JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 1, Februari 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v13i1.9517
Hal 305-315
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Implementasi Arsitektur Keamanan Terintegrasi IDS, WAF Dan FIM Berbasis Wazuh Pada Platform Open Jurnal System Mengacu Pada NIST Cybersecurity Framework

William Carey Kornelius*, Pitrasacha Adytia, Ahmad Fahrijal Pukeng

Program Studi Teknik Informatika, STMIK Widya Cipta Dharma, Samarinda, Indonesia
Email: [1]2243057@wicida.ac.id, [2]pitra@wicida.ac.id, [3]pukeng@wicida.ac.id
Email Penulis Korespondensi: pitra@wicida.ac.id
Submitted **16-01-2026;** Accepted **19-02-2026;** Published **28-02-2026**

Abstrak

Pemanfaatan Open Journal Systems (OJS) sebagai platform publikasi ilmiah menghadapi ancaman keamanan seperti SQL Injection, Cross-Site Scripting (XSS), dan penyisipan webshell yang berpotensi mengganggu integritas serta ketersediaan layanan. Penelitian ini bertujuan merancang dan mengevaluasi arsitektur keamanan terintegrasi berbasis Wazuh melalui implementasi Intrusion Detection System (IDS), Web Application Firewall (WAF), dan File Integrity Monitoring (FIM) dengan pendekatan NIST Cybersecurity Framework. Metodologi penelitian meliputi identifikasi kerentanan pada 11 jurnal di 7 universitas, perancangan arsitektur defense-in-depth, serta pengujian penetrasi terkontrol berbasis skenario OWASP Top 10. Hasil pengujian terhadap 30 skenario serangan menunjukkan tingkat deteksi 100% pada SQL Injection dan penyisipan webshell, serta 80% pada skenario XSS. Sistem mampu memblokir permintaan berbahaya dengan respons 403 Forbidden serta menghasilkan peringatan real-time melalui korelasi log terpusat pada Wazuh. Meskipun demikian, ditemukan potensi false-positive pada beberapa rule generik sehingga diperlukan proses fine-tuning untuk menyesuaikan karakteristik trafik OJS. Secara keseluruhan, integrasi keamanan terpusat ini terbukti meningkatkan kemampuan deteksi dan respons insiden secara terukur.

Kata Kunci: Ids; Waf; Integrasi Keamanan; Keamanan Aplikasi Web; Fim

Abstract

The utilization of Open Journal Systems (OJS) as a scientific publishing platform faces significant security threats, including SQL Injection, Cross-Site Scripting (XSS), and webshell injection, which may compromise data integrity and service availability. This study aims to design and evaluate an integrated security architecture based on Wazuh through the implementation of an Intrusion Detection System (IDS), Web Application Firewall (WAF), and File Integrity Monitoring (FIM) using the NIST Cybersecurity Framework approach. The research methodology includes vulnerability identification across 11 journals in 7 universities, the development of a defense-in-depth architecture, and controlled penetration testing based on OWASP Top 10 scenarios. Testing results from 30 attack scenarios demonstrate a 100% detection rate for SQL Injection and webshell injection, and an 80% detection rate for XSS attacks. The system successfully blocks malicious requests with 403 Forbidden responses and generates real-time alerts through centralized log correlation in Wazuh. However, potential false positives were observed in several generic security rules, indicating the need for rule fine-tuning to align with OJS traffic characteristics. Overall, the integrated security approach measurably enhances threat detection and incident response capabilities.

Keywords: Ids; Waf; Security Integration; Web Application Security; Fim

1. PENDAHULUAN

Open Journal Systems (OJS) telah banyak digunakan oleh perguruan tinggi sebagai platform pengelolaan jurnal ilmiah secara daring, mulai dari proses pengiriman naskah hingga publikasi artikel. Karena dapat diakses secara terbuka melalui internet, sistem ini menyimpan berbagai informasi penting, seperti data penulis, reviewer, serta konfigurasi sistem. Kondisi tersebut menjadikan aspek keamanan sebagai faktor yang tidak dapat diabaikan dalam menjaga keberlangsungan layanan dan integritas data publikasi ilmiah [1].

Dalam praktiknya, berbagai insiden keamanan pada OJS masih sering terjadi. Serangan seperti SQL Injection, Cross-Site Scripting (XSS), penyisipan webshell, hingga defacement umumnya memanfaatkan kelemahan konfigurasi server, plugin yang tidak diperbarui, maupun minimnya mekanisme pemantauan keamanan [2][3]. Dampaknya tidak hanya berupa gangguan tampilan atau layanan, tetapi juga berpotensi mengakibatkan akses tidak sah terhadap sistem dan data jurnal.

Pendekatan pengamanan yang hanya mengandalkan satu mekanisme proteksi sering kali belum memadai untuk menghadapi pola serangan yang semakin kompleks. Serangan terhadap aplikasi web tidak jarang melibatkan eksploitasi pada lapisan aplikasi sekaligus modifikasi berkas pada sisi server. Oleh karena itu, diperlukan pendekatan yang mampu mengawasi lalu lintas jaringan, aktivitas aplikasi, serta integritas berkas secara terpadu [4][5].

Penelitian ini menerapkan integrasi Intrusion Detection System (IDS), Web Application Firewall (WAF), dan File Integrity Monitoring (FIM) dalam ekosistem Wazuh sebagai pusat korelasi dan analisis log. IDS digunakan untuk mengidentifikasi anomali lalu lintas jaringan seperti aktivitas brute force dan pemindaian sistem, WAF berfungsi memfilter serta memblokir serangan terhadap aplikasi web seperti SQL Injection dan XSS sebelum mencapai server, sedangkan FIM memantau perubahan tidak sah pada berkas inti OJS akibat penyisipan webshell atau modifikasi sistem. Seluruh peristiwa keamanan dari ketiga mekanisme tersebut dikorelasikan secara terpusat untuk mendukung proses deteksi dan respons insiden yang lebih terkoordinasi [6][7][8][9].

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 1, Februari 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v13i1.9517
Hal 305-315
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Agar implementasi keamanan dilakukan secara terstruktur, penelitian ini mengacu pada NIST Cybersecurity Framework dengan memetakan mekanisme IDS, WAF, dan FIM ke dalam fungsi Identify, Protect, Detect, dan Respond [10][11]. Pendekatan ini memungkinkan evaluasi keamanan tidak hanya dari sisi teknis, tetapi juga dalam kerangka manajemen risiko yang lebih sistematis.

Beberapa penelitian terdahulu telah membahas penerapan WAF untuk memitigasi serangan berbasis input pada aplikasi web, serta implementasi IDS untuk mendeteksi anomali lalu lintas jaringan. Terdapat pula studi yang memanfaatkan sistem monitoring log terpusat untuk analisis insiden keamanan. Namun, sebagian besar penelitian tersebut masih berfokus pada satu mekanisme proteksi atau belum mengintegrasikan pemantauan integritas berkas sebagai bagian dari strategi pertahanan menyeluruh. Pada konteks OJS, kajian yang ada umumnya berhenti pada tahap identifikasi kerentanan tanpa implementasi arsitektur keamanan terintegrasi yang diuji melalui skenario serangan secara langsung.

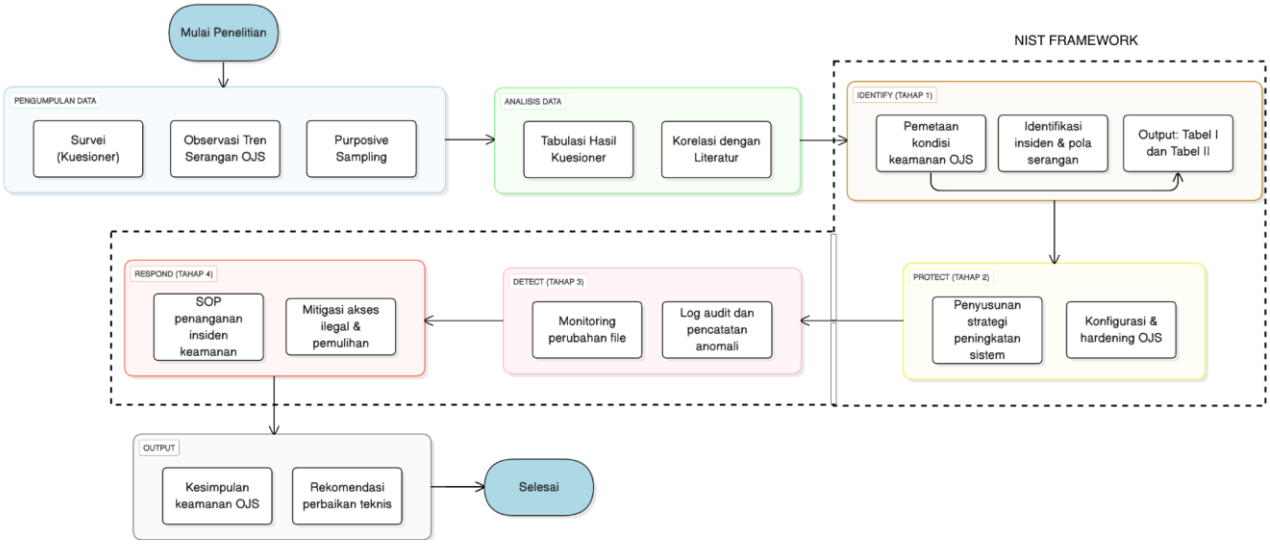
Berdasarkan kondisi tersebut, penelitian ini memfokuskan pada perancangan dan evaluasi arsitektur keamanan terintegrasi yang menggabungkan IDS, WAF, dan FIM dalam satu sistem terpusat, serta memetakan implementasinya secara eksplisit ke dalam kerangka NIST Cybersecurity Framework. Pendekatan ini diharapkan dapat memberikan gambaran yang lebih komprehensif mengenai efektivitas strategi defense-in-depth pada lingkungan OJS [12][13]. Melalui pendekatan tersebut, penelitian ini bertujuan untuk menganalisis sejauh mana integrasi IDS, WAF, dan FIM mampu meningkatkan kemampuan deteksi dan respons terhadap ancaman pada platform OJS, sehingga dapat menjadi referensi dalam upaya penguatan keamanan sistem publikasi ilmiah [14][16].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan melalui tahapan sistematis yang mengacu pada kerangka kerja NIST *Cybersecurity Framework*. Alur metodologi penelitian ditunjukkan pada Gambar 1, yang menggambarkan proses pengumpulan dan analisis data hingga penerapan tahapan *Identify*, *Protect*, *Detect*, dan *Respond* dalam evaluasi keamanan Open Journal Systems (OJS).

Untuk memberikan gambaran yang lebih sistematis mengenai tahapan penelitian yang dilakukan, alur metodologi penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur Metodologi Penelitian.

Gambar 1 menunjukkan alur metodologi penelitian yang dimulai dari tahap pengumpulan data melalui survei kuesioner kepada pengelola jurnal serta observasi terhadap tren serangan pada OJS. Pemilihan objek penelitian dilakukan menggunakan teknik purposive sampling untuk memastikan kesesuaian karakteristik sistem yang dianalisis.

Data yang diperoleh kemudian dianalisis melalui proses tabulasi hasil kuesioner dan dikorelasikan dengan literatur terkait guna mengidentifikasi pola ancaman yang dominan. Hasil analisis ini menjadi dasar dalam penerapan tahapan NIST Cybersecurity Framework.

Pada tahap Identify dilakukan pemetaan kondisi keamanan awal OJS dan identifikasi pola serangan yang sering terjadi. Tahap Protect mencakup penyusunan strategi peningkatan keamanan serta konfigurasi dan hardening sistem. Tahap Detect difokuskan pada monitoring perubahan berkas dan pencatatan log anomali sebagai bagian dari mekanisme deteksi. Selanjutnya, tahap Respond meliputi penyusunan SOP penanganan insiden dan mitigasi akses ilegal.

Seluruh tahapan tersebut menghasilkan kesimpulan tingkat keamanan OJS dan rekomendasi perbaikan teknis sebagai luaran penelitian.

2.2 Pengumpulan Data

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 1, Februari 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v13i1.9517
Hal 305-315
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Metodologi penelitian ini mengadopsi kerangka kerja NIST *Framework* yang dikombinasikan dengan pengumpulan data empiris, sebagaimana diilustrasikan pada Gambar 1. Tahap awal penelitian dimulai dengan pengumpulan data melalui dua pendekatan utama, yaitu survei menggunakan kuesioner kepada pengelola jurnal dan observasi terhadap tren serangan OJS [17][18].

Untuk menentukan subjek penelitian, teknik pengambilan sampel yang digunakan adalah *purposive sampling*. Teknik ini dipilih untuk menetapkan sampel berdasarkan kriteria spesifik, yaitu jurnal aktif yang teridentifikasi pernah mengalami insiden keamanan atau memiliki konfigurasi sistem yang tidak diperbarui. Berdasarkan teknik tersebut, data dikumpulkan dari unit pengelola jurnal di 7 universitas di Kalimantan, yang menghasilkan total 11 jurnal sebagai objek analisis [19].

Setelah proses pengumpulan data selesai, penelitian dilanjutkan ke tahap Analisis Data. Pada tahap ini, hasil kuesioner ditabulasi dan dikorelasikan dengan studi literatur untuk memvalidasi pola serangan dominan sebelum masuk ke tahapan inti NIST (Identify, Protect, Detect, Respond) [20][21][22].

2.2.1 Analisis Data

Analisis data merupakan tahapan lanjutan dari proses pengumpulan data. Pada tahap ini, data hasil kuesioner dan observasi diolah melalui tabulasi dan dikorelasikan dengan studi literatur untuk mengidentifikasi pola ancaman keamanan yang dominan sebagai dasar penerapan tahapan NIST *Framework* [23][24].

Analisis data dilakukan untuk mengolah dan menyaring informasi awal sebelum masuk ke tahapan inti NIST *Framework*. Data hasil kuesioner ditabulasi untuk memetakan kondisi versi OJS, praktik pemeliharaan, serta jenis insiden keamanan yang pernah dialami oleh masing-masing jurnal [10][25].

Selanjutnya, hasil tabulasi tersebut dikorelasikan dengan temuan pada studi literatur dan laporan insiden keamanan OJS di luar lingkungan penelitian. Proses ini bertujuan untuk mengidentifikasi kesesuaian antara kondisi lokal dengan pola ancaman yang telah banyak dilaporkan secara global [26].

Hasil dari tahap analisis data ini berupa pemetaan awal pola serangan dominan dan kelemahan umum pada OJS, yang kemudian digunakan sebagai dasar penentuan fokus pada tahapan *Identify*, *Protect*, *Detect*, dan *Respond* sebagaimana ditunjukkan pada Gambar 1 [26][10].

2.3 Penerapan NIST Framework

Penerapan NIST *Cybersecurity Framework* dalam penelitian ini digunakan sebagai kerangka sistematis untuk menganalisis dan mengelola keamanan website *Open Journal Systems* (OJS) berdasarkan data empiris yang diperoleh dari kuesioner dan observasi. Setiap fungsi utama NIST, yaitu *Identify*, *Protect*, *Detect*, dan *Respond*, diterapkan secara berurutan dan saling terintegrasi untuk memetakan kondisi keamanan OJS, mengidentifikasi kerentanan dominan, serta mengevaluasi kemampuan institusi dalam melakukan pencegahan, deteksi, dan penanganan insiden keamanan siber. Tahapan Identify menjadi fokus awal analisis untuk menggambarkan profil risiko dan pola insiden keamanan OJS sebelum masuk ke tahapan pengendalian berikutnya [27][28].

2.3.1 Identifikasi

Pada tahap identifikasi, kondisi keamanan 10 jurnal yang dikelola oleh enam unit pengelola di wilayah Kalimantan dianalisis berdasarkan dua sumber utama, yaitu hasil kuesioner dan observasi literatur. Kuesioner memberikan gambaran mengenai pola penggunaan, prosedur pemeliharaan, serta frekuensi terjadinya insiden yang dialami secara langsung oleh pengelola jurnal. Sementara itu, observasi literatur dilakukan dengan meninjau laporan insiden, studi terdokumentasi, dan kasus keamanan OJS di luar lingkungan penelitian untuk mengetahui pola serangan yang paling sering terjadi serta teknik eksploitasi yang digunakan [29][30][31].

Hasil dari kedua sumber informasi tersebut menunjukkan kecenderungan yang konsisten. Kuesioner mengungkapkan kurangnya pembaruan versi, ketiadaan monitoring rutin, serta terjadinya insiden seperti *defacement*, penyisipan *webshell*, dan *SQL Injection*. Temuan ini diperkuat oleh observasi literatur, dimana insiden serupa juga tercatat secara luas di platform OJS pada berbagai institusi lainnya. Kombinasi temuan lokal dan global ini memastikan bahwa kelemahan yang teridentifikasi bukan kasus tunggal, tetapi merupakan pola ancaman yang relevan terhadap OJS secara umum [32][33].

Dengan demikian, tahap identifikasi tidak hanya menjelaskan kondisi keamanan pada lingkungan penelitian, tetapi juga membandingkannya dengan tren insiden eksternal, sehingga memberikan dasar yang kuat untuk penentuan strategi proteksi, deteksi, dan respons yang lebih terarah.

Ringkasan hasil identifikasi kondisi keamanan berdasarkan versi OJS disajikan pada Tabel 1, sedangkan pengelompokan pola insiden dominan ditampilkan pada Tabel 2.

Tabel 1. Kondisi Keamanan Berdasarkan Versi OJS

Versi OJS	Dampak Insiden
3.5.0	Pembaruan tidak konsisten, beberapa fitur keamanan belum digunakan optimal
3.3.0 - 3.4.0	Insiden *webshell* dan *defacement* muncul berulang
3.1 - 3.2	Rentan terhadap *SQLI* dan *XSS*, tidak ada monitoring keamanan rutin

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 1, Februari 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v13i1.9517
Hal 305-315
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Tabel 2. Pola Insiden Keamanan Berdasarkan Versi OJS

Versi OJS	Insiden Dominan	Dampak Insiden
3.5.0	*Defacement*	Perubahan tampilan halaman, dan gangguan integritas konten
3.3.0 - 3.4.0	*Webshell*	*Remote* Akses *Server* Sehingga bisa melihat dan merubah direktori dari server
3.1 - 3.2	*SQLI, XSS*	Rentan terhadap *SQLI* dan *XSS*, tidak ada monitoring keamanan rutin

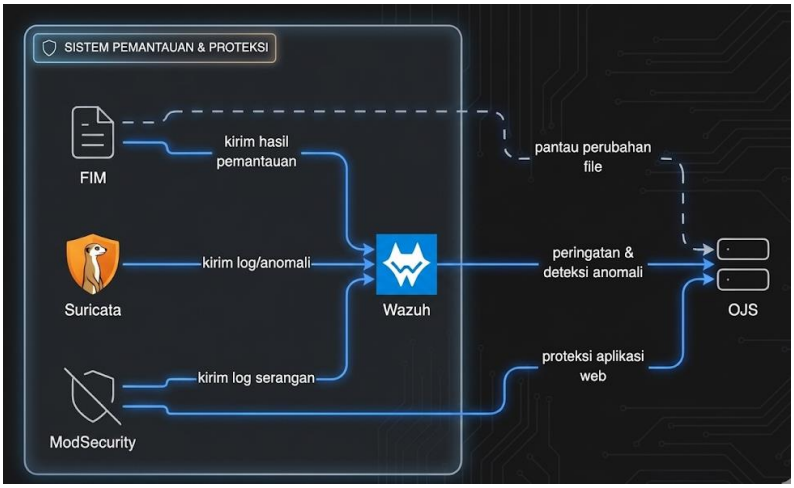
Hasil identifikasi menunjukkan bahwa versi Open Journal Systems (OJS) yang digunakan pada objek penelitian tidak seragam dan sebagian besar belum dikelola dengan pembaruan keamanan yang konsisten. Berdasarkan Tabel 1, OJS versi 3.5.0 masih menunjukkan pemanfaatan fitur keamanan yang belum optimal, sementara pada versi 3.3.0 hingga 3.4.0 ditemukan insiden webshell dan defacement yang terjadi secara berulang. Kondisi ini menunjukkan bahwa ketidakkonsistenan pembaruan versi dan konfigurasi keamanan meningkatkan peluang eksploitasi terhadap celah yang telah diketahui secara publik [32][35].

Temuan tersebut diperkuat oleh Tabel 2 yang menunjukkan bahwa jenis insiden dominan pada OJS versi menengah dan lama didominasi oleh defacement bermuatan iklan judi serta penyisipan webshell, yang memungkinkan penyerang memperoleh akses jarak jauh ke server dan melakukan modifikasi direktori sistem. Pada OJS versi 3.1 hingga 3.2, kerentanan yang muncul lebih banyak berkaitan dengan serangan SQL Injection dan Cross-Site Scripting (XSS), yang berdampak pada gangguan layanan, perubahan konten, serta potensi kebocoran data [34].

Selain pengelompokan berdasarkan versi OJS, hasil observasi langsung pada direktori sistem menunjukkan bahwa serangan umumnya diawali dengan keberadaan berkas asing yang masuk melalui celah unggah atau hasil eksploitasi injeksi. Temuan ini menegaskan bahwa kelemahan tidak hanya berada pada lapisan aplikasi, tetapi juga pada pengelolaan berkas di sisi server, sehingga pemantauan integritas berkas menjadi komponen krusial dalam proses identifikasi awal. Secara keseluruhan, tahap identifikasi ini tidak hanya memetakan kondisi teknis keamanan OJS, tetapi juga mengungkap pola risiko sistemik yang menjadi dasar penerapan mekanisme proteksi, deteksi, dan respons pada tahapan NIST berikutnya [35][10].

2.3.2 Proteksi

Pada tahap Protect, sistem keamanan OJS dirancang menggunakan arsitektur pemantauan terintegrasi yang melibatkan FIM, Suricata, ModSecurity, dan Wazuh. Integrasi ini membentuk mekanisme perlindungan berlapis yang menggabungkan pemantauan integritas file, deteksi lalu lintas jaringan, serta penyaringan permintaan aplikasi web. Topologi arsitektur proteksi tersebut ditunjukkan pada Gambar 2.



Gambar 2. Topologi Proteksi.

Sebagaimana ditunjukkan pada Gambar 2, arsitektur proteksi dirancang dengan Wazuh sebagai pusat pengolahan dan korelasi log dari berbagai komponen keamanan. FIM melakukan pemantauan terhadap perubahan file dan direktori penting pada OJS, kemudian mengirimkan hasilnya ke Wazuh untuk dianalisis lebih lanjut. Mekanisme ini memungkinkan setiap perubahan yang tidak sesuai dengan baseline sistem dapat segera terdeteksi [32][10].

Pada sisi jaringan, Suricata memantau lalu lintas masuk dan keluar untuk mengidentifikasi pola serangan berdasarkan signature maupun anomali trafik. Seluruh event yang terdeteksi diteruskan ke Wazuh sehingga dapat dikorelasikan dengan data dari FIM maupun sumber lainnya. Dengan pendekatan ini, indikasi serangan tidak dinilai secara terpisah, tetapi dianalisis sebagai bagian dari pola aktivitas yang lebih luas [12][36].

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 1, Februari 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v13i1.9517
Hal 305-315
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Sementara itu, ModSecurity berfungsi sebagai lapisan proteksi aplikasi yang menyaring permintaan HTTP sebelum diproses oleh OJS. Permintaan yang terindikasi berbahaya akan diblokir, dan catatan serangannya dikirimkan ke Wazuh sebagai bagian dari proses monitoring terpusat. Kombinasi antara proteksi langsung di sisi aplikasi dan pemantauan terintegrasi memberikan lapisan pertahanan yang lebih komprehensif [32][37].

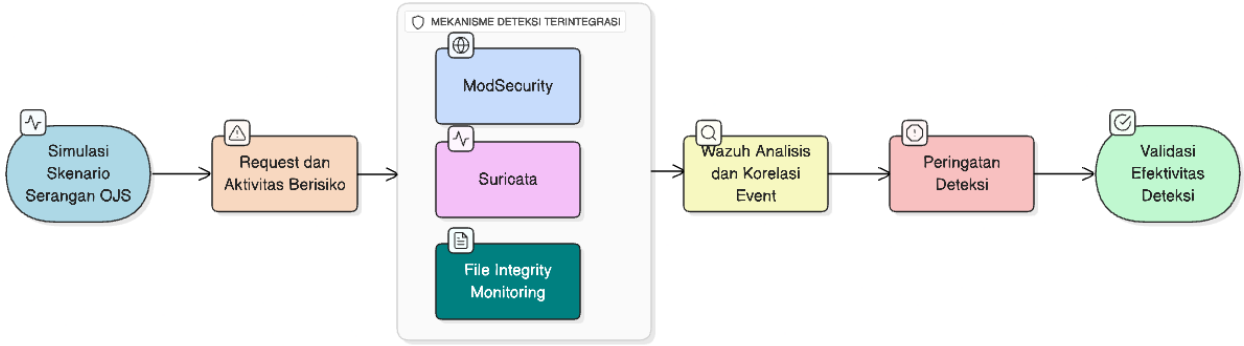
Secara keseluruhan, integrasi FIM, Suricata, dan ModSecurity melalui Wazuh tidak hanya memperkuat kemampuan pencegahan, tetapi juga meningkatkan ketepatan deteksi melalui korelasi multi-sumber. Pendekatan ini mendukung kesiapan sistem dalam mengidentifikasi dan merespons insiden keamanan pada OJS secara lebih terstruktur.

2.3.3 Deteksi

Tahap deteksi difokuskan pada dua pola serangan yang paling sering muncul pada OJS, yaitu penyisipan webshell melalui mekanisme unggah berkas dan percobaan SQL Injection pada parameter aplikasi. Pola ini sejalan dengan hasil identifikasi sebelumnya, di mana insiden defacement dan keberadaan berkas asing umumnya berawal dari keberhasilan unggah file berbahaya [32][38].

Untuk mendukung proses tersebut, ModSecurity dikonfigurasi untuk menganalisis permintaan HTTP, terutama pada proses unggah dan parameter input. FIM memantau perubahan pada direktori penting seperti public/ dan files/, sedangkan Suricata mengawasi lalu lintas jaringan untuk mengidentifikasi indikasi eksploitasi atau pemindaian otomatis. Seluruh event dikirimkan ke Wazuh untuk dikorelasikan dan menghasilkan peringatan terpusat.

Alur mekanisme deteksi yang digunakan dalam pengujian ditunjukkan pada Gambar 3.



Gambar 3. Alur mekanisme deteksi serangan OJS berbasis integrasi ModSecurity, Suricata, FIM, dan Wazuh.

Gambar 3 memperlihatkan alur deteksi yang digunakan dalam pengujian penetrasi terkontrol. Proses dimulai dari simulasi skenario serangan, seperti unggah webshell dan percobaan SQL Injection. Setiap aktivitas yang masuk dianalisis secara paralel oleh ModSecurity, Suricata, dan FIM sesuai dengan perannya masing-masing.

ModSecurity memeriksa permintaan HTTP yang mencurigakan, Suricata mengamati pola lalu lintas jaringan, sedangkan FIM mendeteksi perubahan berkas pada direktori OJS. Event yang dihasilkan kemudian dikirimkan ke Wazuh untuk dikorelasikan sehingga peringatan yang muncul tidak berdiri sendiri, tetapi berdasarkan keterkaitan antar aktivitas.

Peringatan tersebut selanjutnya digunakan untuk mengevaluasi apakah mekanisme deteksi mampu mengidentifikasi skenario serangan yang diuji.

2.3.4 Respon

Pada tahap respon, sistem keamanan melakukan tindakan lanjutan setelah peringatan serangan terdeteksi. Ketika *Suricata* menemukan pola lalu lintas jaringan yang mencurigakan, sistem segera mencatat kejadian tersebut sebagai indikasi awal serangan. Pada saat yang sama, *ModSecurity* menghentikan permintaan berbahaya pada sisi aplikasi web agar tidak berdampak langsung pada layanan OJS [46][47].

Jika terjadi perubahan tidak sah pada berkas inti OJS, *File Integrity Monitoring* (FIM) akan memicu notifikasi yang menandai adanya aktivitas mencurigakan pada sistem. Seluruh informasi dari ketiga komponen tersebut kemudian dikumpulkan oleh Wazuh untuk dianalisis secara terpusat. Berdasarkan hasil analisis ini, administrator dapat melakukan langkah penanganan, seperti mengisolasi sumber serangan, memulihkan berkas dari cadangan, atau memperketat aturan keamanan. Dengan pendekatan ini, tahap respon tidak hanya berfungsi sebagai reaksi terhadap insiden, tetapi juga sebagai upaya untuk menekan dampak serangan agar tidak berkembang lebih jauh [41][48][49].

2.3.5 Ringkasan Metodologi dan Keluaran

Tahapan identifikasi hingga respon pada kerangka NIST menghasilkan keluaran berupa rekomendasi keamanan dan evaluasi efektivitas sistem pemantauan yang diterapkan. Keluaran tersebut diperoleh dari hasil analisis setiap tahapan, mulai dari identifikasi kerentanan hingga mekanisme respon terhadap insiden. Seluruh hasil ini selanjutnya digunakan sebagai dasar penyusunan analisis pada bagian hasil dan pembahasan.

Pemilihan integrasi Intrusion Detection System (IDS), Web Application Firewall (WAF), dan File Integrity Monitoring (FIM) dalam penelitian ini didasarkan pada pendekatan defense-in-depth yang memungkinkan perlindungan pada berbagai lapisan sistem, meliputi jaringan, aplikasi, dan berkas server. Pendekatan ini dipilih karena serangan

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 1, Februari 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v13i1.9517
Hal 305-315
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

terhadap Open Journal Systems (OJS) tidak hanya terjadi pada satu lapisan, tetapi dapat melibatkan eksploitasi aplikasi web yang diikuti dengan perubahan berkas sistem. Oleh karena itu, penerapan mekanisme tunggal seperti IDS atau WAF dinilai belum memadai untuk memberikan perlindungan yang komprehensif. Integrasi ketiga mekanisme tersebut diharapkan mampu meningkatkan kemampuan deteksi dan respons terhadap ancaman secara lebih menyeluruh.

Tabel 3. Pemetaan NIST CSF terhadap Mekanisme Keamanan

Fungsi NIST	Implementasi Pada Penelitian	Komponen
Identy	Identifikasi Aset Dan Kerentanan OJS	Analisis Server Dan Plugin
Protect	Pemfilteran Request Berbahaya	WAF
Detect	Deteksi Anomali Dan Perubahan File	IDS Dan FIM
Respond	Korelasi Log Dan	Wazuh

Pemetaan tersebut menunjukkan bahwa setiap mekanisme keamanan berkontribusi pada tahapan manajemen keamanan secara sistematis sesuai dengan kerangka NIST.

3. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil penerapan metode keamanan berbasis NIST Cybersecurity Framework pada lingkungan Open Journal Systems (OJS) yang menjadi objek penelitian. Implementasi dilakukan secara bertahap mengikuti alur metode yang telah dirancang pada bab sebelumnya, sehingga setiap tahapan memiliki tujuan dan keluaran yang jelas.

Tahap pertama adalah identifikasi aset dan pola ancaman dominan berdasarkan hasil observasi dan analisis insiden. Tahap ini menghasilkan pemetaan risiko yang menjadi dasar dalam menentukan mekanisme perlindungan yang akan diterapkan. Selanjutnya, pada tahap Protect, dilakukan konfigurasi dan integrasi ModSecurity sebagai Web Application Firewall, Suricata sebagai sistem deteksi lalu lintas jaringan, serta File Integrity Monitoring (FIM) untuk memantau perubahan berkas pada direktori penting OJS. Seluruh komponen tersebut diintegrasikan ke dalam Wazuh sebagai pusat pengumpulan log dan korelasi kejadian keamanan.

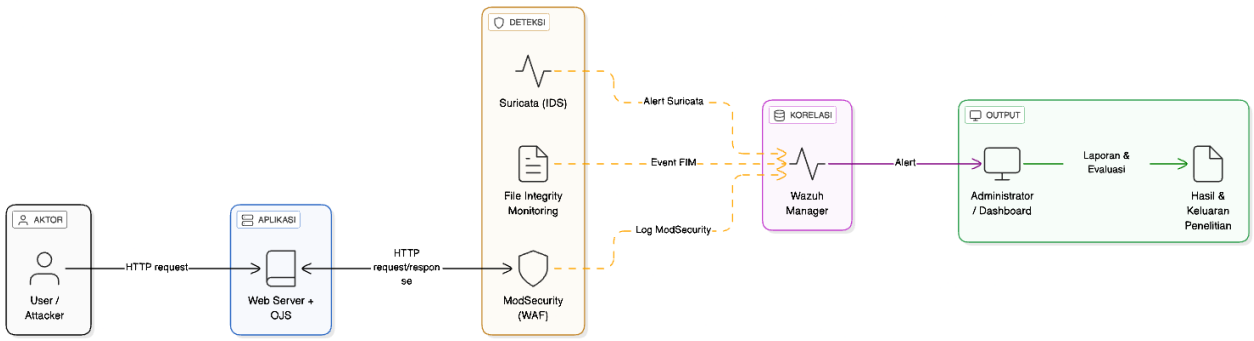
Tahap berikutnya adalah implementasi mekanisme deteksi dengan menyusun aturan (rule) dan skenario pengujian yang disesuaikan dengan pola serangan yang telah diidentifikasi, seperti unggah webshell dan percobaan SQL Injection. Proses ini dilanjutkan dengan pengujian penetrasi terkontrol untuk mengevaluasi kemampuan sistem dalam menghasilkan peringatan deteksi terhadap aktivitas yang disimulasikan.

Hasil dari setiap tahapan tidak hanya dideskripsikan secara teknis, tetapi juga dianalisis untuk melihat sejauh mana metode yang diterapkan mampu mengurangi risiko dan meningkatkan visibilitas keamanan pada OJS. Dengan demikian, pembahasan pada bab ini berfokus pada hubungan antara tahapan metode, implementasi teknis, serta keluaran yang diperoleh dari proses pengujian yang dilakukan.

3.1 Gambaran Hasil Implementasi Sistem Keamanan

Bagian ini memaparkan hasil integrasi sistem keamanan yang telah dirancang pada infrastruktur server OJS. Implementasi dilakukan dengan menyatukan beberapa lapisan pertahanan (*defense in depth*) yang terpusat pada platform Wazuh sebagai pengumpul log dan korelasi kejadian keamanan. Arsitektur sistem yang diimplementasikan dapat dilihat pada Gambar 4.

Untuk memperjelas bagaimana mekanisme keamanan diterapkan pada server OJS, Gambar 4 menampilkan rancangan arsitektur sistem yang mengintegrasikan IDS, WAF, FIM, dan Wazuh dalam satu kesatuan sistem.



Gambar 4. Arsitektur Implementasi Sistem Keamanan OJS Terintegrasi dengan Wazuh.

Dari arsitektur tersebut terlihat bahwa setiap komponen keamanan ditempatkan pada lapisan yang berbeda sesuai dengan fungsinya. ModSecurity berperan pada sisi aplikasi untuk menyaring permintaan HTTP sebelum diproses oleh OJS, sedangkan Suricata memantau lalu lintas jaringan guna mendeteksi pola serangan yang tidak tertangani pada tingkat aplikasi. FIM berfungsi sebagai pengawas integritas berkas untuk memastikan tidak terjadi perubahan tidak sah pada direktori sistem.

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 1, Februari 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v13i1.9517
Hal 305-315
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Integrasi ketiga komponen tersebut melalui Wazuh memungkinkan seluruh peristiwa keamanan dianalisis secara terpusat. Hal ini penting karena serangan terhadap aplikasi web sering kali tidak terjadi dalam satu tahap saja, melainkan melibatkan beberapa lapisan sistem. Dengan pendekatan ini, potensi celah pada satu mekanisme dapat dikompensasi oleh mekanisme lainnya, sehingga pengawasan keamanan menjadi lebih menyeluruh dan tidak bersifat parsial.

3.2 Skenario Pengujian Keamanan Berbasis OWASP Top 10

Pengujian keamanan dilakukan untuk mengevaluasi kemampuan arsitektur sistem yang telah dirancang dalam menghadapi ancaman nyata pada platform Open Journal Systems (OJS). Skenario pengujian disusun dengan mengacu pada OWASP Top 10 karena kerangka tersebut merepresentasikan jenis kerentanan yang paling umum ditemukan pada aplikasi web. Pemilihan acuan ini juga mempertimbangkan karakteristik OJS yang memproses data pengguna dan naskah ilmiah melalui mekanisme input berbasis web.

Skenario yang diuji tidak hanya mencakup eksploitasi pada sisi aplikasi, seperti manipulasi input dan penyisipan skrip berbahaya, tetapi juga mencakup potensi gangguan terhadap integritas sistem melalui modifikasi berkas secara tidak sah. Pendekatan ini bertujuan untuk menguji efektivitas integrasi IDS, WAF, dan FIM pada berbagai lapisan keamanan. Pemetaan antara kategori ancaman OWASP Top 10 dan skenario serangan yang diterapkan pada OJS ditunjukkan pada Tabel 4.

Tabel 4. Pemetaan Skenario Serangan OWASP Top 10 pada Platform OJS

OWASP ID	Kategori Serangan	Skenario Serangan	Relevansi pada Sistem OJS
A03	Injection	Percobaan SQL Injection pada parameter URL dan proses autentikasi	Berpotensi mengubah logika query dan mengakses data tanpa izin
A07	Cross-Site Scripting (XSS)	Penyisipan skrip berbahaya melalui form pengiriman naskah	Memungkinkan eksekusi skrip pada browser pengguna
A08	Software & Data Integrity Failures	Penyisipan webshell atau berkas berbahaya	Mengancam integritas sistem dan kendali server

Jika dilihat dari pemetaan pada Tabel 4, skenario yang diuji tidak hanya menargetkan kelemahan pada input aplikasi, tetapi juga menyentuh aspek integritas sistem secara keseluruhan. Serangan Injection dan XSS berfokus pada bagaimana aplikasi memproses permintaan pengguna, sedangkan penyisipan webshell berkaitan dengan kemungkinan perubahan berkas dan penguasaan server. Variasi ini dipilih agar pengujian tidak terbatas pada satu jenis ancaman saja, melainkan dapat menggambarkan kondisi serangan yang lebih realistis. Melalui pendekatan tersebut, kemampuan integrasi IDS, WAF, dan FIM dapat diamati secara lebih menyeluruh pada berbagai lapisan keamanan.

3.3 Evaluasi Efektivitas Sistem Keamanan Terintegrasi

Setelah seluruh skenario pengujian dijalankan, dilakukan evaluasi untuk menilai sejauh mana sistem keamanan terintegrasi mampu mendeteksi dan merespons setiap percobaan serangan. Evaluasi ini bertujuan untuk melihat kinerja kombinasi *Web Application Firewall* (WAF), *Intrusion Detection System* (IDS), dan *File Integrity Monitoring* (FIM) dalam satu mekanisme pemantauan yang terpusat melalui Wazuh.

Pengukuran efektivitas sistem dilakukan menggunakan indikator Detection Rate, yaitu rasio antara jumlah serangan yang berhasil dikenali oleh sistem keamanan terhadap total skenario pengujian yang dilakukan. Suatu serangan dianggap berhasil terdeteksi apabila sistem memberikan respons yang dapat diamati, seperti pemblokiran akses, pencatatan aktivitas mencurigakan pada log keamanan, atau munculnya peringatan pada dashboard pemantauan. Pendekatan ini dipilih karena mampu menggambarkan kinerja sistem secara praktis tanpa bergantung pada asumsi teoritis semata.

Penyajian hasil evaluasi dalam bentuk matriks bertujuan untuk memberikan gambaran yang komparatif mengenai kinerja masing-masing komponen keamanan terhadap jenis serangan yang diuji. Dengan demikian, efektivitas integrasi WAF, IDS, dan FIM dapat dianalisis tidak hanya dari sisi jumlah deteksi, tetapi juga dari pola respons sistem pada berbagai skenario serangan. Ringkasan hasil evaluasi efektivitas sistem keamanan terintegrasi berdasarkan skenario pengujian yang dilakukan ditampilkan pada Tabel 5.

Tabel 5. Matriks Evaluasi Keamanan Terintegrasi

Skenario Serangan	Komponen Keamanan	Jumlah Uji	Deteksi Berhasil	Tingkat Deteksi	Respon Sistem
SQL Injection	ModSecurity dan Suricata	10	10	100%	Permintaan diblokir & Deteksi Berhasil
Cross-Site Scripting (XSS)	ModSecurity	10	8	80%	Permintaan diblokir

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 1, Februari 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v13i1.9517
Hal 305-315
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Webshell / Backdoor	Wazuh FIM	10	10	100%	Peringatan Real-Time

Hasil pada Tabel 5 menunjukkan bahwa sistem keamanan terintegrasi mampu mendeteksi sebagian besar skenario serangan yang diuji. Pada percobaan SQL Injection dan penyisipan webshell, seluruh upaya serangan berhasil dikenali oleh sistem. Hal ini mengindikasikan bahwa mekanisme penyaringan pada lapisan aplikasi serta pemantauan perubahan berkas berjalan sesuai dengan konfigurasi yang diterapkan.

Berbeda dengan dua skenario tersebut, tingkat deteksi pada pengujian Cross-Site Scripting (XSS) tidak mencapai seluruh percobaan. Temuan ini menunjukkan bahwa efektivitas deteksi sangat dipengaruhi oleh variasi payload dan aturan yang aktif pada WAF. Dengan demikian, meskipun integrasi sistem telah mampu memberikan perlindungan yang cukup kuat, penyempurnaan konfigurasi tetap diperlukan agar cakupan deteksi dapat ditingkatkan.

3.4 Pembahasan Hasil Pengujian Keamanan Sistem

Hasil pengujian yang disajikan pada Tabel 5 menunjukkan bahwa integrasi sistem keamanan berbasis WAF, IDS, dan FIM mampu memberikan tingkat deteksi yang tinggi terhadap sebagian besar skenario serangan yang diuji. Serangan *SQL Injection* dan penyisipan *webshell* berhasil terdeteksi secara konsisten pada seluruh percobaan pengujian, yang mengindikasikan bahwa mekanisme perlindungan pada lapisan aplikasi dan pemantauan integritas berkas telah berjalan dengan efektif.

Deteksi penuh terhadap serangan *SQL Injection* menunjukkan bahwa penerapan *ModSecurity* dengan aturan berbasis OWASP *Core Rule Set* mampu mengidentifikasi pola serangan yang menargetkan parameter aplikasi OJS. Selain itu, dukungan Suricata pada lapisan jaringan memperkuat proses deteksi dengan mengenali aktivitas anomali yang berkaitan dengan eksploitasi layanan. Kombinasi kedua komponen ini memberikan perlindungan berlapis yang mencegah serangan berhasil mencapai sistem inti.

Pada pengujian penyisipan webshell atau backdoor, mekanisme *File Integrity Monitoring* (FIM) pada Wazuh menunjukkan kinerja yang stabil dengan mendeteksi seluruh perubahan berkas yang tidak sah pada direktori sistem. Peringatan yang dihasilkan secara real-time memungkinkan administrator untuk segera mengetahui adanya upaya modifikasi berkas, sehingga risiko pengambilalihan sistem dapat diminimalkan sejak tahap awal.

Namun demikian, hasil pengujian terhadap serangan Cross-Site Scripting (XSS) menunjukkan tingkat deteksi yang lebih rendah dibandingkan skenario lainnya, yaitu sebesar 80%. Kondisi ini menunjukkan bahwa masih terdapat variasi payload XSS yang belum sepenuhnya terakomodasi oleh aturan generik pada ModSecurity. Pembatasan tersebut diterapkan untuk menghindari terjadinya false-positive yang dapat mengganggu proses bisnis OJS, terutama pada fitur yang melibatkan input pengguna. Dalam beberapa kasus, mekanisme internal OJS dengan pola tertentu memiliki kemiripan dengan signature XSS generik sehingga sempat teridentifikasi sebagai aktivitas mencurigakan oleh WAF.

Secara keseluruhan, hasil pengujian memperlihatkan bahwa tidak ada satu komponen keamanan yang bekerja secara mandiri. WAF berperan sebagai garis pertahanan awal pada lapisan aplikasi, IDS melakukan pemantauan lalu lintas jaringan untuk mendeteksi pola serangan, sedangkan FIM memastikan integritas berkas sistem tetap terjaga. Seluruh informasi keamanan tersebut dikorelasikan oleh Wazuh sehingga administrator memperoleh gambaran menyeluruh mengenai kondisi keamanan OJS dalam satu platform pemantauan terpusat.

Hasil evaluasi menunjukkan bahwa integrasi sistem keamanan berbasis Wazuh memberikan perlindungan yang komprehensif, namun terdapat beberapa catatan teknis terkait penggunaan aturan keamanan generik. Sebagai bukti temuan di lapangan, penggunaan *OWASP Core Rule Set* (CRS) pada *ModSecurity* berpotensi menimbulkan *false-positive* yang signifikan; misalnya saat penulis jurnal mencoba mengunggah naskah yang mengandung sitasi berupa potongan kode pemrograman (*source code*), sistem secara otomatis mengidentifikasinya sebagai upaya *Injection* dan memblokir proses *submission* tersebut. Meskipun hasil pengujian secara umum menunjukkan kinerja yang sangat baik dengan tingkat deteksi yang tinggi, penelitian ini masih memiliki keterbatasan pada penggunaan aturan yang belum teroptimasi sepenuhnya untuk karakteristik trafik spesifik OJS. Oleh karena itu, pengembangan lebih lanjut sangat disarankan untuk melakukan penyesuaian aturan (*fine-tuning*) guna meminimalkan hambatan pada proses operasional jurnal tanpa mengurangi integritas keamanan sistem.

Meskipun integrasi sistem menunjukkan tingkat deteksi yang tinggi, evaluasi belum lengkap tanpa menilai dampaknya terhadap aktivitas pengguna yang sah. Oleh karena itu, dilakukan pengujian tambahan untuk mengidentifikasi potensi false-positive yang muncul selama operasional normal OJS. Pengujian ini bertujuan untuk memastikan bahwa mekanisme keamanan tidak menimbulkan hambatan yang signifikan terhadap proses submission maupun interaksi internal sistem. Ringkasan hasil validasi tersebut disajikan pada Tabel 6.

Tabel 6. Validasi Akurasi Deteksi dan Temuan *False-Positive*

Skenario Aktivitas Sah	Tujuan Pengujian	Respon Sistem Keamanan (WAF)	Klasifikasi Hasil

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 1, Februari 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v13i1.9517
Hal 305-315
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Submission naskah teks biasa (.docx)	Menguji fungsi normal OJS	Berhasil (200 OK)	True Negative
Submission naskah berisi skrip PHP/XSS	Menguji ketelitian *rule* generik	Berhasil (200 OK)	True Negative
Inject Payload XSS di form input	Menguji fungsi proteksi	Diblokir (403 Forbidden)	True Positive
Manipulasi format skrips webshell dari php ke pdf	Menguji ketelitian *rule* generik	Berhasil (200 OK)	True Negative
Eksploitasi SQL Injection menggunakan SQLMap	Menguji fungsi proteksi	Diblokir (403 Forbidden)	True Positive
Pemanggilan internal OJS menggunakan `$$call$$`	Menguji kompatibilitas rule dengan mekanisme OJS	Diblokir (403 Forbidden)	False Positive

Data pada Tabel 6 memperlihatkan bahwa sistem tidak hanya efektif dalam mendeteksi serangan, tetapi juga cukup selektif dalam membedakan aktivitas normal dan aktivitas berbahaya. Mayoritas skenario operasional berjalan tanpa gangguan, yang menunjukkan bahwa penerapan mekanisme keamanan tidak secara signifikan menghambat proses bisnis OJS.

Meskipun demikian, kemunculan satu kasus false-positive pada pemanggilan internal OJS menunjukkan bahwa aturan generik masih memiliki keterbatasan dalam mengenali pola komunikasi internal aplikasi. Temuan ini menjadi indikator bahwa efektivitas sistem tidak hanya ditentukan oleh tingkat deteksi serangan, tetapi juga oleh ketepatan klasifikasi aktivitas yang sah. Oleh karena itu, proses penyesuaian aturan menjadi faktor penting dalam menjaga keseimbangan antara keamanan dan kenyamanan operasional.

Jika dibandingkan dengan sejumlah penelitian terdahulu yang lebih banyak menitikberatkan pada penerapan satu mekanisme keamanan secara terpisah, pendekatan dalam penelitian ini menawarkan cakupan pengawasan yang lebih menyeluruh. Integrasi WAF, IDS, dan FIM memungkinkan deteksi pada lapisan aplikasi, jaringan, serta integritas berkas dilakukan secara bersamaan dalam satu sistem pemantauan. Dengan demikian, potensi serangan yang lolos pada satu lapisan masih dapat teridentifikasi melalui mekanisme lainnya. Pendekatan ini memperlihatkan nilai tambah pada aspek korelasi dan visibilitas keamanan yang belum banyak dibahas pada implementasi tunggal.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan arsitektur keamanan terintegrasi yang menggabungkan Web Application Firewall, Intrusion Detection System, dan File Integrity Monitoring mampu meningkatkan kemampuan deteksi ancaman pada platform OJS secara terukur. Berdasarkan skenario pengujian berbasis OWASP Top 10, sistem mencapai tingkat deteksi 100% pada serangan SQL Injection dan penyisipan webshell, serta 80% pada skenario Cross-Site Scripting, yang menunjukkan bahwa pendekatan berlapis efektif dalam mengurangi risiko eksploitasi pada berbagai lapisan sistem. Integrasi melalui Wazuh memungkinkan korelasi peristiwa keamanan dari sisi aplikasi, jaringan, dan integritas berkas dilakukan dalam satu mekanisme pemantauan terpusat, sehingga administrator memperoleh gambaran kondisi keamanan secara menyeluruh. Meskipun sebagian besar aktivitas pengguna yang sah tetap dapat diproses tanpa gangguan, masih ditemukan potensi false-positive akibat penggunaan aturan keamanan generik yang belum sepenuhnya disesuaikan dengan karakteristik trafik OJS. Oleh karena itu, penelitian ini tidak hanya menegaskan efektivitas pendekatan keamanan berlapis, tetapi juga menunjukkan pentingnya proses penyesuaian aturan sebagai bagian dari pengelolaan keamanan yang berkelanjutan.

REFERENCES

[1] H. Wintolo, I. Riadi, and A. Yudhana, "Post Attack Mitigation on Open Journal System Services using Knowledge Understanding Assessment Defense (KUAD) Method," KINETIK, vol. 10, no. 4, pp. 455-464, Nov. 2025.
[2] M. R. Syaifudin, M. A. Murtadho, M. S. Wafa, and M. Masrur, "Analisis Keamanan Website Kampus UNIPDU Melalui Metode Vulnerability Assessment (VA) dengan Menggunakan Tools Acunetix," KOMPUTA: Jurnal Ilmiah Komputer dan Informatika, vol. 14, no. 1, pp. 21-32, Apr. 2025.
[3] S. E. Prasetyo, Haeruddin, and K. Ariesryo, "SISTEM KEamanan WEBSITE DARI SERANGAN DENIAL OF SERVICE, SQL INJECTION, CROSS SITE SCRIPTING MENGGUNAKAN WEB APPLICATION FIREWALL," ANTIVIRUS: Jurnal Ilmiah Teknik Informatika, vol. 18, no. 1, pp. 27-36, May 2024.
[4] S. A. Noswantoro, M. Ziaurrahman, Miftahurrizqi, M. A. Haq, and R. A. Rashid, "Implementasi Wazuh-ELK-Suricata untuk Deteksi Privilege Escalation di Ubuntu Server," Jurnal SAINTEKOM, vol. 15, no. 2, pp. 153-164, 2025.
[5] R. Aditya, Y. Muhyidin, and D. Singasatia, "Implementasi Security Information And Event Management (SIEM) Untuk Monitoring Keamanan Server Menggunakan Wazuh," Merkurius: Jurnal Riset Sistem Informasi dan Teknik Informatika, vol. 2, no. 5, pp. 137-144, Sep. 2024.

[6] A. Kamil, D. Rizaludin, and A. T. Ni'mah, "Implementasi Wazuh FIM (File Integrity Monitoring) untuk Perlindungan Keamanan Sistem Informasi," Sains Data: Jurnal Studi Matematika dan Teknologi, vol. 2, no. 2, pp. 80-92, 2024.
[7] A. Z. A. Adrian, R. A. Megantara, and F. A. Zami, "Hybrid Multilayer Architecture Integrating Suricata, Wazuh, and Cyber Threat Intelligence for Drive-by-Download Malvertising Detection," Sinkron: Jurnal dan Penelitian Teknik Informatika, vol. 10, no. 1, Jan. 2026.
[8] A. Shafiyyah, G. F. Nama, and R. A. Pradipta, "IMPLEMENTASI WAZUH MENGGUNAKAN METODE PPDIOO DI SISTEM KEAMANAN JARINGAN PSDKU UNIVERSITAS LAMPUNG WAYKANAN SEBAGAI DETEKSI DAN RESPON SERANGAN SIBER," JITET (Jurnal Informatika dan Teknik Elektro Terapan), vol. 12, no. 2, pp. 973-981, Apr. 2024.
[9] C. Kurniawan and A. Triayudi, "File Integrity Monitoring as a Method for Detecting and Preventing Web Defacement Attacks," *JOIN (Jurnal Online Informatika)*, vol. 9, no. 2, pp. 276-285, Dec. 2024.
[10] R. E. Reyes-Acosta et al., "Cybersecurity Conceptual Framework Applied to Edge Computing and Internet of Things Environments," Electronics, vol. 14, no. 11, p. 2109, May 2025.
[11] E. Handoyo and I. E. Nigrum, "Penilaian risiko keamanan siber kampus menggunakan framework cybersecurity NIST 1.1," CoSciTech Jurnal Computer Science and Information Technology, vol. 4, no. 3, pp. 677-685, Dec. 2023.
[12] A. Zulfa, I. Riadi, and A. Fadlil, "Hybrid Multilayer Architecture Integrating Suricata, Wazuh, and Cyber Threat Intelligence for Drive-by-Download Malvertising Detection," International Journal of Information Security, vol. 25, no. 2, pp. 445-458, Feb. 2026.
[13] S. Sulaiman and H. Hartono, "Analisis Forensik Terhadap Serangan Berbasis Injeksi pada Platform Open Journal System (OJS)," JURTI (Jurnal Rekayasa Teknologi Informasi), vol. 7, no. 1, pp. 45-53, Jun. 2023.
[14] J. Kim and S. Lee, "Evaluation of File Integrity Monitoring Systems in Cloud Environments: A Comparative Study using Wazuh and OSSEC," IEEE Transactions on Cloud Computing, vol. 13, no. 4, pp. 2112-2125, 2025.
[15] H. Wintolo, I. Riadi, and A. Yudhana, "Analisis Deteksi Penyusup pada Layanan Open Journal System Menggunakan Metode Network Forensic Development Life Cycle," SKANIKA, vol. 8, no. 1, pp. 133-144, Jan. 2025.
[16] R. S. Prasetyo and H. Suyono, "Implementasi dan Pengujian Keamanan Sistem Informasi menggunakan SIEM Wazuh: Studi Kasus Aplikasi Web Publik," Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), vol. 10, no. 5, pp. 987-996, Oct. 2023.
[17] Wahdana and K. H. Hanif, "IMPLEMENTASI KEAMANAN INFORMASI MENGGUNAKAN METODE WEB APPLICATION FIREWALL TERHADAP SQL INJECTION," JIP (Jurnal Informatika Polinema), vol. 11, no. 1, 2024.
[18] M. Fadya and D. N. Utama, "Towards Secure Information Systems: Developing and Implementing an Information Security Evaluation Model Using NIST CSF and COBIT 2019," TEM Journal, vol. 14, no. 1, pp. 182-191, Feb. 2025.
[19] J. L. Salas-Riega, Y. Riega-Virú, M. Ninaquispe-Soto, and J. M. Salas-Riega, "Cybersecurity and the NIST Framework: A Systematic Review of its Implementation and Effectiveness Against Cyber Threats," IJACSA, vol. 16, no. 6, 2025.
[20] S. J. Stratton, "Purposeful Sampling: Advantages and Pitfalls," Prehospital and Disaster Medicine, vol. 39, no. 2, pp. 121-122, 2024.
[21] A. A. Bahashwan et al., "A Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-Defined Networking," Sensors, vol. 23, no. 9, p. 4441, May 2023.
[22] M. R. Rahman et al., "Mining temporal attack patterns from cyberthreat intelligence reports," Knowledge and Information Systems, vol. 67, pp. 8941-8981, 2025.
[23] O. Aljumaiah, W. Jiang, S. R. Addula, and M. A. Almaiah, "Analyzing Cybersecurity Risks and Threats in IT Infrastructure based on NIST Framework," Journal of Cyber Security and Risk Auditing, vol. 2025, no. 2, pp. 26-40, Apr. 2025.
[24] J. M. Krishna and B. S. M. Yadav, "Cyber Threat Data Collection and Threat Analysis: Building a Foundation for Proactive Cybersecurity," IJETMS, vol. 9, no. 1, Mar.-Apr. 2025.
[25] V. Grigas, A. Gudinavičius, T. Petreikis, and A. Šuminas, "An Exploratory Study into Professional Scholarly Journals Publishing Software Adoption in Lithuania," Information & Media, vol. 96, pp. 179-201, 2023.
[26] C. R. Prabowo, D. Irmanto, and E. Rohadi, "STUDY AND ANALYSIS OF DOCKER INTERNAL SYSTEM SECURITY FROM DOCKER DAEMON ATTACK AND DDOS ON THE OPEN JOURNAL SYSTEM," JPUA, vol. 14, no. 1, pp. 61-68, 2024.
[27] C. E. Alozie, "Cloud Computing Baseline Security Requirements Within Enterprise Risk Management Framework," IJRPR, vol. 6, no. 6, pp. 426-428, June 2025.
[28] K. Svandova and Z. Smutny, "Internet of Medical Things Security Frameworks for Risk Assessment and Management: A Scoping Review," Journal of Multidisciplinary Healthcare, vol. 17, pp. 1-24, 2024.
[29] S. Liu et al., "An Empirical Study on Vulnerability Disclosure Management of Open Source Software Systems," ACM Transactions on Software Engineering and Methodology, vol. 34, no. 7, Sep. 2025.
[30] M. E. Durmuşkaya and S. Bayraklı, "Web application firewall based on machine learning models," *PeerJ Computer Science*, vol. 11, e2975, Jul. 2025.
[31] C. Betts and N. Power, "Enhancing incident reporting systems: insights from JOL Online and the emergency services," Cognition, Technology & Work, vol. 27, pp. 241-255, Apr. 2025.
[32] S. Buchyk, R. Ziubina, T. Yuzhakova, and A. Shabanova, "Detection of XSS vulnerabilities in OJS," International Journal of Electronics and Telecommunications, vol. 71, no. 1, pp. 101-106, 2025.
[33] T. Bakhshi, B. Ghita, and I. Kuzminykh, "A Review of IoT Firmware Vulnerabilities and Auditing Techniques," Sensors, vol. 24, no. 2, p. 708, Jan. 2024.
[34] R. Bakır, "UniEmbed: A Novel Approach to Detect XSS and SQL Injection Attacks Leveraging Multiple Feature Fusion with Machine Learning Techniques," Arabian Journal for Science and Engineering, vol. 50, pp. 15591-15604, Jan. 2025.
[35] M. R. Gupta, R. S. Jana, J. B. Ojha, and S. Borade, "UNALTERED: A FILE INTEGRATION MANAGEMENT SYSTEM," IJSREM, vol. 8, no. 4, Apr. 2024.
[36] Ismail et al., "Enhancing Security Operations Center: Wazuh Security Event Response with Retrieval-Augmented-Generation-Driven Copilot," Sensors, vol. 25, no. 3, p. 870, Jan. 2025.
[37] J. KONG et al., "Information Security Protection Scheme of Multi Meter Integration System," Electronics, Communications and Networks, pp. 251-259, 2024.

[38] J. R. Tadhani et al., "Securing web applications against XSS and SQLi attacks using a novel deep learning approach," Scientific Reports, vol. 14, no. 1803, 2024.
[39] Nelmiawati and K. Dealova, "Analysis of Polyglot Obfuscation Techniques against ModSecurity in Preventing Cross-Site Scripting (XSS) and SQL Injection Attacks with Experimental Method," JUTIF, vol. 6, no. 4, pp. 2540-2549, Aug. 2025.
[40] M. Annas, R. T. Adek, and Y. Afrillia, "Web Application Firewall (WAF) Design to Detect and Anticipate Hacking in Web-Based Applications," JACKA, vol. 1, no. 3, pp. 52-58, Jul. 2024.
[41] A. Kamil, D. Rizaludin, and A. T. Ni'mah, "Implementasi Wazuh FIM (File Integrity Monitoring) untuk Perlindungan Keamanan Sistem Informasi pada Unit Kegiatan Mahasiswa di Universitas Trunojoyo Madura," Sains Data: Jurnal Studi Matematika dan Teknologi, vol. 2, no. 2, pp. 80-92, 2024.
[42] A. R. Zain, P. Oktivasari, N. F. Soelaiman, and F. W. Umam, "Implementasi Intrusion Detection System (Ids) Suricata Dan Management Log Elk Stack Untuk Pendeteksian Kegiatan Mining," Jurnal Poli-Teknologi, vol. 22, no. 1, Jan. 2023.
[43] K. Abdulghaffar, N. Elmrabit, and M. Yousefi, "Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners," Computers, vol. 12, no. 11, p. 235, Nov. 2023.
[44] A. Suryadin et al., "Evaluasi Penerapan Pertahanan Proaktif Waf Dan Hids Terhadap Eksploitasi Kerentanan Aplikasi Web," Seminar Nasional CORISINDO, pp. 491-500, 2024.
[45] D. S. Ghazi et al., "SNORT VERSUS SURICATA IN INTRUSION DETECTION," Iraqi Journal of Information and Communications Technology (IJICT), vol. 7, no. 2, Aug. 2024.
[46] I. Heremba, N. S. Irjanto, and T. B. Bun, "Implementasi Intrusion Prevention System Menggunakan Suricata Sebagai Pengamanan Dari Serangan DDoS," Progresif: Jurnal Ilmiah Komputer, vol. 20, no. 2, pp. 364-376, 2024.
[47] M. N. A. Nur et al., "Strategi Keamanan Jurnal Berbasis OJS melalui Program Pendampingan dan Pencegahan Malware di Universitas Halu Oleo," Jurnal Pengabdian Masyarakat Ilmu Terapan, vol. 7, no. 2, pp. 182-192, Oct. 2025.
[48] N. F. Pratama, "Perancangan Sistem Deteksi Dini Keamanan Informasi DISKOMINFO Kabupaten Bandung," JATISI: Jurnal Teknik Informatika dan Sistem Informasi, vol. 10, no. 1, pp. 808-820, Mar. 2023.
[49] N. Hidayasari et al., "Implementasi Prototipe SIEM Berbasis Wazuh pada Website dengan Pengujian FIM dan Threat Hunting," JITSI: Jurnal Ilmiah Teknologi Sistem Informasi, vol. 6, no. 4, pp. 372-382, Dec. 2025.
[50] Z. Abdillah, P. Adytia, and M. Fahmi, "Implementasi Intrusion Detection System (IDS) Suricata Untuk Mendeteksi Serangan DDoS Menggunakan Metode TAARA Pada Jaringan Internet di Pixel Esport Arena Samarinda," Jurnal Widya Cipta Dharma, vol. 13, no. 1, 2024.