypot Menggunakan ELK Stack," *Jurnal Repositor*, vol. 2, no. 4, pp. 475–484, 2020, doi: 10.22219/repositor.v2i4.114.

[4] J. M. Akbar, "Penetration testing Website PT. Sekarlaut Tbk menggunakan Open Web Application Security Project (OWASP) standart top 10," 2024. [Online]. Available: https://eprints.umm.ac.id/id/eprint/7794/2/Bab

[5] G. W. Wahidin, S. Syaifuddin, and Z. Sari, "Analisis Ransomware Wannacry Menggunakan Aplikasi Cuckoo Sandbox," *Jurnal Repositor*, vol. 4, no. 1, pp. 83–94, 2022, doi: 10.22219/repositor.v4i1.1373.

[6] Y. Pan, "Interactive application security testing," in *Proceedings - 2019 International Conference on Smart Grid and Electrical Automation, ICSGEA 2019*, 2019, pp. 558–561. doi: 10.1109/ICSGEA.2019.00131.

[7] F. M. Tudela, J. R. B. Higuera, J. B. Higuera, J. A. S. Montalvo, and M. I. Argyros, "On combining static, dynamic and interactive analysis security testing tools to improve OWASP top ten security vulnerability detection in web applications," *Applied Sciences (Switzerland)*, vol. 10, no. 24, pp. 1–26, 2020, doi: 10.3390/app10249119.

[8] H. Setiawan, L. E. Erlangga, and I. Baskoro, "Vulnerability analysis using the Interactive Application Security Testing (IAST) approach for government x website applications," in *2020 3rd International Conference on Information and Communications Technology, ICOIACT 2020*, 2020, pp. 471–475. doi: 10.1109/ICOIACT50329.2020.9332116.

[9] Y. Ardiansah, "Analisis vulnerability Sistem Manajemen Tugas Akhir (Simanta) Universitas Muhammadiyah Malang," 2024. [Online]. Available: https://eprints.umm.ac.id/id/eprint/12991/2/BAB

[10] H. I. Perdhana, "Analisis dan mitigasi celah keamanan Website SIMPKN Informatika menggunakan Metode OWASP Zed Attack Proxy (ZAP)," 2024. [Online]. Available: https://eprints.umm.ac.id/id/eprint/5627/2/BAB

[11] S. Margareth and others, "Uji Penetration Testing Web Server XYZ, menggunakan Metode OWASP TOP 10 dan CVSS," 2024. [Online]. Available: https://conferences.ittelkom-pwt.ac.id/index.php/centive/article/view/400/321

[12] F. Noeraini, "Evaluasi Keamanan Website Dinas AB di Jawa Timur Terhadap Temuan dan Solusi Kerentanan," *Galang Tanjung*, pp. 1–9, 2023.

[13] Sunardi, I. Riadi, and P. A. Raharja, "Vulnerability analysis of E-voting application using open web application security project (OWASP) framework," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 11, pp. 135–143, 2019, doi: 10.14569/IJACSA.2019.0101118.

[14] R. Vibhandik and A. K. Bose, "Vulnerability assessment of web applications-a testing approach," in *2015 4th International Conference on e-Technologies and Networks for Development, ICeND 2015*, 2015, pp. 16–21. doi: 10.1109/ICeND.2015.7328531.

[15] 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, 2022, doi: 10.3390/app12084077.

[16] S. Andriyani, M. F. Sidiq, and B. P. Zen, "Analisis celah keamanan pada website dengan menggunakan metode penetration testing dan framework issaf pada Website SMK Al-Kautsar," *Journal Informatic and Information Technology*, vol. 8798, pp. 1–13, 2023.

[17] A. Alkatiri, "Analisis celah Keamanan dan monitoring website menggunakan OWASP Zed Attack Proxy (ZAP) \& WAZUH (Studi Kasus: Website Dukcapil Kab. Nganjuk)," 2024. [Online]. Available: https://eprints.umm.ac.id/id/eprint/13016/49/BAB

[18] OWASP, "Testing guide 4.0," 2014. [Online]. Available: http://www.owasp.org

[19] Fauzan Rivaldo Sukardi, "Evaluasi keamanan manajemen persuratan berbasis website menggunakan framework OWASP Web Security Testing Guide (WSTG)," 2024. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/76429

[20] Owasp, "OWASP risk rating methodology," 2013. [Online]. Available: https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology