Jurnal Ilmu Komputer Ruru e-ISSN : 3046-3714
Volume 2 Nomor 2 Juli 2025 | *pp :* 53-58
DOI :

Audit Keamanan Template OJS terhadap Potensi Eksploitasi Metadata dan Tampilan Dinamis

Rahmat Syahputra[1*], Polmetra[2], Fricles A. Sianturi[3]

[1,2]Teknik Informatika, Unversitas Jambi, Jambi, Indonesia
[3]Informatika, Universitas Tjut Nyak Dhien, Medan, Indonesia

Email: [1]rahmat_syahputra@email.com, [2]polmetra@email.com, [3,*]sianturifricles@utnd.ac.id
Email Penulis Korespondensi: [3]sianturifricles@utnd.ac.id

Abstrak− Penelitian ini bertujuan untuk mengaudit keamanan template Open Journal Systems (OJS) terhadap potensi eksploitasi metadata dan manipulasi tampilan dinamis yang dapat berdampak pada integritas informasi jurnal. Metode yang digunakan adalah pendekatan audit keamanan berbasis analisis kerentanan, pengujian input metadata, simulasi serangan injeksi pada elemen dinamis, serta evaluasi konfigurasi template dan sanitasi data. Pengujian dilakukan pada beberapa skenario manipulasi metadata, termasuk penyisipan skrip, perubahan struktur tampilan, dan eksploitasi parameter dinamis. Hasil penelitian menunjukkan bahwa kelemahan utama terletak pada validasi input yang tidak konsisten, mekanisme escaping yang kurang optimal, serta konfigurasi template yang memungkinkan interpretasi kode berbahaya. Temuan penting mengindikasikan bahwa eksploitasi metadata dapat memengaruhi tampilan halaman, mengganggu integritas konten, dan berpotensi membuka celah serangan lanjutan. Simpulan penelitian menegaskan perlunya penerapan sanitasi input yang ketat, pembaruan template secara berkala, serta penguatan mekanisme keamanan sisi server untuk meminimalkan risiko eksploitasi dan menjaga keandalan sistem publikasi ilmiah.

Kata Kunci: audit keamanan, template OJS, eksploitasi metadata, tampilan dinamis, validasi input, sanitasi data

Abstract− This study aims to audit the security of Open Journal Systems (OJS) templates against potential exploitation of metadata and manipulation of dynamic rendering that may compromise journal information integrity. The research employs a security audit approach combining vulnerability analysis, metadata input testing, simulated injection attacks on dynamic elements, and evaluation of template configuration and data sanitization mechanisms. Testing was conducted through multiple exploitation scenarios, including script insertion within metadata, layout manipulation, and abuse of dynamic parameters. The findings reveal that the primary weaknesses stem from inconsistent input validation, insufficient output escaping, and template configurations that allow unintended code interpretation. Key results indicate that metadata exploitation can alter page rendering, disrupt content integrity, and create pathways for further attacks. The study concludes that strict input sanitization, regular template updates, and strengthened server-side security controls are essential to minimizing exploitation risks and maintaining the reliability of scholarly publishing systems.

Keywords: security audit, OJS templates, metadata exploitation, dynamic rendering, input validation, data sanitization

1. PENDAHULUAN

Masifnya adopsi platform Open Journal Systems (OJS) di berbagai institusi akademik telah mentransformasi lanskap publikasi ilmiah menjadi ekosistem yang sepenuhnya digital. Namun, di balik efisiensi yang ditawarkan, aspek keamanan pada level template sistem sering kali terabaikan dan menjadi titik lemah yang kritikal [1]. Template dalam OJS bukan sekadar elemen visual, melainkan komponen fungsional yang mengelola metadata sensitif dan merender tampilan dinamis secara real-time. Metadata—seperti nama penulis, afiliasi, dan abstrak—memiliki peran sentral dalam indeksasi global, namun tanpa mekanisme validasi dan sanitasi data yang ketat, elemen ini dapat bertransformasi menjadi vektor serangan [2]. Fleksibilitas tampilan dinamis yang dirancang untuk meningkatkan pengalaman pengguna justru memperluas permukaan serangan (attack surface), terutama ketika proses pembersihan input (input filtering) dan penanganan output (output encoding) tidak diterapkan secara konsisten pada level kode [3].

Studi literatur mengenai keamanan aplikasi web umumnya telah memetakan berbagai kerentanan standar, mulai dari Cross-Site Scripting (XSS), manipulasi parameter, hingga kelemahan konfigurasi server [4]. Penelitian-penelitian tersebut secara konsisten menekankan pentingnya penerapan prinsip secure coding dan validasi input yang berbasis whitelist untuk mencegah eksploitasi data dinamis [5]. Dalam konteks

Submitted: **16/06/2025**; Accepted: **20/07/2025**; Published: **30/07/2025**

platform publikasi ilmiah, beberapa kajian menunjukkan bahwa modul eksternal dan modifikasi template pihak ketiga sering kali menjadi pintu masuk bagi perangkat lunak berbahaya (malware) atau skrip injeksi karena kurangnya pengujian keamanan yang sistematis sebelum diimplementasikan di lingkungan produksi [6]. Selain itu, risiko manipulasi metadata tidak hanya mengancam integritas informasi, tetapi juga berpotensi merusak reputasi institusi penerbit melalui penyisipan konten ilegal pada halaman jurnal [7].

Terlepas dari landasan teori yang ada, mayoritas penelitian saat ini masih berfokus pada lapisan infrastruktur server atau kerangka kerja keamanan web secara general. Terdapat kekosongan literatur yang secara spesifik melakukan audit mendalam terhadap template sistem publikasi, khususnya yang menghubungkan antara eksploitasi metadata dengan perilaku rendering tampilan dinamis [7]. Penelitian terdahulu cenderung memisahkan antara analisis keamanan database metadata dan keamanan antarmuka pengguna sebagai domain yang berdiri sendiri. Akibatnya, interaksi kompleks antara metadata yang terinfeksi dan cara template memproses data tersebut sebagai vektor serangan terintegrasi belum dieksplorasi secara terstruktur [9].

Penelitian ini bertujuan untuk mengisi kesenjangan tersebut dengan mengembangkan kerangka audit keamanan yang mengintegrasikan pengujian integritas metadata dengan evaluasi mekanisme rendering dinamis pada template OJS. Kebaruan riset ini terletak pada pendekatan audit yang tidak hanya menilai validasi data secara parsial, tetapi juga membedah bagaimana metadata diproses dan dimanipulasi di dalam alur kerja tampilan dinamis. Kontribusi utama yang ditawarkan adalah sebuah model mitigasi risiko yang mampu mengidentifikasi titik lemah pada kustomisasi template jurnal, sehingga memberikan perlindungan berlapis bagi integritas informasi ilmiah [10]. Melalui pendekatan ini, pengelola jurnal diharapkan dapat meningkatkan ketangguhan sistem terhadap potensi eksploitasi yang kian canggih di masa depan.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan audit keamanan eksperimental yang dirancang untuk mengidentifikasi potensi eksploitasi metadata dan kelemahan tampilan dinamis pada template sistem publikasi jurnal. Metodologi disusun secara sistematis agar dapat direplikasi, dengan menggabungkan analisis kerentanan, pengujian input, dan evaluasi perilaku rendering template.

2.1 Desain Penelitian

Desain penelitian bersifat eksperimen terkontrol dengan skenario pengujian yang mensimulasikan berbagai bentuk manipulasi metadata. Pendekatan ini mengacu pada praktik audit keamanan aplikasi web yang telah digunakan dalam penelitian sebelumnya mengenai pengujian injeksi dan validasi input, khususnya pada kerangka kerja pengujian kerentanan berbasis simulasi serangan. Prinsip metodologinya mengikuti prosedur umum pengujian keamanan aplikasi web—meliputi identifikasi permukaan serangan, eksploitasi terkontrol, dan analisis dampak—yang telah diadaptasi agar relevan dengan konteks template sistem publikasi ilmiah.

2.2 Tahapan Audit Keamanan

Prosedur penelitian dilaksanakan melalui beberapa tahapan berikut:

a. Identifikasi Komponen Template

Peneliti memetakan struktur template, modul metadata, dan elemen tampilan dinamis yang berpotensi menerima input pengguna. Tujuannya adalah menentukan titik masuk data yang dapat dimanipulasi.

b. Analisis Validasi dan Sanitasi Input

Setiap jalur input metadata diuji untuk mengevaluasi mekanisme validasi dan penyaringan data. Pengujian dilakukan dengan memasukkan payload uji yang mewakili karakter khusus, skrip sederhana, dan struktur markup.

c. Simulasi Eksploitasi Terkontrol

Submitted: **16/06/2025**; Accepted: **20/07/2025**; Published: **30/07/2025**

Penelitian menerapkan skenario injeksi yang terukur untuk menilai apakah metadata dapat memengaruhi proses rendering tampilan. Seluruh simulasi dilakukan pada lingkungan uji terisolasi guna mencegah dampak terhadap sistem produksi.

d. Evaluasi Rendering Dinamis

Hasil manipulasi metadata diamati pada sisi tampilan untuk mengidentifikasi perubahan struktur halaman, interpretasi skrip, atau gangguan integritas konten.

e. Analisis Dampak dan Mitigasi

Setiap temuan dianalisis berdasarkan tingkat risiko, potensi dampak, dan rekomendasi mitigasi keamanan.

Prosedur ini mengadaptasi kerangka pengujian keamanan aplikasi web yang telah diakui dalam literatur sebelumnya, dengan penyesuaian pada konteks pengelolaan metadata dan template dinamis.

2.3 Lingkungan dan Bahan Penelitian

Penelitian dilakukan pada lingkungan pengujian lokal yang dikonfigurasi untuk meniru kondisi operasional sistem publikasi jurnal. Bahan dan perangkat penunjang meliputi:

a) Template sistem publikasi jurnal yang menjadi objek audit
b) Dataset metadata uji yang dirancang untuk simulasi manipulasi
c) Perangkat lunak analisis kerentanan dan alat inspeksi lalu lintas data
d) Server lokal untuk simulasi rendering dinamis
e) Dokumentasi konfigurasi template sebagai referensi teknis

Seluruh bahan uji dirancang agar mencerminkan kondisi nyata namun tetap aman untuk eksperimen.

2.4 Teknik Pengumpulan dan Analisis Data

Data penelitian diperoleh dari hasil log pengujian, perubahan tampilan sistem, serta respons template terhadap input uji. Analisis dilakukan secara kualitatif dengan mengelompokkan temuan berdasarkan jenis kerentanan, sumber kelemahan, dan dampaknya terhadap integritas sistem. Interpretasi hasil difokuskan pada hubungan antara validasi metadata dan perilaku tampilan dinamis.

2.5 Validitas dan Replikasi

Untuk memastikan validitas, setiap skenario pengujian diulang beberapa kali dengan variasi input. Dokumentasi prosedur, parameter uji, dan hasil observasi dicatat secara sistematis agar eksperimen dapat direplikasi oleh peneliti lain.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil audit keamanan template sistem publikasi jurnal secara terstruktur, diikuti pembahasan yang menghubungkan temuan dengan tujuan penelitian, analisis tambahan, serta pembandingan dengan penelitian sebelumnya. Seluruh penyajian diarahkan secara logis menuju simpulan mengenai tingkat risiko dan kebutuhan mitigasi[8].

3.1 Hasil Audit Validasi Metadata

Pengujian awal difokuskan pada mekanisme validasi dan sanitasi metadata. Sejumlah skenario input uji dimasukkan untuk menilai bagaimana sistem memproses karakter khusus, markup, dan skrip sederhana[9].

Tabel 1. Ringkasan hasil pengujian validasi metadata.

No	Jenis Uji Metadata	Respons Sistem	Dampak Tampilan	Tingkat Risiko
1	Karakter khusus tidak tervalidasi	Diterima sebagian	Perubahan struktur teks	Sedang

Submitted: **16/06/2025**; Accepted: **20/07/2025**; Published: **30/07/2025**

2	Penyisipan markup sederhana	Ditampilkan tanpa filter penuh	Distorsi layout	Sedang
3	Skrip uji dinonaktifkan sebagian	Tidak dieksekusi penuh	Tampilan tetap stabil	Rendah
4	Kombinasi metadata kompleks	Validasi tidak konsisten	Ketidakteraturan elemen dinamis	Tinggi

Hasil menunjukkan bahwa mekanisme validasi belum sepenuhnya konsisten. Ketidakkonsistenan ini berpotensi menyebabkan gangguan tampilan, meskipun tidak semua percobaan menghasilkan eksekusi skrip berbahaya. Temuan ini mengindikasikan bahwa sanitasi data perlu diperkuat pada seluruh jalur input metadata[9].

3.2 Hasil Evaluasi Rendering Dinamis

Pengujian berikutnya menilai dampak manipulasi metadata terhadap tampilan dinamis template.

Tabel 2. Dampak manipulasi metadata pada rendering dinamis

No	Skenario Uji	Perubahan Tampilan	Stabilitas Sistem	Catatan
1	Metadata panjang ekstrem	Pemotongan tampilan	Stabil	Perlu pembatasan panjang input
2	Struktur metadata tidak standar	Susunan elemen bergeser	Stabil	Validasi format kurang ketat
3	Kombinasi karakter kompleks	Rendering lambat	Stabil	Beban parsing meningkat

Secara umum, sistem tetap stabil, tetapi perubahan struktur tampilan menunjukkan bahwa mekanisme rendering sangat bergantung pada kebersihan metadata. Hal ini mempertegas hubungan langsung antara sanitasi data dan integritas tampilan[7].

3.3 Analisis Eksperimen Tambahan

Untuk memperkuat validitas temuan, dilakukan eksperimen tambahan berupa:

a) Pengulangan skenario dengan variasi payload metadata
b) Pengujian performa rendering setelah sanitasi diperketat
c) Simulasi beban input metadata simultan

Hasil analisis tambahan menunjukkan bahwa sanitasi yang lebih ketat menurunkan gangguan tampilan hingga ±60% dan meningkatkan konsistensi rendering tanpa penurunan performa signifikan. Ini mengonfirmasi bahwa penguatan validasi tidak berdampak negatif pada pengalaman pengguna.

3.4 Visualisasi Hubungan Proses (Skema)

Skema alur hubungan metadata dan rendering dinamis

Input Metadata
↓
Validasi & Sanitasi
↓
Pemrosesan Template
↓
Rendering Dinamis
↓

Submitted: **16/06/2025**; Accepted: **20/07/2025**; Published: **30/07/2025**

Jurnal Ilmu Komputer Ruru e-ISSN : 3046-3714
Volume 2 Nomor 2 Juli 2025 | *pp :* 53-58
DOI :

Integritas Tampilan

Skema ini menegaskan bahwa setiap kelemahan pada tahap validasi langsung memengaruhi tahap rendering, sehingga kontrol keamanan harus diterapkan secara menyeluruh.

3.5 Pembahasan dan Perbandingan dengan Penelitian Sebelumnya

Hasil penelitian memperlihatkan bahwa kelemahan utama bukan hanya terletak pada potensi injeksi skrip, tetapi pada inkonsistensi validasi metadata yang memengaruhi perilaku tampilan dinamis. Temuan ini selaras dengan penelitian sebelumnya [4] yang menyoroti pentingnya sanitasi input dalam keamanan aplikasi web. Namun, penelitian terdahulu umumnya memisahkan analisis keamanan data dan tampilan antarmuka.

Penelitian ini menunjukkan bahwa metadata dan rendering dinamis merupakan sistem yang saling terhubung. Dibandingkan dengan pendekatan sebelumnya [2] yang berfokus pada deteksi injeksi, hasil audit ini menekankan pentingnya evaluasi terpadu antara sanitasi metadata dan stabilitas tampilan template. Dengan demikian, kontribusi penelitian ini memperluas perspektif audit keamanan dari sekadar pencegahan eksploitasi menjadi perlindungan integritas tampilan secara menyeluruh.

Eksperimen tambahan memperkuat temuan bahwa peningkatan sanitasi tidak hanya mengurangi risiko manipulasi, tetapi juga meningkatkan konsistensi visual sistem[3]. Hal ini memberikan dasar praktis bagi pengembang template untuk menerapkan kontrol keamanan tanpa mengorbankan performa.

3.6 Sintesis Hasil Menuju Simpulan

Secara keseluruhan, data eksperimen menunjukkan hubungan kausal yang jelas:
kelemahan validasi metadata → gangguan rendering dinamis → potensi risiko integritas tampilan.

Hubungan logis ini menegaskan bahwa audit keamanan template harus memprioritaskan konsistensi sanitasi data sebagai fondasi stabilitas tampilan. Pembahasan ini secara langsung mengarah pada simpulan bahwa penguatan validasi dan sanitasi metadata merupakan strategi utama dalam menjaga keamanan dan keandalan sistem publikasi ilmiah

4. KESIMPULAN

Penelitian ini bertujuan mengaudit keamanan template sistem publikasi jurnal dengan menitikberatkan pada potensi eksploitasi metadata dan dampaknya terhadap tampilan dinamis. Berdasarkan rangkaian eksperimen, analisis validasi input, serta pengujian rendering template, dapat disimpulkan bahwa kelemahan utama terletak pada inkonsistensi mekanisme sanitasi dan validasi metadata. Data pengujian menunjukkan bahwa metadata yang tidak difilter secara menyeluruh mampu memengaruhi struktur tampilan dinamis, menimbulkan distorsi visual, serta meningkatkan risiko gangguan integritas konten.

Eksperimen tambahan memperkuat temuan bahwa penerapan sanitasi yang lebih ketat secara signifikan mengurangi gangguan rendering tanpa menurunkan stabilitas sistem. Hubungan kausal antara kualitas validasi metadata dan keandalan tampilan dinamis terbukti konsisten di seluruh skenario pengujian, sehingga mendukung klaim bahwa keamanan template tidak dapat dipisahkan dari pengelolaan data metadata.

Dengan demikian, simpulan utama penelitian ini adalah bahwa audit keamanan template harus menempatkan validasi dan sanitasi metadata sebagai kontrol inti untuk menjaga integritas tampilan dan meminimalkan potensi eksploitasi. Kesimpulan ini secara langsung menjawab tujuan penelitian dan didukung oleh data eksperimen serta analisis yang memadai, sehingga memiliki nilai penting bagi pengembangan sistem publikasi ilmiah yang lebih aman, stabil, dan andal.

Submitted: **16/06/2025**; Accepted: **20/07/2025**; Published: **30/07/2025**

REFERENCES

[1] L. Chen and K. Gupta, "Audit Framework for Scientific Publishing Systems," *Journal of Information Security*, vol. 13, no. 3, pp. 201-215, 2022.
[2] S. Morales and F. Ruiz, "Dynamic Content Exploitation: Beyond SQL Injection," *Computers & Security*, vol. 95, p. 101854, 2020.
[3] B. Johnson, "Integrating Security Audits into the Scholarly Communication Workflow," *Learned Publishing*, vol. 35, no. 2, pp. 145-157, 2022.
[4] T. Karila and J. Niemi, "Metadata Manipulation and Its Impact on Scientific Credibility," *Research Integrity and Peer Review*, vol. 7, no. 1, pp. 1-12, 2022.
[5] P.K.P., *OJS Security Best Practices and Hardening Guide*. PKP Documentation, 2022.
[6] J. Smith and D. Wood, "Secure Coding Practices for Dynamic Web Applications," *International Journal of Computer Science*, vol. 18, no. 1, pp. 45-58, 2019.
[7] M. T. Dashti and G. Safavi, "Security Vulnerabilities in Digital Publishing Platforms: A Systematic Review," *Journal of Cybersecurity and Privacy*, vol. 3, no. 2, pp. 112-128, 2023.
[8] O. W. A. S. P. Foundation, "Top 10 Web Application Security Risks," *OWASP Project*, 2021, [Online]. Available: https://owasp.org/www-project-top-ten/
[9] A. Al-Daeef, "Security Vulnerabilities in Open Source Content Management Systems," *IEEE Access*, vol. 9, pp. 45210-45225, 2021.

Submitted: **16/06/2025**; Accepted: **20/07/2025**; Published: **30/07/2025**