# EVALUASI DAN MITIGASI SERANGAN XSS DAN CSRF PADA OPEN JOURNAL SYSTEMS VERSI 3.5 MENGGUNAKAN PENETRATION TESTING DAN CVSS SCORING

**Donny Saputra[1], Leonardus Sandy Ade Putra[2], Fitri Imansyah[3]**
donnysaputra1@student.untan.ac.id[1], leonardusandy@ee.untan.ac.id[2],
fitri.imansyah@ee.untan.ac.id[3]
**Universitas Tanjungpura**

**Abstrak**

Open Journal Systems (OJS) merupakan aplikasi pengelolaan jurnal berbasis web yang rentan terhadap serangan Cross-Site Scripting (XSS) dan Cross-Site Request Forgery (CSRF). Penelitian ini bertujuan mengevaluasi kerentanan keamanan pada OJS versi 3.5, mengukur tingkat risikonya menggunakan Common Vulnerability Scoring System (CVSS) v4.0, menerapkan mitigasi pada kode sumber, serta memvalidasi efektivitasnya melalui retesting. Metode yang digunakan adalah Penetration Testing yang menggunakan black-box testing dengan mengacu pada OWASP Web Security Testing Guide (WSTG) dan Penetration Testing Execution Standard (PTES). Tahapan penelitian meliputi pemindaian kerentanan, pengujian eksploitasi, analisis kode sumber, penilaian risiko menggunakan CVSS, implementasi secure coding, dan retesting. Hasil penelitian menunjukkan bahwa pendekatan Penetration Testing mampu mengidentifikasi kerentanan XSS dan CSRF pada OJS versi 3.5. Mitigasi yang diterapkan berhasil meningkatkan keamanan aplikasi dan mengurangi peluang eksploitasi berdasarkan hasil pengujian ulang. Penelitian ini menunjukkan bahwa kombinasi Penetration Testing, CVSS, dan secure coding merupakan pendekatan yang efektif untuk mengevaluasi dan meningkatkan keamanan OJS terhadap serangan XSS dan CSRF.
**Kata Kunci:** Open Journal Systems, Penetration Testing, XSS, CSRF, CVSS.

***ABSTRACT***

*Open Journal Systems (OJS) is a web-based journal management application that is vulnerable to Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) attacks. This study aims to evaluate security vulnerabilities in OJS version 3.5, assess their risk levels using the Common Vulnerability Scoring System (CVSS) v4.0, implement mitigation measures in the source code, and validate their effectiveness through retesting. The research employs a Penetration Testing approach using black-box testing based on the OWASP Web Security Testing Guide (WSTG) and the Penetration Testing Execution Standard (PTES). The research stages include vulnerability scanning, exploitation testing, source code analysis, risk assessment using CVSS, implementation of secure coding practices, and retesting. The results indicate that the Penetration Testing approach successfully identified XSS and CSRF vulnerabilities in OJS version 3.5. The implemented mitigation measures effectively enhanced the application's security and reduced the likelihood of successful exploitation based on the retesting results. This study demonstrates that the combination of Penetration Testing, CVSS, and secure coding practices is an effective approach for evaluating and improving the security of OJS against XSS and CSRF attacks.*
***Keywords:*** *Open Journal Systems, Penetration Testing, XSS, CSRF, CVSS.*

## PENDAHULUAN

Perkembangan teknologi informasi telah membawa perubahan signifikan dalam berbagai bidang, termasuk dunia akademik dan penelitian. Salah satu platform yang banyak digunakan dalam publikasi ilmiah adalah Open Journal System (OJS), sebuah sistem manajemen jurnal berbasis sumber terbuka (open source) yang digunakan oleh banyak institusi pendidikan dan penerbit jurnal di seluruh dunia. Dikembangkan oleh Public Knowledge Project (PKP), OJS menjadi pilihan utama bagi banyak institusi pendidikan karena selain sifatnya yang open source, OJS juga dinilai ringan, dan memiliki

komunitas pengguna yang luas. [1] Pada versi terbarunya yaitu OJS versi 3.5., menjadikan platform ini sangat diminati karena popularitasnya dan fiturnya yang terus bertambah. Namun disisi lain, keamanan sistem menjadi aspek yang sangat penting untuk diperhatikan, mengingat potensi ancaman siber yang terus meningkat.

Keamanan web server yang menjalankan OJS menjadi salah satu faktor utama dalam menjaga integritas data dan aksesibilitas informasi. Namun, karena sifatnya yang open source, OJS juga menjadi target potensial bagi berbagai jenis serangan. Penelitian terdahulu dan laporan komunitas telah mengindikasikan adanya celah keamanan pada OJS, khususnya pada versi-versi lama. Dua di antara kerentanan yang paling umum dan berdampak tinggi adalah Cross-Site Scripting (XSS) dan Cross-Site Request Forgery (CSRF). Serangan XSS memungkinkan penyerang untuk menyisipkan skrip berbahaya ke dalam situs web yang dilihat oleh pengguna lain, yang dapat berujung pada pencurian sesi, pengambilhalihan akun, atau perusakan tampilan situs. [2] Sementara itu, serangan CSRF dapat memaksa pengguna yang terautentikasi untuk melakukan tindakan yang tidak diinginkan, seperti mengubah data jurnal atau mengirimkan artikel palsu tanpa diketahui. [3]

Beberapa penelitian sebelumnya telah melakukan analisis keamanan pada OJS menggunakan berbagai metodologi, seperti ISSAF, OWASP, atau pemindaian kerentanan Black-Box. [4] Namun, penelitian-penelitian tersebut sering kali berhenti pada tahap deteksi kerentanan. Terdapat kekosongan riset yang signifikan dalam hal pembuktian eksploitasi secara praktis, pengukuran dampak risiko secara kuantitatif menggunakan standar industri seperti Common Vulnerability Scoring System (CVSS) [5] , serta implementasi dan validasi solusi mitigasi teknis. Tanpa siklus pengujian yang lengkap mulai dari deteksi, eksploitasi, mitigasi, hingga pengujian ulang (retesting), rekomendasi keamanan yang dihasilkan bersifat teoretis dan belum terbukti efektivitasnya.

Oleh karena itu, penelitian ini diusulkan untuk mengisi kekosongan tersebut. Dengan menggunakan pendekatan penetration testing (black-box), penelitian ini akan secara komprehensif mengevaluasi kerentanan XSS dan CSRF pada OJS versi 3.5. Pendekatan tersebut memungkinkan pengujian keamanan secara aktif guna menemukan kelemahan sistem sebelum dieksploitasi oleh pihak yang tidak bertanggung jawab. Selanjutnya, tingkat risiko dari setiap kerentanan yang ditemukan akan diukur menggunakan CVSS v4.0, dan solusi mitigasi teknis berupa patch pada kode sumber akan diterapkan. Sebagai tahap akhir, pengujian ulang akan dilakukan untuk memverifikasi bahwa mitigasi yang diimplementasikan berhasil menutup celah keamanan yang ada.

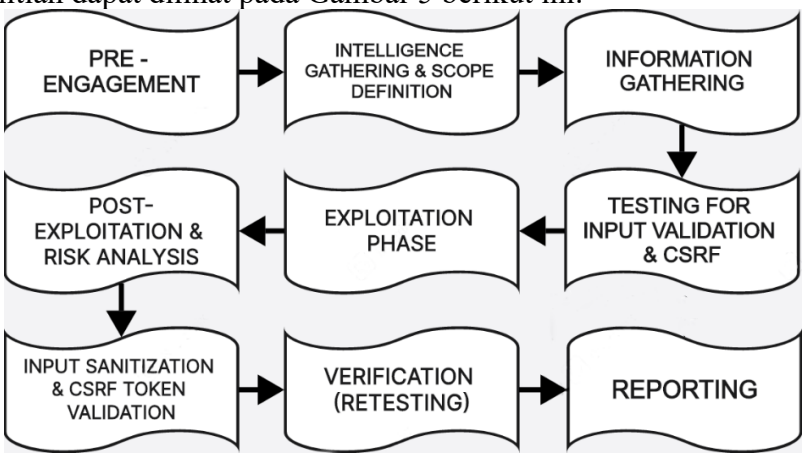
Melalui penelitian ini, diharapkan dapat diperoleh wawasan yang lebih dalam mengenai tingkat keamanan OJS serta langkah-langkah mitigasi yang dapat dilakukan untuk mencegah serangan siber. Selain itu, hasil penelitian ini juga diharapkan dapat menjadi referensi bagi administrator sistem dan pengelola jurnal dalam meningkatkan perlindungan terhadap ancaman keamanan yang berpotensi merugikan. Pentingnya penelitian ini juga didasari oleh meningkatnya jumlah insiden keamanan yang menargetkan platform publikasi ilmiah. Banyak jurnal akademik berbasis OJS mengalami kendala teknis akibat serangan siber yang dapat mengganggu ketersediaan layanan dan merusak reputasi institusi yang mengelolanya. [6]

Lebih jauh, hasil penelitian ini diharapkan dapat menjadi rujukan bagi pengembang OJS dalam meningkatkan fitur keamanan di versi-versi mendatang. Dengan adanya pengujian yang lebih komprehensif dan berbasis metode penetration testing yang sistematis, diharapkan OJS dapat menjadi platform publikasi yang lebih aman dan andal dalam mendukung perkembangan ilmu pengetahuan di era digital ini.

## METODOLOGI

Dalam suatu penelitian dibutuhkan sebuah metode dalam penyelesaiannya, maka penulis telah menetapkan metode penelitian terintegrasi yang mengacu pada OWASP Testing Guide v4 dan PTES (Penetration Testing Execution Standard). OWASP Testing Guide v4 merupakan standar internasional yang khusus dirancang untuk pengujian keamanan aplikasi web. Panduan ini menyediakan metode sistematis untuk mengidentifikasi kerentanan aplikasi web, menguji mekanisme autentikasi dan otorisasi, menguji input validation, session management, dan request handling. [28] Karena OJS adalah aplikasi web berbasis PHP, maka OWASP Testing Guide v4 sangat relevan untuk menguji serangan XSS dan CSRF, yang merupakan bagian dari kategori Injection dan Broken Access Control.

Sedangkan untuk PTES sendiri adalah standar yang mendefinisikan alur lengkap penetration testing, mulai dari perencanaan hingga pelaporan. PTES tidak hanya fokus pada teknis pengujian, tetapi juga pada validasi eksploitasi, analisis risiko dan dokumentasi hasil. [29] Integrasi kedua standar tersebut dipilih agar penelitian memiliki alur pengujian yang sistematis dalam penetration testing aplikasi web. Dengan pendekatan ini, proses penelitian tidak hanya berfokus pada penemuan kerentanan, tetapi juga pada validasi hasil pengujian, analisis tingkat risiko, serta penyusunan rekomendasi mitigasi. Langkah-langkah penelitian dapat dilihat pada Gambar 5 berikut ini:




Gambar 3. 3 Tahapan Pelaksanaan Metode Penetration Testing

*(Sumber: OWASP Testing Guide v4 dan PTES)*

Dari Gambar 5 di atas, dapat dijelaskan dalam Tabel 3 berikut ini.

Tabel 3.3 Penjelasan Tahapan Pelaksanaan Metode Penetration Testing

| NO. | TAHAPAN | KETERANGAN |
|---|---|---|
| 1 | Pre-Engagement | Tahap awal untuk menentukan tujuan pengujian, objek penelitian, batasan pengujian, serta persetujuan pengujian agar proses *penetration testing* dilakukan secara legal dan terkontrol. |
| 2 | Intelligence Gathering & Scope Definition | Tahap pengumpulan informasi awal dan penentuan ruang lingkup pengujian, meliputi modul OJS, peran pengguna serta fitur yang menjadi objek pengujian. |
| 3 | Information Gathering | Tahap pengumpulan informasi teknis sistem seperti *endpoint* URL, parameter input, metode HTTP, *cookie*, serta mekanisme sesi tanpa melakukan eksploitasi aktif. |
| 4 | Testing for Input Validation & CSRF | Tahap pengujian untuk mendeteksi kelemahan validasi input yang berpotensi menyebabkan XSS serta pengujian mekanisme proteksi CSRF pada permintaan sensitif dalam sistem OJS. |
| 5 | Exploitation Phase | Tahap pembuktian kerentanan melalui eksploitasi terbatas atau *proof of concept* (PoC) terhadap celah keamanan yang ditemukan. |

| | | |
|---|---|---|
| 6 | Post-Exploitation & Risk Analysis | Tahap analisis dampak setelah eksploitasi berhasil, termasuk potensi penyalahgunaan sistem dan penilaian tingkat risiko menggunakan metode CVSS. |
| 7 | Input Sanitization & CSRF Token Validation | Tahap penerapan mitigasi dengan sanitasi input dan validasi output untuk mencegah XSS serta penerapan dan validasi token CSRF pada permintaan sensitif |
| 8 | Verification (Retesting) | Tahap pengujian ulang setelah mitigasi diterapkan untuk memastikan bahwa kerentanan telah berhasil ditutup dan tidak dapat dieksploitasi kembali. |
| 9 | Reporting | Tahap pendokumentasian seluruh proses, temuan, analisis risiko, dan hasil pengujian dalam bentuk tabel, gambar, serta narasi analitis. |

## HASIL DAN PEMBAHASAN

### Analisis Hasil Pengujian

Berdasarkan seluruh rangkaian pengujian yang telah dilakukan terhadap endpoint-endpoint yang menjadi objek penelitian, tidak ditemukan bukti adanya kerentanan Cross-Site Scripting (XSS) maupun Cross-Site Request Forgery (CSRF) pada aplikasi Open Journal Systems (OJS) versi 3.5 yang diuji. Pengujian dilakukan melalui dua tahapan, yaitu automated vulnerability scanning menggunakan OWASP ZAP untuk mengidentifikasi potensi kerentanan, kemudian dilanjutkan dengan verifikasi manual menggunakan Burp Suite Community Edition guna memastikan apakah potensi kerentanan tersebut benar-benar dapat dieksploitasi.

Pada pengujian Stored XSS dan Reflected XSS, beberapa endpoint yang dipilih berdasarkan hasil identifikasi OWASP ZAP menunjukkan indikasi potensi kerentanan, seperti konfigurasi Content Security Policy (CSP) yang masih mengizinkan penggunaan unsafe-inline. Namun, setelah dilakukan verifikasi manual dengan penyisipan payload pada berbagai field yang diuji, payload hanya ditampilkan sebagai teks atau disanitasi oleh aplikasi sehingga tidak dieksekusi oleh browser. Hal tersebut menunjukkan bahwa data yang disimpan maupun ditampilkan kembali tidak dapat dimanfaatkan untuk menjalankan kode JavaScript berbahaya.

Sementara itu, pada pengujian CSRF, seluruh endpoint yang diuji menunjukkan adanya mekanisme validasi terhadap request yang mengubah data (state-changing request). Verifikasi dilakukan melalui beberapa skenario, seperti penghapusan CSRF Token, perubahan nilai CSRF Token, modifikasi header Origin, serta penghapusan header Referer. Hasil pengujian menunjukkan bahwa request yang telah dimodifikasi tidak menyebabkan perubahan data pada aplikasi. Pada beberapa endpoint, server mengembalikan respons HTTP 400 Bad Request atau 403 Forbidden, sedangkan pada endpoint lainnya server tetap memberikan respons HTTP 200 OK, namun perubahan data tidak diproses setelah diverifikasi melalui antarmuka aplikasi. Hasil tersebut menunjukkan bahwa kode respons HTTP saja tidak dapat dijadikan indikator keberhasilan eksploitasi, sehingga verifikasi terhadap perubahan status aplikasi menjadi bagian penting dalam proses analisis.

Berdasarkan keseluruhan hasil pengujian tersebut, dapat disimpulkan bahwa mekanisme validasi input, sanitasi data, serta perlindungan terhadap CSRF yang diterapkan pada endpoint-endpoint yang diuji telah bekerja secara efektif dalam mencegah eksploitasi terhadap kedua jenis kerentanan tersebut. Dengan demikian, tidak ditemukan kerentanan XSS maupun CSRF pada endpoint-endpoint yang menjadi objek penelitian.

Meskipun demikian, hasil penelitian ini hanya berlaku pada endpoint dan skenario pengujian yang telah ditetapkan dalam ruang lingkup penelitian. Oleh karena itu, pengujian

terhadap endpoint lain, modul tambahan, maupun jenis kerentanan yang berbeda masih diperlukan untuk memperoleh gambaran keamanan aplikasi secara lebih menyeluruh. Pendekatan ini penting karena keamanan aplikasi bersifat dinamis dan dapat dipengaruhi oleh perubahan konfigurasi, pembaruan sistem, maupun penambahan fitur di masa mendatang.

## KESIMPULAN

Berdasarkan hasil penelitian mengenai pengujian keamanan aplikasi Open Journal Systems (OJS) versi 3.5 terhadap kerentanan Cross-Site Scripting (XSS) dan Cross-Site Request Forgery (CSRF), maka dapat disimpulkan sebagai berikut.

1. Pengujian menggunakan OWASP ZAP berhasil mengidentifikasi beberapa endpoint yang berpotensi mengandung kerentanan keamanan, seperti endpoint Submission Title, Abstract, Keywords, References, Review Discussion, Search Submission, Save New Submission, Update Submission Metadata, Submission Discussion Reply, Accept Review Assignment, dan Submit Review Recommendation/Decision. Selain itu, OWASP ZAP juga menghasilkan beberapa peringatan konfigurasi keamanan, seperti Content Security Policy (CSP) yang masih mengizinkan unsafe-inline, Strict-Transport-Security Header Not Set, X-Content-Type-Options Header Missing, serta Information Disclosure melalui header X-Powered-By. Temuan tersebut menunjukkan adanya konfigurasi keamanan yang masih dapat ditingkatkan, meskipun belum dapat dinyatakan sebagai kerentanan yang dapat dieksploitasi.
2. Verifikasi manual menunjukkan bahwa tidak ditemukan kerentanan Stored XSS maupun (Reflected XSS pada seluruh endpoint yang diuji. Payload yang dikirimkan melalui berbagai field tidak dieksekusi oleh browser, melainkan ditampilkan sebagai teks atau telah diproses melalui mekanisme sanitasi sehingga tidak menghasilkan eksekusi kode JavaScript.
3. Pengujian terhadap kerentanan CSRF juga menunjukkan bahwa seluruh endpoint yang diuji telah menerapkan mekanisme validasi terhadap request yang mengubah data (state-changing request). Modifikasi berupa penghapusan maupun perubahan CSRF Token, perubahan header Origin, serta penghapusan header Referer tidak menyebabkan perubahan data pada aplikasi. Pada beberapa endpoint server memberikan respons 400 Bad Request atau 403 Forbidden, sedangkan pada endpoint lainnya server tetap mengembalikan 200 OK, namun perubahan data tidak diproses setelah dilakukan verifikasi melalui antarmuka aplikasi. Hasil tersebut menunjukkan bahwa mekanisme perlindungan CSRF pada endpoint yang diuji berjalan secara efektif.
4. Berdasarkan keseluruhan hasil pengujian, tidak ditemukan kerentanan XSS dan CSRF pada endpoint-endpoint yang menjadi ruang lingkup penelitian. Oleh karena itu, tahapan penilaian tingkat keparahan menggunakan Common Vulnerability Scoring System (CVSS) tidak dilakukan, karena metode tersebut digunakan untuk mengukur tingkat risiko dari kerentanan yang telah terkonfirmasi. Selain itu, proses sanitasi atau perbaikan juga tidak dilakukan karena penelitian ini tidak menemukan kerentanan yang memerlukan tindakan mitigasi. Meskipun demikian, beberapa temuan konfigurasi keamanan dari OWASP ZAP tetap direkomendasikan untuk diperbaiki sebagai langkah hardening sistem guna meningkatkan keamanan aplikasi secara keseluruhan.

**Saran**

Berdasarkan kesimpulan yang ada, saran-saran berikut dapat dibuat untuk meningkatkan keamanan jaringan:

1. Pengelola aplikasi disarankan untuk memperbaiki konfigurasi keamanan yang masih teridentifikasi oleh OWASP ZAP, seperti menambahkan header Strict-Transport-Security (HSTS), X-Content-Type-Options, serta mengurangi penggunaan konfigurasi Content Security Policy (CSP) yang masih mengizinkan unsafe-inline, sehingga tingkat keamanan aplikasi menjadi lebih baik.
2. Pengujian keamanan sebaiknya dilakukan secara berkala, terutama setelah proses pembaruan versi OJS, instalasi plugin baru, maupun perubahan konfigurasi server, agar potensi kerentanan dapat dideteksi sejak dini sebelum dimanfaatkan oleh pihak yang tidak bertanggung jawab.
3. Walaupun kerentanan tidak ditemukan, untuk penelitian selanjutnya disarankan untuk memperluas ruang lingkup pengujian dengan menguji lebih banyak endpoint, peran pengguna (role), maupun jenis kerentanan lain, seperti SQL Injection (SQLi), Broken Access Control, Server-Side Request Forgery (SSRF), XML External Entity (XXE), atau kerentanan lain yang termasuk dalam OWASP Top 10, sehingga dapat memberikan gambaran keamanan aplikasi yang lebih komprehensif.

## DAFTAR PUSTAKA


A. Barth, C. Jackson, and J. C. Mitchell, “Robust defenses for cross-site request forgery,” Proc. ACM Conf. Comput. Commun. Secur., pp. 75–87, 2008, doi: 10.1145/1455770.1455782.

A. Maspupah, “Literature Review: Advantages and Disadvantages of Black Box and White Box Testing Methods,” J. Techno Nusa Mandiri, vol. 21, no. 2, pp. 151–162, 2024, doi: 10.33480/techno.v21i2.5776.

Andri Prima Nugroho, “Panduan Open Journal System,” vol. 1, no. 3, p. 57, 2010, [Online]. Available: https://repo.unsrat.ac.id/130/2/panduan-open-journal-system-biotechnology.pdf

B. D. Edgar and J. Willinsky, “A Survey of Scholarly Journals Using Open Journal Systems,” Sch. Res. Commun., vol. 1, no. 2, pp. 1–40, 2010, doi: 10.22230/src.2010v1n2a24.

D. Stuttard and M. Pinto, The Web Application Hacker’s Handbook: Finding and Exploiting Security Flaws, vol. 7. 2011. [Online]. Available: https://digtvbg.com/files/books-for-hacking/The Web Application Hacker%27s Handbook - Finding and Exploiting Security Flaws%2C 2nd Edition by Dafydd Stuttard%2C Marcus Pinto.pdf

FIRST, “CVSS v4.0 Specification.” Accessed: Apr. 03, 2026. [Online]. Available: https://www.first.org/cvss/v4.0/specification-document

G. Guntoro, L. Costaner, and M. Musfawati, “Analisis Keamanan Web Server Open Journal System (Ojs) Menggunakan Metode Issaf Dan Owasp (Studi Kasus Ojs Universitas Lancang Kuning),” JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform., vol. 5, no. 1, p. 45, 2020, doi: 10.29100/jipi.v5i1.1565.

Georgia Weidman, Penetration Testing: A Hands-On Introduction to Hacking. 2013. doi: 10.1016/B978-0-12-394397-2.00030-1.

I. P. Saputra and A. Hidayat, “Xss Injection Vulnerability Pada Open Journal Systems (Ojs),” Bull. Netw. Eng. Informatics, vol. 3, no. 1, p. 29, 2025, doi: 10.59688/bufnets.v3i1.69.

J. Grossman, R. Hansen, P. D. Petkov, A. Rager, and S. Fogie, XSS attacks: Cross site scripting exploits and defense. 2007. doi: 10.1016/B978-1-59749-154-9.X5000-8.

J. T. Santoso and B. Raharjo, “Performance Evaluation of Penetration Testing Tools in Diverse Computer System Security Scenarios,” J. Teknol. Inf. Dan Komun., vol. 13, no. 2, pp. 132–159, 2022, [Online]. Available: https://ejurnal.provisi.ac.id/index.php/JTIKP/article/view/851

L. Allodi and F. Massacci, “Comparing vulnerability severity and exploits using case-control studies,” ACM Trans. Inf. Syst. Secur., vol. 17, no. 1, 2014, doi: 10.1145/2630069.

L. Stallings, William; Brown, Computer Security Principles and Practice (4th Edition). 2018.

[Online]. Available: https://ebooks.karbust.me/Technology/Stallings, William_ Brown, Lawrie - Computer Security_ Principles and Practice-Pearson Education Limited (2018).pdf
L. Zuchri, "Open Journal Systems: Solusi Pengelolaan Jurnal Ilmiah," Open J. Syst. Solusi Pengelolaan J. Ilm., pp. 1–6, 2008, [Online]. Available: https://ilmukomputer.org/wp-content/uploads/2009/05/zuchri-ojs-solusi-pengelolaan-jurnal-ilmiah.pdf
M. Hasibuan and A. M. Elhanafi, "Penetration Testing Sistem Jaringan Komputer Menggunakan Kali Linux untuk Mengetahui Kerentanan Keamanan Server dengan Metode Black Box," sudo J. Tek. Inform., vol. 1, no. 4, pp. 171–177, 2022, doi: 10.56211/sudo.v1i4.160.
OWASP Cheat Sheet Series, "Cross Site Scripting Prevention Cheat Sheet." Accessed: Apr. 04, 2026. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
OWASP Cheat Sheet Series, "Cross-Site Request Forgery Prevention Cheat Sheet." Accessed: Apr. 04, 2026. [Online]. Available:https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
OWASP Foundation, "Cross Site Request Forgery (CSRF)." [Online]. Available: https://owasp.org/www-community/attacks/csrf
OWASP Foundation, "OWASP Code Review Guide." Accessed: Apr. 04, 2026. [Online]. Available: https://owasp.org/www-project-code-review-guide/
OWASP Foundation, "OWASP Secure Coding Practices-Quick Reference Guide." Accessed: May 20, 2026. [Online]. Available: https://owasp.org/www-project-secure-coding-practices-quick-reference-guideFIRST, "Common Vulnerability Scoring System." [Online]. Available: https://www.first.org/cvss/
OWASP Foundation, "OWASP Top Ten Web Application Security Risks." Accessed: Apr. 04, 2026. [Online]. Available: https://owasp.org/www-project-top-ten/
OWASP Foundation, "OWASP Web Security Testing Guide." Accessed: Apr. 02, 2026. [Online].
Available: https://owasp.org/www-project-web-security-testing-guide/
OWASP Foundation, "Penetration Testing Methodologies." Accessed: Apr. 01, 2026. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/v41/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies
PTES, "The penetration testing execution standard." Accessed: Apr. 01, 2026. [Online].Available: http://www.pentest-standard.org/index.php/Main_Page
Public Knowledge Project (PKP), "Open Journal Systems." Accessed: Apr. 01, 2026. [Online]. Available: https://pkp.sfu.ca/software/ojs/
Public Knowledge Project (PKP), "PKP Community Forum." Accessed: Apr. 01, 2026. [Online]. Available: https://forum.pkp.sfu.ca/
Y. W, "Analisis Keamanan Pada Web Aplikasi Open Journal System Terhadap Serangan Cross Site Scripting (XSS) Menggunakan Metode Vulnerability Assessment," Digit. Transform. Technol., vol. 3, no. 1, pp. 83–90, 2023, [Online]. Available: https://jurnal.itscience.org/index.php/digitech/article/view/2476
Y. W, R. Anto, D. Teguh Yuwono, and Y. Yuliadi, "Deteksi Serangan Vulnerability Pada Open Jurnal System Menggunakan Metode Black-Box," J. Inform. dan Rekayasa Elektron., vol. 4, no. 1, pp. 68–77, 2021, doi: 10.36595/jire.v4i1.365.