JURNAL ILMIAH TEKNOLOGI DAN REKAYASA
Vol. 31, No. 01, April 2026, 78 – 94
p-ISSN: 1410-9093, e-ISSN: 2089-8088
https://ejournal.gunadarma.ac.id/index.php/tekno | tekrek@gunadarma.ac.id

Evaluasi keamanan sistem *e-journal* berbasis web menggunakan *framework* NIST SP 800-115 dan analisis berdasarkan OWASP Top 10

[1]Ibnu Ajis, [2]Tri Rochmadi, [3]Yanuar Wicaksono, [4]Dadang Heksaputra

[1,2,3,4] Sistem Informasi, Fakultas Sains, Rekayasa dan Teknologi, Universitas Alma Ata
[1,2,3,4] Jl Brawijaya No 99 Jadan, Tamantirto, Bantul, Daerah Istimewa Yogyakarta 55183
[1]223100292@almaata.ac.id, [2]trirochmadi@almaata.ac.id, [3]yanuar@almaata.ac.id, [4]dadang@almaata.ac.id

Abstract

The use of web-based information systems in higher education institutions, particularly e-journal platforms, continues to rise. However, this is often not accompanied by adequate security measures, thereby threatening *Confidentiality*, *Integrity*, and *Availability* (CIA). This study aims to evaluate the security of a proceedings system based on Open Journal Systems (OJS) at the Adisutjipto Institute of Aeronautical Technology (ITDA) in Yogyakarta. The method used was penetration testing based on the NIST SP 800-115 standard via a black-box approach, which included the planning, discovery, attack, and reporting phases. Initial identification using the OWASP ZAP automated scanner identified 11 vulnerability indicators with estimated severity levels ranging from low to medium. However, after conducting the exploitation validation (proof-of-concept) phase and assessment using the OWASP Risk Rating method, it was found that some vulnerabilities posed a higher risk impact than the initial scan results indicated. The final research results confirmed the presence of 3 high-risk findings and 1 medium-risk finding. These high-risk vulnerabilities include Host Header Injection, which enables Reflected XSS attacks; a Security Misconfiguration on the /server-status endpoint; and the absence of a rate-limiting mechanism in the authentication feature. This study concludes that the system has significant security vulnerabilities that require immediate mitigation. Key recommendations include updating the OJS platform version, implementing server configuration hardening, and enhancing *login* security mechanisms to continuously protect the *Integrity* and *Confidentiality* of the institution's academic data.

Keywords: cybersecurity, NIST SP 800-115, OWASP TOP 10, penetration testing, proceeding

Abstrak

Pemanfaatan sistem informasi berbasis web di perguruan tinggi, khususnya platform *e-journal*, terus meningkat. Namun, hal ini sering tidak diimbangi penguatan keamanan yang memadai sehingga mengancam aspek *Confidentiality*, *Integrity*, dan *Availability* (CIA). Penelitian ini bertujuan mengevaluasi keamanan sistem *proceeding* berbasis *Open Journal Systems* (OJS) di Institut Teknologi Dirgantara Adisutjipto (ITDA) Yogyakarta. Metode yang digunakan adalah *penetration testing* dengan standar NIST SP 800-115 melalui pendekatan *black-box*, yang mencakup tahapan *planning*, *discovery*, *attack*, dan *reporting*. Identifikasi awal menggunakan pemindai otomatis OWASP ZAP menemukan 11 indikasi kerentanan dengan estimasi tingkat keparahan rendah hingga menengah. Namun, setelah dilakukan tahap validasi eksploitasi (*proof-of-concept*) dan penghitungan menggunakan metode OWASP *Risk Rating*, ditemukan bahwa beberapa celah memiliki dampak risiko yang lebih tinggi dari hasil pemindaian awal. Hasil akhir penelitian mengonfirmasi adanya 3 temuan berisiko tinggi (*High*) dan 1 temuan berisiko menengah (*Medium*). Kerentanan berisiko tinggi tersebut meliputi *Host Header Injection* yang memungkinkan serangan *Reflected* XSS, *Security Misconfiguration* pada *endpoint /server-status*, serta ketiadaan mekanisme *rate limiting* pada fitur autentikasi. Hal ini menyimpulkan bahwa sistem memiliki celah keamanan signifikan yang memerlukan mitigasi segera. Rekomendasi

* Corresponding Author

utama meliputi pembaruan versi platform OJS, penerapan *hardening* pada konfigurasi server, serta peningkatan mekanisme keamanan *login* untuk melindungi integritas dan kerahasiaan data akademik institusi secara berkelanjutan.

Kata Kunci: *cybersecurity*, NIST SP 800-115, OWASP TOP 10, *penetration testing, proceeding*

1. Pendahuluan

Transformasi digital dalam sektor pendidikan tinggi telah mendorong peningkatan signifikan penggunaan sistem informasi berbasis web untuk mendukung aktivitas akademik, termasuk pengelolaan publikasi ilmiah. Laporan *International Telecommunication Union* (ITU) tahun 2024 menunjukkan bahwa jumlah pengguna internet global telah mencapai 5.5 miliar jiwa atau sekitar 68% populasi dunia [1]. Pertumbuhan tersebut mendorong peningkatan adopsi platform digital di perguruan tinggi, khususnya sistem *e-journal* dan *proceeding* berbasis web [2], [3]. Digitalisasi publikasi ilmiah mempercepat penyebaran ilmu dan meningkatkan reputasi global institusi. Namun demikian, ekspansi infrastruktur digital tersebut menghadirkan konsekuensi terhadap meningkatnya risiko keamanan siber yang dapat mengancam prinsip *Confidentiality*, *Integrity*, dan *Availability* (CIA) [4].

Data BSSN tahun 2024 menunjukkan adanya 330 juta anomali trafik, dan secara spesifik, sektor akademik di Indonesia sering menjadi target serangan *web defacement* dan *ransomware* karena banyaknya sistem *legacy* yang tidak terawat. Insiden keamanan pada platform *e-journal* berbasis OJS secara global sering kali mengeksploitasi celah *unrestricted file upload* dan *remote code execution* dan *Cross-Site Scripting* (XSS), yang berakibat pada lumpuhnya akses terhadap ribuan naskah ilmiah [5], [6]. Kerentanan seperti SQL *Injection*, XSS, *Broken Authentication*, serta *Security Misconfiguration* terus mendominasi pola serangan aplikasi web secara global [6]–[8]. Dalam kerangka regulatif, diberlakukannya Undang-Undang Perlindungan Data Pribadi (UU PDP) semakin memperkuat urgensi pengelolaan keamanan sistem informasi akademik [9]. Dengan demikian, keamanan *e-journal* tidak lagi bersifat opsional, melainkan menjadi kebutuhan strategis institusi.

Platform *Open Journal Systems* (OJS) sebagai sistem manajemen publikasi ilmiah yang banyak digunakan di Indonesia turut menghadapi tantangan keamanan. Literatur menunjukkan bahwa beberapa versi OJS memiliki kerentanan terdokumentasi, termasuk kelemahan validasi *input* yang berpotensi memicu XSS maupun eksploitasi berbasis manipulasi *header* HTTP [10], [11]. Risiko semakin meningkat apabila sistem tidak diperbarui secara berkala atau menggunakan pustaka pihak ketiga (*third-party libraries*) yang usang, sebagaimana dikategorikan dalam OWASP Top 10 2025 sebagai *Software Supply Chain Failures* dan *Security Misconfiguration* [12]. Penelitian sebelumnya menegaskan bahwa lemahnya praktik *hardening*, penggunaan konfigurasi *default*, serta rendahnya kesadaran pengelola jurnal menjadi faktor dominan munculnya kerentanan pada sistem OJS [13]–[15].

Tinjauan terhadap penelitian terdahulu menunjukkan adanya kesenjangan penelitian yang nyata. Berbeda dengan penelitian oleh Saputra [7] yang hanya melakukan verifikasi kerentanan XSS, penelitian ini mengisi celah dengan melakukan validasi eksploitasi (*proof-of-concept*) terkontrol pada OJS versi *legacy* (2.4.8.1). Pemilihan versi ini bersifat representatif karena masih banyak digunakan oleh institusi di Indonesia meski dukungan keamanannya telah berakhir. Hingga saat ini, belum terdapat evaluasi keamanan spesifik pada sistem *proceeding* Seminar Nasional Teknologi Informasi dan Kedirgantaraan Institut Teknologi Dirgantara Adisutjipto (SENATIK ITDA) Yogyakarta yang menggunakan versi tersebut.

Penelitian ini mengoperasionalkan kerangka teoretis secara terintegrasi. Prinsip CIA Triad digunakan sebagai parameter utama untuk mengukur dampak kerusakan data. Metodologi *National Institute of Standards and Technology Special Publication* (NIST SP)

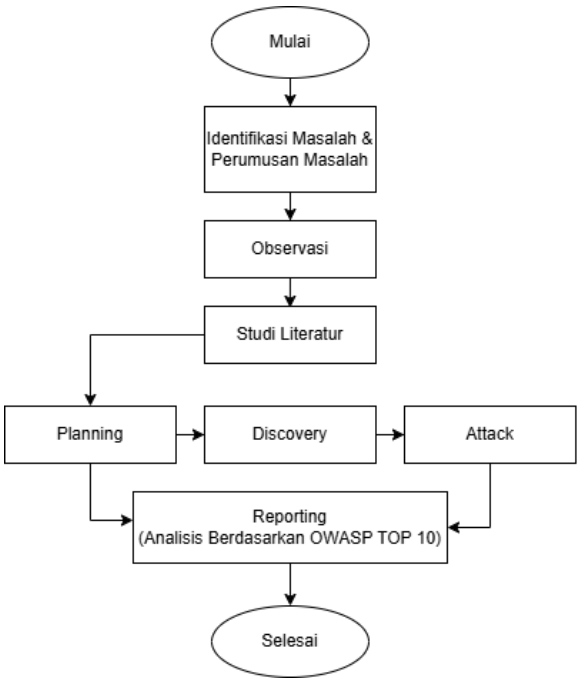
800-115 diterapkan sebagai panduan langkah kerja teknis (*planning*, *discovery*, *attack*, dan *reporting*). Standar OWASP Top 10 2025 digunakan sebagai instrumen klasifikasi tingkat risiko serta penentuan prioritas mitigasi.

Permasalahan penelitian ini adalah belum dilakukannya evaluasi keamanan menyeluruh terhadap sistem *proceeding* SENATIK ITDA berbasis OJS 2.4.8.1, sehingga potensi bahaya belum teridentifikasi secara sistematis. Sebagai bentuk kepatuhan terhadap etika keamanan siber (*ethical hacking*), penelitian ini telah mendapatkan otorisasi resmi melalui surat izin pengujian dari pengelola sistem terkait. Pengujian dilakukan dengan batasan ketat tanpa melakukan tindakan destruktif terhadap basis data maupun mengganggu ketersediaan layanan sistem bagi pengguna umum.

Tujuan utama penelitian ini adalah mengidentifikasi kerentanan aktual dan mengklasifikasikan risiko berdasarkan ancaman terkini. Kebaruan penelitian ini terletak pada pengembangan model evaluasi risiko yang memadukan validasi eksploitasi nyata dengan kerangka NIST pada sistem publikasi ilmiah *legacy*. Penelitian ini diharapkan memberikan kontribusi berupa model referensi *hardening* sistem bagi institusi dengan sumber daya terbatas yang masih bergantung pada sistem lama, untuk menjaga integritas ekosistem publikasi ilmiah nasional. Keberhasilan evaluasi diukur secara kuantitatif melalui akurasi penilaian *risk score* (skala 1–9) serta perumusan rekomendasi mitigasi teknis yang aplikatif [16].

2. Metode Penelitian

Penelitian ini menggunakan pendekatan studi kasus dengan metode analisis keamanan berbasis *penetration testing*. Desain ini dipilih karena permasalahan yang dikaji bersifat kontekstual dan spesifik pada sistem *proceeding* SENATIK berbasis OJS versi 2.4.8.1 di ITDA Yogyakarta. Pendekatan ini memungkinkan evaluasi mendalam terhadap konfigurasi, versi perangkat lunak, serta kondisi operasional aktual sistem yang tidak dapat digeneralisasi secara langsung pada institusi lain. Tahapan penelitian disusun secara sistematis seperti yang diilustrasikan pada Gambar 1, yang menggambarkan alur kerja mulai dari identifikasi masalah, pelaksanaan pengujian teknis, hingga penyusunan rekomendasi mitigasi.



Gambar 1. Alur penelitian

Gambar 1 menunjukkan alur kerja penelitian yang dimulai dari tahap pra-pengujian hingga penyusunan hasil akhir. Alur penelitian diawali dengan identifikasi permasalahan keamanan pada sistem *proceeding* SENATIK ITDA, dilanjutkan studi literatur untuk menentukan dasar teoritis dan metode pengujian. Pengujian dilakukan mengikuti tahapan NIST SP 800-115 (*planning*, *discovery*, *attack*, dan *reporting*), kemudian temuan diklasifikasikan berdasarkan OWASP Top 10 dan dinilai menggunakan metode OWASP *Risk Rating*. Tahap akhir berupa penyusunan laporan evaluasi yang memuat temuan, tingkat risiko, dan rekomendasi mitigasi untuk peningkatan keamanan sistem.

2.1 Desain Penelitian

Penelitian ini menggunakan desain studi kasus dengan pendekatan deskriptif kualitatif. Objek penelitian adalah sistem *proceeding* SENATIK ITDA Yogyakarta yang berbasis OJS. Pendekatan kualitatif digunakan untuk mendeskripsikan dan menganalisis temuan kerentanan secara sistematis berdasarkan hasil pengujian keamanan. Evaluasi dilakukan menggunakan metode *penetration testing* dengan pendekatan *black-box security testing*, di mana penguji bertindak sebagai penyerang eksternal tanpa memiliki akses internal terhadap sistem [17], [18]. Pendekatan ini dipilih untuk merepresentasikan kondisi serangan nyata dari luar sistem [19].

2.2 Metode Pengumpulan Data

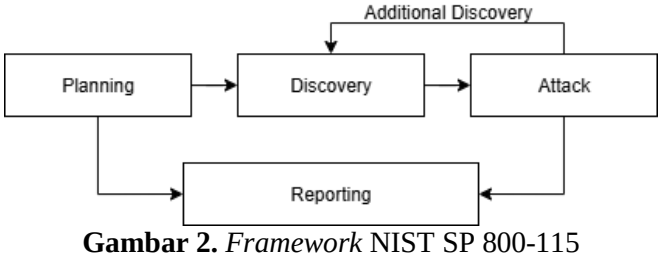
Pengumpulan data dilakukan melalui studi literatur serta observasi dan pengujian langsung terhadap sistem *proceeding* SENATIK ITDA. Studi literatur bertujuan memperoleh landasan teoritis mengenai keamanan informasi, *penetration testing*, NIST SP 800-115, dan OWASP Top 10 dari sumber ilmiah bereputasi. Sementara itu, observasi dan pengujian dilakukan dengan pendekatan *black-box* untuk mengidentifikasi teknologi, layanan, serta kerentanan sistem melalui *vulnerability scanning* dan validasi terbatas (*proof of concept*) dari perspektif penyerang eksternal tanpa akses internal.

2.3 Analisis Kebutuhan

Analisis kebutuhan penelitian difokuskan pada perangkat lunak dan lingkungan pengujian yang mendukung pelaksanaan *penetration testing*. Lingkungan pengujian menggunakan sistem operasi Windows 11 dan Kali Linux yang dijalankan melalui *VirtualBox*. *Tools* yang digunakan meliputi *Wappalyzer* untuk identifikasi teknologi web, *Nmap* untuk pemetaan jaringan dan *port scanning*, SSLScan untuk evaluasi konfigurasi SSL/TLS, OWASP ZAP untuk *vulnerability scanning*, *Burp Suite* untuk analisis dan validasi eksploitasi. Kombinasi *tools* tersebut digunakan untuk memastikan proses identifikasi kerentanan dilakukan secara komprehensif [20].

2.4 Metode Pengujian

Metode pengujian mengacu pada *framework* NIST SP 800-115 yang terdiri dari empat tahapan utama. NIST SP 800-115 merupakan pedoman teknis yang diterbitkan oleh NIST untuk membantu organisasi dalam merencanakan, melaksanakan, dan melaporkan kegiatan pengujian keamanan informasi [21], [22]. Tahapan pengujian tersebut dapat dilihat pada Gambar 2.



Gambar 2. *Framework* NIST SP 800-115

Berdasarkan Gambar 2, pengujian dilakukan secara sistematis mengikuti standar NIST SP 800-115 yang terbagi dalam empat fase teknis yang saling berkesinambungan. Tahap perencanaan (*planning*) diawali dengan menentukan ruang lingkup pengujian pada domain *senatik.itda.ac.id*, menyusun *rules of engagement*, memastikan aspek legalitas dari pengelola sistem, serta menyiapkan perangkat seperti *Nmap* dan *Burp Suite* pada lingkungan Kali Linux. Selanjutnya, pada tahap penemuan (*discovery*), dilakukan aktivitas *information gathering* melalui teknik *active scanning* dan *service enumeration* untuk memetakan arsitektur teknologi serta mengidentifikasi celah keamanan yang ada. Setelah kerentanan ditemukan, tahap penyerangan (*attack*) dilaksanakan dengan melakukan validasi risiko melalui metode eksploitasi terbatas (*proof-of-concept*) pada celah yang ditemukan, seperti kerentanan *input* dan manipulasi *header*, dengan batasan ketat agar tidak merusak basis data atau mengganggu ketersediaan layanan sistem. Rangkaian ini diakhiri dengan tahap pelaporan (*reporting*) yang mencakup dokumentasi seluruh temuan, klasifikasi jenis ancaman berdasarkan standar OWASP Top 10 2025, serta perhitungan bobot risiko menggunakan metode OWASP *Risk Rating* untuk menghasilkan rekomendasi mitigasi yang solutif. Kerangka kerja ini diterapkan untuk menjamin bahwa evaluasi keamanan aplikasi web dilakukan secara terstruktur dan menyeluruh [12].

Untuk memastikan proses pengujian keamanan berjalan secara terstruktur dan dapat direplikasi, penelitian ini menggunakan lingkungan pengujian yang terstandardisasi. Lingkungan tersebut dibangun menggunakan kombinasi beberapa sistem operasi, *tools* pemindaian, dan perangkat lunak pendukung penetration testing yang disesuaikan dengan kebutuhan penelitian. Rincian dari setiap komponen yang digunakan disajikan pada Tabel 1.

Tabel 1. *Environment*

No.	Nama Perangkat Lunak	*Application Software*
1	Windows 11 Pro 64-bit	- VirtualBox Version 7.2.2 - OWASP ZAP 2.16.1 - Wappalyzer - Burp Suite - Google Colab
2	Kali Linux	- Nmap - SSLScan

Pada tahap pengujian otomatis, OWASP ZAP dikonfigurasi dengan *medium threshold* untuk memindai kerentanan pada aplikasi web. Proses *information gathering* dan *service enumeration* dilakukan menggunakan *Nmap*, sementara analisis konfigurasi SSL/TLS menggunakan SSLScan. Identifikasi arsitektur teknologi pada situs web dilakukan dengan menggunakan *Wappalyzer*.

Sementara itu, pengujian manual dilakukan melalui fitur *intercepting proxy* pada *Burp Suite*, dengan fokus pada manipulasi *request* HTTP terhadap sistem OJS 2.4.8.1 yang menjalankan pustaka jQuery 1.4.4. Seluruh pengujian dilaksanakan dalam ruang lingkup yang telah ditentukan untuk menjaga integritas basis data dan ketersediaan layanan selama proses berlangsung.

2.5 Teknik Analisis Data

Kerentanan yang ditemukan diklasifikasikan berdasarkan OWASP Top 10 2025 untuk memastikan relevansinya dengan tren ancaman aplikasi web terkini. Penilaian tingkat risiko dilakukan menggunakan metode OWASP *Risk Rating* [16] dengan Persamaan (1).

$$Risk = likelihood \times impact \quad (1)$$

Parameter *likelihood* digunakan untuk mengukur seberapa besar kemungkinan atau kemudahan celah tersebut dieksploitasi. Sementara itu, *impact* digunakan untuk menilai

dampak kerusakan pada aspek kerahasiaan (*Confidentiality*), keutuhan (*Integrity*), dan ketersediaan (*Availability*). Setiap aspek penilaian diberi nilai menggunakan skala 1–3. Detail mengenai kategori nilai tersebut disajikan pada Tabel 2.

Tabel 2. Skala penilaian

Skor	Keterangan
1	Rendah
2	Sedang
3	Tinggi

Berdasarkan Tabel 2, semakin tinggi angka skor yang diberikan, maka semakin besar pula tingkat kemungkinan (*likelihood*) maupun dampak (*impact*) yang ditimbulkan. Selanjutnya, hasil perkalian antara nilai *likelihood* dan *impact* akan menghasilkan skor risiko (*risk score*) dengan rentang 1–9. Skor tersebut kemudian dikelompokkan menjadi tiga tingkat risiko (*Low*, *Medium*, dan *High*) untuk menentukan prioritas perbaikan (*mitigasi*). Pengelompokan tingkat risiko ini dirinci pada Tabel 3.

Tabel 3. Penilaian risiko (*risk level*)

Risk Score	***Risk Level***
1 to <3	*Low*
3 to <6	*Medium*
6 to 9	*High*

Berdasarkan Tabel 3, dapat dipahami bahwa celah keamanan yang memiliki skor 6 ke atas dikategorikan sebagai risiko tinggi (*High*) dan harus menjadi prioritas utama untuk segera diperbaiki. Hasil perhitungan ini menjadi dasar bagi organisasi dalam mengambil tindakan mitigasi yang tepat [16].

3. Hasil dan Pembahasan

3.1 Hasil Perencanaan (*Planning*) Pengujian

Tahap perencanaan menghasilkan penetapan ruang lingkup, metode, dan batasan pengujian keamanan pada sistem *proceeding* SENATIK ITDA Yogyakarta. Pengujian menggunakan pendekatan *black-box penetration* testing tanpa akses internal dan dibatasi pada domain publik tanpa eksploitasi destruktif. *Tools* yang digunakan meliputi *Wappalyzer*, *Nmap*, SSLScan, OWASP ZAP, dan *Burp Suite*. Tahap ini memastikan pengujian dilakukan secara terkontrol, etis, dan sesuai kerangka kerja NIST SP 800-115.

3.2 Penemuan *(Discovery)*

Tahap penemuan terdiri dari *information gathering* dan *vulnerability scanning*.

1. *Information Gathering*

Hasil identifikasi teknologi menggunakan *Wappalyzer* menunjukkan bahwa sistem *proceeding* SENATIK ITDA teridentifikasi beroperasi menggunakan platform *legacy* PKP OJS versi 2.4.8.1. Temuan ini secara teoretis menjadi akar masalah dari berbagai kerentanan yang ditemukan, mengingat versi 2.4.x merupakan kategori perangkat lunak usang yang tidak lagi menerima pembaruan keamanan rutin. Penemuan ini selaras dengan argumen Ginanjar [13] bahwa faktor dominan kerentanan pada sistem informasi di Indonesia adalah lemahnya manajemen pembaruan (*patch management*). Hasil pemetaan *port* dan evaluasi enkripsi SSL/TLS pada Tabel 4 menunjukkan bahwa meskipun sistem telah mendukung protokol

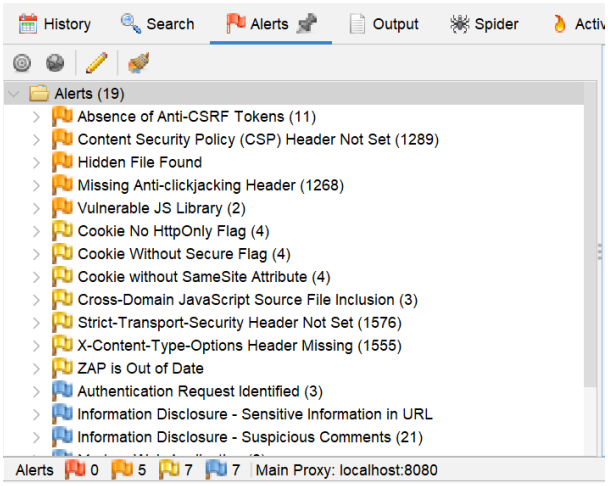
modern, terdapat konfigurasi default yang masih aktif dan memperluas permukaan serangan (*attack surface*).

Tabel 4. Ringkasan konfigurasi layanan dan keamanan jaringan

Kategori	Parameter	Temuan Teknis	Implikasi Keamanan
Layanan Aktif	*Port* 80, 82, 443, 8081	*Apache* HTTP Server	Fokus utama permukaan serangan web.
Enkripsi	Protokol SSL/TLS	TLS 1.0, 1.1, 1.2, 1.3	Aktifnya TLS 1.1 memungkinkan *downgrade attack*.
Sertifikat	Algoritma Tanda Tangan	SHA256withRSAEncryption	Integritas sertifikat saat ini masih stabil.

2. *Vulnerability Scanning*

Pemindaian menggunakan OWASP ZAP pada Gambar 3 mengidentifikasi total 11 temuan kerentanan dengan tingkat keparahan rendah hingga menengah. Temuan tersebut mencakup kategori *Security Misconfiguration, Vulnerable JavaScript Library,* serta indikasi kelemahan pada mekanisme autentikasi.



Gambar 1. Hasil *scanning* OWASP ZAP

Hasil *scanning* menjadi dasar untuk dilakukan validasi lebih lanjut pada tahap penyerangan sehingga tingkat risiko aktual dari setiap kerentanan dapat dipastikan.

3.3 Penyerangan (*Attack*)

Tahap penyerangan dilakukan untuk memvalidasi kerentanan yang teridentifikasi pada tahap sebelumnya melalui eksploitasi terbatas (*proof of concept*).

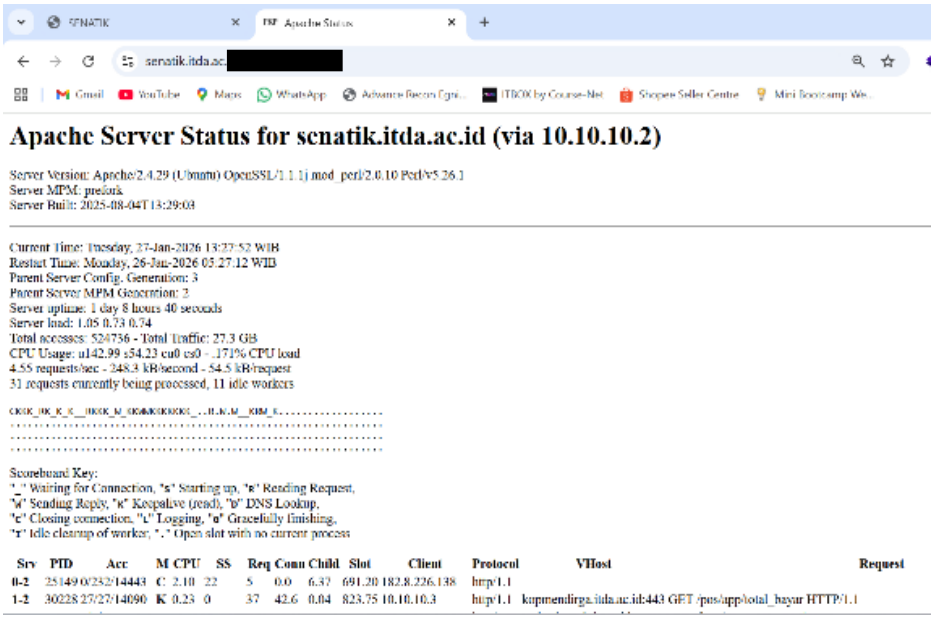
1. A01:2025-*Broken Access Control*

Evaluasi terhadap mekanisme kontrol akses dilakukan melalui teknik manipulasi parameter URL dan identifikasi ID artikel secara sekuensial. Hasil pengujian menunjukkan bahwa sistem secara efektif memvalidasi otorisasi pada antarmuka publik, sehingga tidak ditemukan adanya akses ilegal terhadap dokumen privat atau konten pra-publikasi. Temuan ini mengonfirmasi terjaganya aspek *Confidentiality* pada level akses publik. Namun, efikasi sistem pada level manajerial (*editor, reviewer,* dan *administrator*) menjadi limitasi dalam penelitian ini. Pendekatan *black-box testing* yang digunakan membatasi pengujian hanya pada sudut pandang pengguna eksternal, tanpa akses ke kode sumber aplikasi, sehingga kerentanan yang bersifat internal kemungkinan belum dapat teridentifikasi secara menyeluruh [23]–[25].

Di sisi lain, karena penelitian ini merupakan studi kasus pada satu sistem, temuan yang dihasilkan belum tentu dapat digeneralisasi terhadap sistem informasi di institusi lain yang memiliki arsitektur, lingkungan, maupun kebijakan keamanan yang berbeda.

2. A02:2025-*Security Misconfiguration*

Hasil *vulnerability scanning* menggunakan OWASP ZAP menemukan *endpoint /server-status* pada Gambar 4 yang terbuka tanpa autentikasi dan mengungkap informasi sensitif seperti versi Apache, modul aktif, serta detail permintaan klien, yang mengindikasikan *misconfiguration* pada server. Berdasarkan hasil pengujian, ditemukan temuan kerentanan terkait kategori A02:2025 *Security Misconfiguration* pada *endpoint /server-status.* Temuan ini menunjukkan adanya kesalahan konfigurasi pada Apache Web Server yang menyebabkan informasi teknis sensitif terbuka untuk publik. Dampak utamanya terletak pada aspek *Confidentiality*, di mana penyerang dapat memetakan struktur server, memantau IP klien yang aktif, serta melihat beban sistem secara real-time untuk merencanakan serangan lebih lanjut.

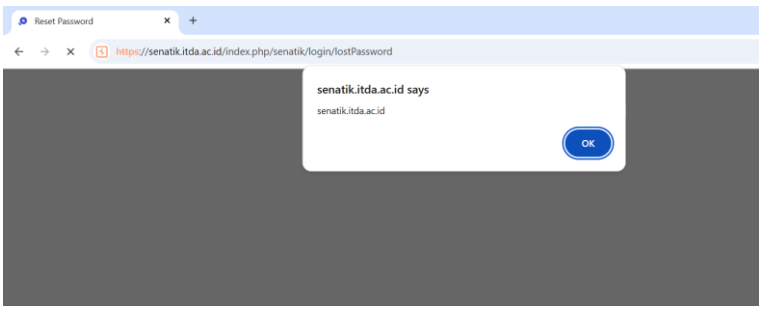


Gambar 2. *Endpoint*/server-status

3. A03:2025-*Software Supply Chain Failures*

A. PKP OJS versi 2.4.8.1

Tahap *discovery* menggunakan *Wappalyzer* mengidentifikasi penggunaan OJS versi 2.4.8.1. Berdasarkan Google *Hacking Database* (GHDB), versi tersebut terdampak kerentanan XSS melalui *Host Header Injection.* Verifikasi menggunakan *Burp Suite* dengan manipulasi *header X-Forwarded-Host* pada halaman *lostPassword* berhasil memicu eksekusi skrip di sisi klien, ditunjukkan oleh munculnya *alert box* seperti yang ditunjukkan pada Gambar 5. Hal ini mengonfirmasi adanya *Reflected* XSS pada sistem yang diuji.



Gambar 3. *Alert* XSS

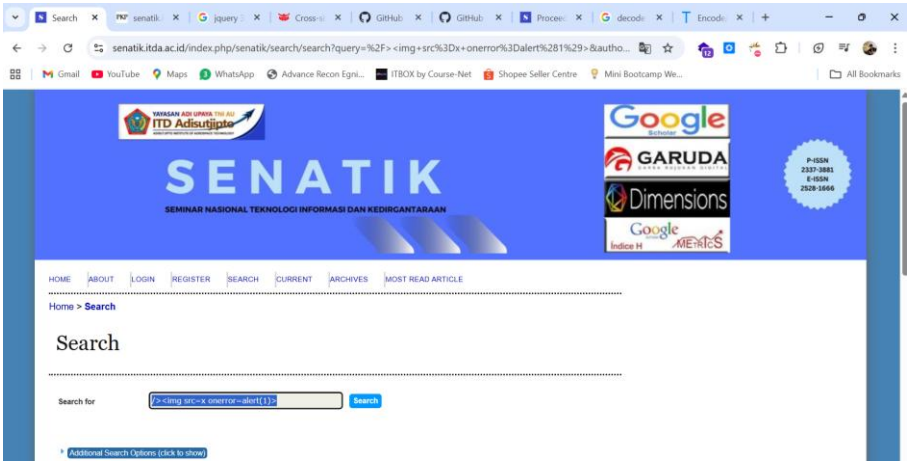
B. JQueryUI 1.8.6

Wappalyzer mendeteksi penggunaan jQuery UI versi 1.8.6 yang memiliki kerentanan XSS terdokumentasi. Validasi dilakukan dengan mengeksekusi *payload* melalui *browser console* pada Gambar 6 berdasarkan referensi *public exploit*. Meskipun tidak muncul *alert dialog*, perubahan tampilan halaman membuktikan keberhasilan eksekusi kode berbahaya secara *client-side*, yang mengindikasikan kerentanan DOM-based XSS.

Gambar 6. Validasi kerentanan DOM-based XSS pada JQuery UI 1.8.6

C. JQuery 1.4.4

Wappalyzer mengidentifikasi penggunaan jQuery versi 1.4.4 yang memiliki kerentanan XSS terdokumentasi. Namun, pengujian menggunakan *payload* eksploitasi pada parameter *input* tidak menunjukkan eksekusi skrip maupun *alert dialog* pada Gambar 7. Oleh karena itu, kerentanan tersebut tidak dapat tervalidasi pada implementasi sistem yang diuji.



Gambar 7. Hasil pengujian

4. A04:2025-*Cryptographic Failures*

Hasil pengujian menggunakan SSLscan pada Gambar 5 menunjukkan bahwa server telah mengimplementasikan SSL/TLS dengan menonaktifkan SSLv2 dan SSLv3, serta mengaktifkan TLSv1.1, TLSv1.2, dan TLSv1.3. Sertifikat menggunakan algoritma *SHA256withRSAEncryption* dengan panjang kunci 2048 bit dan masih valid. Namun, dukungan terhadap TLSv1.1 menunjukkan konfigurasi kriptografi belum optimal karena protokol tersebut telah usang dan berpotensi dimanfaatkan dalam serangan *downgrade attack.*

5. A05:2025-*Injection*

Pengujian *Injection* dilakukan pada formulir *login* dan fitur pencarian dengan memasukkan beberapa *payload* SQL *Injection* dan XSS untuk mengamati respons sistem terhadap *input* yang tidak wajar. Hasil pengujian tidak menunjukkan adanya pesan kesalahan *database, bypass autentikasi,* maupun eksekusi skrip berbahaya. Namun, karena pengujian dilakukan secara terbatas dan tanpa akses ke log server, hasil ini belum dapat memastikan bahwa sistem sepenuhnya bebas dari kerentanan *Injection.*

6. A06:2025-*Insecure Design*

Pengujian *Insecure Design* dilakukan dengan mengirim 20 permintaan berturut-turut ke *endpoint login, pendaftaran,* dan pencarian. Simulasi ini dioperasionalkan menggunakan skrip berbasis Python *requests* untuk menguji efikasi kebijakan kontrol frekuensi atau *rate limiting* yang diterapkan pada aplikasi. Berdasarkan observasi terhadap respons server, ditemukan bahwa sistem SENATIK ITDA memproses seluruh permintaan (100%) dengan status kode HTTP 200 OK tanpa adanya mekanisme proteksi aktif, seperti pemblokiran alamat IP sementara, tantangan CAPTCHA, maupun pembatasan sesi pada *endpoint* target. Fenomena ketiadaan pembatasan frekuensi ini mengonfirmasi adanya celah signifikan pada kategori *Insecure Design*. Secara teknis, kelemahan arsitektur ini memberikan peluang bagi aktor ancaman untuk melakukan serangan *brute force* pada kredensial administrator atau melakukan pengambilan data secara masif (*scraping*) yang dapat menguras sumber daya server. Dalam perspektif prinsip CIA Triad, kerentanan ini memberikan dampak kritis pada aspek ketersediaan (*Availability*) akibat potensi beban berlebih pada infrastruktur (*resource exhaustion*) dan aspek kerahasiaan (*Confidentiality*) melalui upaya pembajakan akun.

7. A07:2025-*Authentication Failures*

Pengujian dilakukan dengan mensimulasikan serangan *brute force* secara terkontrol pada *endpoint* /index.php/journal/login/signIn. Simulasi ini dirancang untuk menguji ketahanan logika bisnis aplikasi terhadap upaya akses ilegal dalam skala besar, yakni dengan mengirimkan 50 percobaan *login* menggunakan kredensial tidak valid. Setiap permintaan dikirimkan dengan interval 0,5 detik secara sekuensial, agar efektivitas mekanisme *rate limiting* dapat dievaluasi secara objektif.

Hasil pengujian menunjukkan adanya celah pada lapisan keamanan sistem. Server memproses seluruh 50 permintaan tanpa terkecuali, semuanya mengembalikan status HTTP 200 OK. Selama pengujian berlangsung, tidak ada satu pun mekanisme proteksi yang aktif, baik berupa pemblokiran IP sementara, munculnya tantangan CAPTCHA, maupun respons HTTP 429 (*Too Many Requests*) yang seharusnya muncul ketika sistem mendeteksi lonjakan trafik yang tidak wajar.

Ketiadaan mekanisme pembatasan *login* ini merupakan kerentanan yang masuk dalam kategori A07:2025 – *Authentication Failures* pada OWASP Top 10. Kondisi ini secara teoritis membuka peluang bagi penyerang untuk melancarkan serangan *dictionary attack* maupun *credential stuffing* secara terus-menerus hingga berhasil memperoleh akses yang sah. Jika ditinjau dari prinsip CIA Triad, kerentanan ini berdampak langsung pada dua aspek utama. Pertama, aspek *Confidentiality*, karena pengambilalihan akun administrator berpotensi mengekspos data riset yang bersifat sensitif. Kedua, aspek *Integrity*, karena akses yang tidak sah dapat membuka jalan bagi manipulasi konten publikasi ilmiah pada sistem.

8. A08:2025-*Software and Data Integrity Failures*

Pengujian difokuskan pada konsistensi metadata artikel serta integritas navigasi dan tautan. Akses berulang pada halaman artikel menunjukkan metadata seperti judul dan nama penulis konsisten, meskipun beberapa informasi tercatat kosong atau N/A. Hal ini

mengindikasikan integritas konten publik relatif terjaga, namun keterbatasan metadata menjadi catatan penting dari sisi transparansi informasi. Pengujian navigasi menunjukkan sebagian besar tautan internal berfungsi dengan baik (status 200 OK) dan tidak ditemukan *broken link* pada konten utama. Beberapa tautan tambahan eksternal teridentifikasi, namun hanya berpotensi menimbulkan gangguan minor. Secara keseluruhan, integritas konten dan navigasi publik OJS tergolong stabil.

9. A09:2025-*Logging & Alerting Failures*

Pengujian *Logging & Alerting Failures* dilakukan dengan mengakses *login* gagal, URL tidak valid, dan *input* anomali pada *form* melalui akses publik. Seluruh permintaan menghasilkan respons 200 OK tanpa pesan kesalahan, pembatasan, atau mekanisme perlindungan tambahan. Hal ini menunjukkan tidak adanya mekanisme *logging* dan *alerting* yang memadai terhadap aktivitas abnormal, sehingga meningkatkan risiko serangan seperti *brute force* dan eksploitasi berulang.

10. A10:2025-*Mishandling of Exceptional Conditions*

Pengujian difokuskan pada bagaimana sistem menangani *input* tidak valid dan URL yang tidak tersedia. Beberapa *input* anomali seperti <, >, *' OR '1'='1*, dan *<script>alert(1)</script>* dikirim melalui *form* pencarian, serta dilakukan akses ke URL artikel yang tidak ada. Hasil menunjukkan sistem menangani kondisi tersebut dengan aman melalui mekanisme *direct redirect*, tanpa menampilkan *stack trace, error database*, maupun informasi internal. Dengan demikian, risiko *Mishandling of Exceptional Conditions* pada akses publik tergolong minimal.

3.4 Pelaporan *(Reporting)*

1. *Security Misconfiguration (/server-status)*

Endpoint/server-status ditemukan dapat diakses publik tanpa autentikasi, sehingga mengekspos informasi teknis sensitif seperti versi Apache, koneksi aktif, IP klien, dan beban kerja server secara *real-time*. Kondisi ini menunjukkan adanya konfigurasi Apache yang belum dinonaktifkan pada lingkungan produksi. Berdasarkan metode penilaian risiko, hasil analisis disajikan pada Tabel 5.

Tabel 5. Penilaian risiko *security missconfiguration*

Parameter	Nilai	Justifikasi
Likelihood	2 *(Medium)*	*Endpoint* dapat diakses publik
Impact	2 *(Medium)*	Informasi sensitif server terekspos

Penilaian risiko terhadap temuan ini menghasilkan skor 4 (tingkat Medium). Nilai *Likelihood* (2) didasarkan pada kemudahan akses endpoint tanpa autentikasi, meski tetap memerlukan tahap pemindaian oleh penyerang. Sementara itu, nilai *Impact* (2) diberikan karena pelanggaran aspek *Confidentiality* data teknis. Eksposur informasi ini dapat dimanfaatkan penyerang sebagai basis intelijen untuk merencanakan serangan yang lebih presisi, seperti eksploitasi celah spesifik pada versi perangkat lunak atau serangan *Denial of Service (DoS)* yang mengancam ketersediaan layanan sistem.

2. *Host Header Injection* (*Reflected* XSS)

Kerentanan ini terjadi karena ketiadaan validasi pada header *X-Forwarded-Host.* Pengujian menggunakan Burp Suite membuktikan bahwa penyisipan payload skrip pada

header tersebut berhasil dieksekusi oleh peramban *(Reflected XSS)*. Berdasarkan metode penilaian risiko, hasil analisis disajikan pada Tabel 6.

Tabel 6. Penilaian risiko *host header injection*

Parameter	Nilai	Justifikasi
Likelihood	3 *(High)*	Mudah dieksploitasi dengan tools umum
Impact	2 *(Medium)*	Eksekusi script *client-side*

Hasil analisis menetapkan skor risiko sebesar 6 (tingkat *High*). Nilai *Likelihood* (3) mencerminkan tingginya probabilitas serangan karena celah ini dapat dieksploitasi dengan alat standar tanpa memerlukan autentikasi. Sementara itu, nilai *Impact* (2) didasarkan pada ancaman terhadap aspek *Confidentiality* (pencurian *cookie* atau sesi) dan *Integrity* (manipulasi konten halaman). Mengingat tingginya kemudahan eksploitasi dan dampak langsung terhadap keamanan pengguna, kerentanan ini ditetapkan sebagai prioritas utama dalam langkah mitigasi.

3. *Lack of Rate Limiting*

Kerentanan ini ditemukan pada fitur *login* dan pemulihan kata sandi (*lost password*), di mana ketiadaan mekanisme pembatasan frekuensi permintaan (*rate limiting*) memungkinkan terjadinya serangan otomatis seperti *brute force*. Berdasarkan metode penilaian risiko, hasil analisis disajikan pada Tabel 7.

Tabel 7. Penilaian risiko *lack of rate limiting*

Parameter	Nilai	Justifikasi
Likelihood	3 *(High)*	Dapat dilakukan dengan *script* otomatis
Impact	2 *(Medium)*	Pengambilalihan akun

Hasil analisis menetapkan skor risiko sebesar 6 (tingkat *High*). Nilai *likelihood* (3) dinilai tinggi karena sistem tidak menerapkan hambatan teknis seperti CAPTCHA atau pemblokiran IP, sehingga memudahkan penggunaan skrip otomatis. Nilai *Impact* (2) didasarkan pada ancaman terhadap aspek *Confidentiality* melalui potensi pengambilalihan akun pengguna, serta aspek *Integrity* melalui penyalahgunaan akun yang telah dikuasai. Selain itu, serangan otomatis berskala besar berisiko membebani sumber daya peladen yang dapat mengganggu aspek *Availability* atau ketersediaan layanan bagi pengguna sah.

4. *Vulnerable* JavaScript *Library*

Sistem ditemukan menggunakan pustaka jQuery UI versi 1.8.6 yang sudah usang dan memiliki berbagai celah XSS yang terdokumentasi secara publik. Kondisi ini mencerminkan kelemahan dalam manajemen pemeliharaan dependensi (*patch management*) aplikasi. Berdasarkan metode penilaian risiko, hasil analisis disajikan pada Tabel 8.

Tabel 8. Penilaian risiko *vulnerable* JavaScript *library*

Parameter	Nilai	Justifikasi
Likelihood	3 *(High)*	*Exploit* publik tersedia
Impact	2 *(Medium)*	Eksekusi *script* sisi klien

Hasil analisis menetapkan skor risiko sebesar 6 (tingkat *High*). Nilai *likelihood* (3) dinilai tinggi karena keberadaan celah keamanan yang sudah diketahui umum (*known*

vulnerabilities) sangat memudahkan penyerang dalam melakukan eksploitasi. Sementara itu, nilai *impact* (2) didasarkan pada ancaman terhadap aspek *Confidentiality* melalui potensi pencurian data pengguna, serta aspek *Integrity* melalui manipulasi struktur DOM atau konten halaman (*web defacement*). Mengingat risiko eksploitasi *client-side* yang dapat mengganggu fungsi normal sistem, pembaruan pustaka ke versi stabil merupakan langkah mitigasi yang krusial.

Setelah dilakukan pengujian teknis dan validasi eksploitasi, seluruh temuan kerentanan dianalisis bobot risikonya. Ringkasan hasil penilaian risiko yang mencakup parameter *likelihood*, *impact*, serta skor risiko akhir disajikan pada Tabel 9.

Tabel 9. Penilaian risiko

No.	Kerentanan	*Likelihood*	*Impact*	*Risk Score*	*Level*
1	*Exposed/server-status*	3	2	6	*High*
2	*Host Header Injection* (XSS)	3	2	6	*High*
3	*Lack of Rate Limiting*	3	3	9	*High*
4	*Vulnerable JS Library*	2	2	4	*Medium*

Berdasarkan data pada Tabel 9, mayoritas kerentanan berada pada tingkat risiko tinggi *(High)*, terutama pada aspek kontrol akses dan konfigurasi server. Skor tertinggi (9) terdapat pada celah *Lack of Rate Limiting*, yang menunjukkan urgensi penanganan pada mekanisme autentikasi sistem.

Tabel 10. Rekomendasi perbaikan

No.	Kerentanan	Prioritas	Tindakan Mitigasi Utama
1	*Lack of Rate Limiting*	*High*	Terapkan pembatasan percobaan *login* melalui WAF seperti Cloudflare/Akamai dan terapkan fitur CAPTCHA.
2	*Exposed /server-status*	*High*	Nonaktifkan atau batasi akses ke modul *mod_status* Apache.
3	*Host Header Injection* (XSS)	*High*	*Upgrade* ke OJS versi 3.5.0-3
4	*Vulnerable JS Library*	*Medium*	Perbarui pustaka JavaScript.

Berdasarkan hasil pengujian, ditemukan empat kerentanan yang memerlukan penanganan dengan tingkat prioritas berbeda. Dua kerentanan dikategorikan sebagai prioritas tinggi (*High*), satu kerentanan pada tingkat menengah *(Medium)*, dan satu kerentanan lainnya juga berada pada tingkat tinggi namun berkaitan langsung dengan komponen inti sistem.

Kerentanan pertama adalah *Lack of Rate Limiting*, yang memungkinkan penyerang melakukan percobaan *login* secara berulang tanpa hambatan. Untuk mengatasinya, disarankan agar sistem menerapkan pembatasan percobaan *login* melalui WAF seperti Cloudflare atau Akamai, disertai dengan implementasi CAPTCHA sebagai lapisan perlindungan tambahan.

Kerentanan kedua berupa *endpoint/server-status* yang dapat diakses secara publik, sehingga berpotensi mengekspos informasi teknis server kepada pihak yang tidak berwenang. Mitigasi yang direkomendasikan adalah menonaktifkan atau membatasi akses ke modul *mod_status* pada Apache.

Kerentanan ketiga, yaitu *Host Header Injection* yang berpotensi memicu serangan XSS, ditemukan pada versi OJS yang sedang digunakan. Penanganan paling efektif untuk kerentanan ini adalah melakukan *upgrade* sistem ke OJS versi 3.5.0-3 yang telah memperbaiki celah tersebut.

Terakhir, ditemukan penggunaan pustaka JavaScript yang sudah usang (*Vulnerable JS Library*) dengan tingkat prioritas menengah. Meskipun tidak sekritis kerentanan sebelumnya, pembaruan pustaka JavaScript tetap perlu dilakukan untuk mencegah eksploitasi yang memanfaatkan celah pada versi lama.

Secara keseluruhan, keempat kerentanan tersebut hendaknya ditangani secara bertahap dengan mengutamakan kerentanan berkategori *High* terlebih dahulu, mengingat potensi dampaknya yang lebih signifikan terhadap keamanan dan ketersediaan sistem.

4. Kesimpulan

Evaluasi keamanan pada sistem *proceeding* berbasis OJS versi *legacy* telah berhasil mengidentifikasi sejumlah titik lemah yang signifikan melalui pendekatan terstruktur NIST SP 800-115. Hasil pengujian menunjukkan bahwa penggunaan sistem yang telah melewati masa dukungan (*end-of-life*) menyimpan risiko keamanan yang tinggi. Berdasarkan analisis risiko, kerentanan kritis ditemukan pada mekanisme autentikasi dan konfigurasi peladen yang dapat mengancam aspek kerahasiaan, keutuhan, serta ketersediaan data pada sistem informasi akademik institusi.

Secara praktis, temuan ini menunjukkan adanya ketergantungan pada teknologi usang dan kurangnya praktik *hardening* sistem yang menjadi faktor utama kerentanan. Secara teoretis, penelitian ini menunjukkan bahwa validasi eksploitasi terkontrol memberikan gambaran risiko yang lebih akurat dibandingkan sekadar pemindaian pasif. Hal ini menegaskan pentingnya evaluasi keamanan berkala sebagai bagian dari tata kelola teknologi informasi di perguruan tinggi untuk menghadapi tren ancaman aplikasi web yang terus berkembang.

Rekomendasi utama yang diusulkan adalah segera melakukan migrasi ke platform manajemen jurnal versi terbaru untuk memitigasi risiko dari kerentanan bawaan sistem lama. Selain pembaruan perangkat lunak, institusi disarankan untuk menerapkan kebijakan server *hardening* yang lebih ketat, mengoptimalkan protokol transmisi data, serta menambahkan mekanisme perlindungan tambahan pada gerbang autentikasi. Langkah-langkah tersebut merupakan prioritas strategis untuk memperkuat pertahanan sistem dari ancaman eksploitasi di masa mendatang.

Penelitian selanjutnya disarankan untuk memperluas cakupan pengujian dengan menggunakan pendekatan *gray-box* atau *white-box testing* sehingga potensi kerentanan pada tingkat kode sumber dan hak akses pengguna internal dapat dievaluasi. Selain itu, evaluasi kuantitatif pasca-mitigasi perlu dilakukan untuk mengukur efektivitas perbaikan yang telah diimplementasikan. Integrasi antara berbagai alat pemindai komersial juga diharapkan dapat memberikan cakupan deteksi yang lebih luas dalam mengamankan ekosistem publikasi ilmiah.

Ucapan Terima Kasih

Penulis menyampaikan terima kasih kepada ITDA Yogyakarta atas dukungan, izin penelitian, dan penyediaan data yang diperlukan dalam penelitian ini. Penulis juga menyampaikan apresiasi kepada para dosen Universitas Alma Ata atas bimbingan dan arahan selama proses penelitian.

Pernyataan

Kontribusi Penulis. I.A.: Konseptualisasi, perancangan metode pengujian teknis, pelaksanaan *penetration testing* (*planning, discovery, attack*), analisis data temuan, dan penulisan draf awal naskah. T.R.: Koordinasi perizinan (*rules of engagement*), evaluasi hasil,

serta peninjauan dan penyuntingan naskah. Y.W.: Validasi eksperimen eksploitasi, analisis kebutuhan lingkungan, dan peninjauan draf naskah. D.H.: Validasi exploitasi, analisis perhitungan *risk score*.

Pendanaan. Penelitian ini didanai secara mandiri oleh penulis.

Konflik Kepentingan. Penulis menyatakan tidak terdapat konflik kepentingan terkait publikasi artikel ini.

Ketersediaan Data. Data teknis berupa ringkasan hasil pemindaian dan dokumentasi *proof-of-concept* tersedia dari penulis korespondensi atas permintaan yang wajar. Detail *log* eksploitasi penuh tidak dibagikan secara publik demi menjaga keamanan data infrastruktur institusi yang menjadi objek studi kasus.

Penggunaan Kecerdasan Buatan (AI). Penulis menyatakan bahwa penggunaan alat berbasis kecerdasan buatan (AI) hanya terbatas pada aspek penyuntingan bahasa dan tidak mempengaruhi substansi ilmiah penelitian.

Daftar Referensi

[1] International Telecommunication Union (ITU), "Internet use continues to grow, but universality remains elusive, especially in low-income regions," *International Telecommunication Union (ITU)*, 2024. [Online]. Available: https://www.itu.int/itu-d/reports/statistics/2024/11/10/ff24-internet-use/ (accessed Sep. 6, 2025).

[2] E. Wahyudi, "Implementasi e-journal berbasis Open Journal System (OJS) untuk meningkatkan jumlah publikasi penelitian dosen IPDN Kampus NTB," *Jurnal Informatika & Komputer (EXPLORE)*, vol. 14, no. 1, pp. 35–41, Jan. 2024, doi: 10.35200/ex.v14i1.110.

[3] S. Khanna, "OJS Growth Statistics," *RPubs*, 2024. [Online]. Available: https://rpubs.com/saurabh90/ojs-stats-2024 (accessed Oct. 7, 2025).

[4] T. Rochmadi and I. Y. Pasa, "Pengukuran risiko dan evaluasi keamanan informasi menggunakan indeks keamanan informasi di BKD XYZ berdasarkan ISO 27001/SNI," *CyberSecurity dan Forensik Digital*, vol. 4, no. 1, pp. 38–43, May 2021, doi: 10.14421/csecurity.2021.4.1.2439.

[5] Direktorat Operasi Keamanan Siber, *Lanskap Keamanan Siber Indonesia 2024*, Jakarta, Indonesia: Badan Siber dan Sandi Negara (BSSN), 2024.

[6] W. Yunanri, "Analisis keamanan pada web aplikasi Open Journal System terhadap serangan Cross Site Scripting (XSS) menggunakan metode vulnerability assessment," *Digital Transformation Technology (Digitech)*, vol. 3, no. 1, pp. 83–90, Mar. 2023, doi: 10.47709/digitech.v3i1.2476.

[7] I. P. Saputra and A. Hidayat, "XSS injection vulnerability pada Open Journal Systems (OJS)," *Bulletin of Network Engineer and Informatics*, vol. 3, no. 1, pp. 29–33, Apr. 2025, doi: 10.59688/bufnets.v3i1.69.

[8] Mifthahuddin, H. J. Setyadi, and M. R. Ibrahim, "Penetration testing website e-journals metode NIST SP 800-115 dan OWASP," *Media Teknologi Informasi dan Komputer (METIK)*, vol. 9, no. 1, pp. 72–81, Jun. 2025, doi: 10.47002/metik.v9i1.1030.

[9] D. Supriadi, E. Suryadi, R. Muslim, and L. D. Samsumar, "Implementasi vulnerability assesment OWASP (Open Web Application Security Project) pada website Universitas Teknologi Mataram," *Journal of Data Analytics, Information, and Computer Science (JDAICS)*, vol. 1, no. 4, pp. 232–240, Oct. 2024, doi: 10.70248/jdaics.v1i4.1368.

[10] H. Kashyap, "PKP Open Journals System 3.3 - Cross-Site Scripting (XSS)," *Exploit Database*, 2022. [Online]. Available: https://www.exploit-db.com/exploits/50881 (accessed Oct. 15, 2025).
[11] Riswanda, "Kerentanan keamanan Open Journal System," *Universitas Muhammadiyah Surabaya*, 2025. [Online]. Available: https://lp2ihki.um-surabaya.ac.id/homepage/news_article?slug=kerentanan-keamanan-open-journal-system (accessed Oct. 5, 2025).
[12] H. Sulaeman and A. Takwim, "Analisa kualitas keamanan pada aplikasi SLiMS Akasia dengan metode NIST SP 800-115 dan OWASP," in *Proc. Seminar Nasional Penelitian (SEMNAS CORISINDO 2024)*, Bandung, Indonesia, Oct. 2024, pp. 500–506.
[13] Y. Ginanjar, "Strategi Indonesia membentuk cyber security dalam menghadapi ancaman cyber crime melalui Badan Siber dan Sandi Negara," *Jurnal Dinamika Global*, vol. 7, no. 2, pp. 295–316, Dec. 2022, doi: 10.36859/jdg.v7i02.1187.
[14] Purwadi and Irwansyah, "Prospek dan tantangan industri penerbitan jurnal dan prosiding melalui teknologi e-publishing di era digital," *BACA: Jurnal Dokumentasi dan Informasi*, vol. 41, no. 1, pp. 87–98, Jun. 2020, doi: 10.14203/j.baca.v41i1.509.
[15] E. Z. Darojat, E. Sediyono, and I. Sembiring, "Vulnerability assessment website e-government dengan NIST SP 800-115 dan OWASP menggunakan web vulnerability scanner," *Jurnal Sistem Informasi Bisnis*, vol. 12, no. 1, pp. 36–44, Sep. 2022, doi: 10.21456/vol12iss1pp36-44.
[16] OWASP, "OWASP Risk Rating Methodology," *OWASP*, 2025. [Online]. Available: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology (accessed Nov. 7, 2025).
[17] S. A. Nugroho and T. Rochmadi, "Analisis keamanan sistem informasi Pusaka Magelang menggunakan Open Web Application Security Project (OWASP) dan Information Systems Security Assessment Framework (ISSAF)," *Cyber Security dan Forensik Digital*, vol. 7, no. 1, pp. 56–61, May 2024, doi: 10.14421/csecurity.2024.7.1.4555.
[18] A. Luthfi, E. H. Nurkifli, and I. Maulana, "Security testing (white box penetration testing) pada authentication sistem login website," *Jurnal Ilmiah Informatika (JIF)*, vol. 13, no. 2, pp. 184–189, Sep. 2025, doi: 10.33884/jif.v13i02.10660.
[19] S. Handaya and R. Islamadina, "Implementasi penetration testing pada aplikasi web sistem evaluasi data bidang TIK Polda Aceh menggunakan metode OWASP dan NIST SP 800-115," *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, vol. 9, no. 1, pp. 27–41, Mar. 2025, doi: 10.22373/cj.v9i1.27978.
[20] A. A. Chandra, A. T. Zy, and A. Nugroho, "Penerapan teknik penetration testing terhadap Cross Site Sripting (XSS) dalam pengembangan website," *RABIT: Jurnal Teknologi dan Sistem Informasi Univrab*, vol. 9, no. 2, pp. 262–270, Jul. 2024, doi: 10.36341/rabit.v9i2.4822.
[21] M. K. Putri and A. R. Hakim, "Perancangan manajemen risiko keamanan informasi layanan jaringan MKP berdasarkan kerangka kerja ISO/IEC 27005:2018 dan NIST SP 800-30 Revisi 1," *Jurnal Info Kripto*, vol. 15, no. 3, pp. 133–141, Nov. 2021, doi: 10.56706/ik.v15i3.34.
[22] S. Aprianti, R. P. Sari, and I. Rusi, "Manajemen risiko keamanan Simbada menggunakan metode NIST SP 800-30 Revisi 1 dan kontrol ISO/IEC 27001:2013,"

Jurnal Buana Informatika, vol. 14, no. 1, pp. 50–59, Apr. 2023, doi: 10.24002/jbi.v14i01.7043.

[23] F. Septian, M. H. Arfian, J. S. Asri, and B. Tjahjono, "Pengujian keamanan website dengan metode penetration testing (studi kasus: Universitas Esa Unggul)," *Innovative: Journal of Social Science Research*, vol. 4, no. 5, pp. 3629–3647, 2024.

[24] M. B. Imtias, K. Umam, H. Mustofa, and M. H. Subowo, "Comparative analysis of penetration testing frameworks: OWASP, PTES, and NIST SP 800-115 for detecting web application vulnerabilities," *Journal of Applied Informatics and Computing (JAIC)*, vol. 9, no. 6, pp. 3689–3696, 2025, doi: 10.30871/jaic.v9i6.9846.

[25] N. Hidayat and M. A. Nugroho, "Analisis celah keamanan pada website SMA Negeri 3 Berau dengan metode penetration testing," *Journal of Information System Management (JOISM)*, vol. 6, no. 2, pp. 102–108, 2025, doi: 10.24076/joism.2025v6i2.1858.