Journal of Authentic Research
https://journal-center.litpam.com/index.php/jar/index

August 2026 Vol. 5, No. 3
e-ISSN: 2828-3724
pp. 3988-4002

Rancang Bangun Dan Evaluasi Awal Aplikasi Early Warning Serangan Pada Web Server Berbasis Dual-Intrusion Detection System

[1*] Rizky Andika Putra, [2] Ahmaddul Hadi
[1,2] Universitas Negeri Padang, Jln. Prof. Dr. Hamka No.1, Air Tawar Barat, Kecamatan Padang Utara, Kota Padang, Sumatera Barat 25173, Indonesia
*Corresponding Author e-mail: andikarizky040@gmail.com

Received: Mei 2026; Revised: Juny 2026; Published: August 2026

Abstrak

Peningkatan insiden kebocoran data pribadi menuntut mekanisme keamanan web server yang tidak hanya bersifat reaktif, tetapi juga mampu memberikan peringatan dini sebelum serangan berkembang menjadi insiden yang lebih luas. Penelitian ini bertujuan merancang dan mengevaluasi awal aplikasi Early Warning System berbasis dual-Intrusion Detection System yang mengintegrasikan Suricata dan Snort dengan Log Analyzer berbasis Python, notifikasi Telegram, basis data MySQL, dan dashboard Flask. Pendekatan penelitian menggunakan Design Science Research melalui tahapan identifikasi masalah, perumusan tujuan solusi, desain dan pengembangan artefak, demonstrasi, evaluasi, serta komunikasi hasil. Evaluasi dilakukan pada lingkungan virtual terkontrol menggunakan simulasi SQL Injection, Brute Force, dan Port Scanning terhadap web server uji yang memuat data pribadi sintetis. Hasil awal pada 180 paket/skenario serangan simulatif menunjukkan bahwa sistem terintegrasi mampu menghasilkan Detection Rate 100%, False Positive Rate 0%, dan rata-rata Time-to-Detect 2 detik. Namun, hasil tersebut ditafsirkan secara terbatas karena belum merepresentasikan trafik produksi kompleks, serangan evasive, maupun uji beban skala besar. Kontribusi utama penelitian ini adalah rancangan arsitektur EWS ringan yang menggabungkan normalisasi log, filtering severity, deduplikasi, dan korelasi alert lintas-IDS untuk menghasilkan informasi yang lebih dapat ditindaklanjuti oleh administrator. Penelitian ini memperlihatkan potensi penerapan EWS berbasis open-source untuk mendukung kewajiban perlindungan data pribadi, dengan kebutuhan validasi lanjutan melalui eksperimen komparatif terhadap Snort saja, Suricata saja, dual-IDS tanpa korelasi, dan dual-IDS dengan Log Analyzer.

Kata kunci: Early Warning System; Dual-IDS; Suricata; Snort; Log Analyzer; Web Server; Pelindungan Data Pribadi

Design and Preliminary Evaluation of a Web Server Attack Early Warning Application Based on a Dual-Intrusion Detection System

Abstract

The rising incidence of personal data breaches necessitates web server security mechanisms that are not merely reactive but capable of providing early warnings before an attack escalates into a widespread incident. This study aims to design and conduct a preliminary evaluation of an Early Warning System (EWS) based on a dual-Intrusion Detection System (IDS) architecture, integrating Suricata and Snort with a Python-based Log Analyzer, Telegram notifications, a MySQL database, and a Flask dashboard. The research employs the Design Science Research methodology, encompassing problem identification, solution objective formulation, artifact design and development, demonstration, evaluation, and communication of results. Evaluation was conducted in a controlled virtual environment using simulated SQL Injection, Brute Force, and Port Scanning attacks against a test web server containing synthetic personal data. Preliminary results from 180 simulated attack packets/scenarios indicate that the integrated system achieved a 100% detection rate, a 0% false positive rate, and an average time-to-detect of 2 seconds. However, these results are interpreted with limitations, as they do not yet represent complex production traffic, evasive attacks, or large-scale load testing. The study's primary contribution is the design of a lightweight EWS architecture that combines log normalization, severity filtering, deduplication, and cross-IDS alert correlation to generate information that is more actionable for administrators. This research demonstrates the potential of implementing an open-source-based EWS to support personal data protection obligations, while highlighting the need for further validation through comparative experiments involving Snort-only, Suricata-only, dual-IDS without correlation, and dual-IDS with the Log Analyzer configurations.

Keywords: *Early Warning System; Dual-IDS; Suricata; Snort; Log Analyzer; Web Server; Personal Data Protection*

How to Cite: Putra, R. A., & Hadi, A. . (2026). Rancang Bangun Dan Evaluasi Awal Aplikasi Early Warning Serangan Pada Web Server Berbasis Dual-Intrusion Detection System. *Journal of Authentic Research*, *5*(3), 3988-4002. https://doi.org/10.36312/38d3wq50

https://doi.org/10.36312/38d3wq50

Copyright© 2026, Putra et al.
This is an open-access article under the CC-BY-SA License.

PENDAHULUAN

Transformasi digital telah meningkatkan ketergantungan organisasi terhadap aplikasi web, layanan berbasis cloud, dan basis data daring yang memproses informasi identitas, kredensial, data finansial, serta rekam aktivitas pengguna. Dalam ekosistem seperti ini, data pribadi berubah menjadi aset strategis sekaligus target bernilai tinggi bagi pelaku kejahatan siber karena dapat dimonetisasi, digunakan untuk eskalasi akses, atau dieksploitasi dalam serangan lanjutan. Laporan ancaman global menempatkan serangan terhadap data, penyalahgunaan kredensial, dan kompromi aplikasi sebagai risiko yang terus relevan, sehingga organisasi membutuhkan kapabilitas deteksi dan respons yang lebih cepat daripada sekadar mengandalkan pemeriksaan manual pascainsiden (ENISA, 2024; Verizon, 2025; Utami et al., 2025).

Konteks Indonesia memperkuat urgensi tersebut karena Undang-Undang Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi menempatkan pelindungan data sebagai kewajiban hukum bagi pengendali dan prosesor data pribadi. Kewajiban ini tidak hanya berkaitan dengan penyimpanan data secara aman, tetapi juga dengan kemampuan organisasi mencegah, mendeteksi, merespons, dan mendokumentasikan kegagalan pelindungan data. Dengan demikian, sistem keamanan web server perlu diposisikan sebagai bagian dari tata kelola risiko keamanan siber, bukan sekadar perangkat teknis tambahan yang bekerja setelah insiden terjadi (Republik Indonesia, 2022; NIST, 2024; NIST, 2025).

Pada praktiknya, banyak organisasi kecil dan menengah masih memiliki keterbatasan sumber daya, baik dari sisi biaya, tenaga analis keamanan, maupun infrastruktur Security Operation Center. Kondisi ini membuat alert mentah dari firewall, server log, atau IDS sering kali tidak segera dianalisis secara memadai, padahal keterlambatan deteksi dapat memperpanjang durasi paparan dan memperbesar dampak kebocoran data. Literatur mengenai intrusion detection juga menunjukkan bahwa tantangan utama sistem deteksi bukan hanya menangkap serangan, melainkan mengurangi alarm palsu, mengelola volume alert, dan mengubah log teknis menjadi informasi yang dapat ditindaklanjuti (Axelsson, 2000; Scarfone & Mell, 2007; Khraisat et al., 2019).

Intrusion Detection System merupakan salah satu pendekatan penting untuk memantau lalu lintas jaringan dan mengenali pola serangan berdasarkan signature, anomali, maupun analisis protokol. Suricata dan Snort banyak digunakan karena bersifat open-source, memiliki dukungan ruleset yang luas, serta mampu menghasilkan log yang dapat diproses lebih lanjut. Namun, keluaran default IDS sering kali masih berupa log mentah yang memerlukan normalisasi, korelasi, prioritisasi, dan visualisasi agar administrator dapat mengambil keputusan secara cepat dan tepat (Scarfone & Mell, 2007; OISF, 2026; Cisco Talos, n.d.).

Serangan terhadap web server yang relevan dengan penelitian ini mencakup SQL Injection, Brute Force, dan Port Scanning. SQL Injection tetap penting karena termasuk dalam kategori injection pada OWASP Top 10, terutama ketika input pengguna tidak divalidasi atau query basis data dibangun secara tidak aman. Brute Force dan credential stuffing juga relevan karena berkaitan dengan kegagalan identifikasi dan autentikasi, sedangkan Port Scanning sering menjadi tahap rekognisi sebelum eksploitasi layanan. Ketiga skenario ini memberikan dasar uji awal untuk menilai kemampuan sistem EWS mendeteksi ancaman yang umum terjadi pada

aplikasi web (OWASP Foundation, 2021a; OWASP Foundation, 2021b; Ramot et al., 2025).

Penelitian terdahulu telah menunjukkan manfaat integrasi IDS dengan media notifikasi seperti Telegram atau WhatsApp, misalnya melalui Snort yang mengirimkan alert kepada administrator. Akan tetapi, beberapa studi masih berfokus pada pengiriman alert mentah dan belum menekankan normalisasi skema log, filtering severity, deduplikasi, korelasi lintas-IDS, maupun penyajian historis melalui dashboard interaktif. Di sisi lain, pendekatan berbasis machine learning dapat meningkatkan kemampuan deteksi anomali, tetapi sering memerlukan dataset, pelatihan model, dan sumber daya komputasi yang tidak selalu tersedia pada organisasi kecil dan menengah (Erlansari et al., 2020; Purnama et al., 2023; Safitri et al., 2024; Buczak & Guven, 2016; Nikhlis & Ghofur, 2025).

Research gap dalam penelitian ini terletak pada kebutuhan artefak EWS yang ringan, dapat direplikasi, dan mampu mengubah alert IDS menjadi informasi prioritas melalui mekanisme korelasi yang eksplisit. Gap ini bersifat teknis sekaligus ilmiah karena performa IDS tidak dapat diklaim hanya dari keberhasilan instalasi alat, tetapi harus dijelaskan melalui desain evaluasi, definisi metrik, batasan skenario, serta pembandingan terhadap studi terdahulu. Literatur IDS menegaskan bahwa klaim akurasi tinggi perlu dibaca dengan hati-hati apabila trafik normal terbatas, payload serangan terlalu mudah dikenali, atau belum ada evaluasi terhadap serangan yang dimodifikasi (Axelsson, 2000; Ring et al., 2019; Khraisat et al., 2019).

Berdasarkan gap tersebut, penelitian ini bertujuan untuk: merancang arsitektur EWS berbasis dual-IDS Suricata dan Snort; mengimplementasikan Log Analyzer Python yang melakukan normalisasi, filtering severity, deduplikasi, dan korelasi alert; menyediakan notifikasi Telegram dan dashboard Flask untuk pemantauan; serta mengevaluasi kinerja awal sistem menggunakan Detection Rate, False Positive Rate, dan Time-to-Detect pada simulasi SQL Injection, Brute Force, dan Port Scanning. Kebaruan penelitian ini terletak pada formulasi arsitektur dan alur korelasi alert lintas-IDS yang dirancang untuk menghasilkan actionable alert bagi administrator, dengan klaim performa yang dibatasi pada lingkungan virtual terkontrol (Hevner et al., 2004; Peffers et al., 2007; NIST, 2025).

METODE

Jenis dan Kerangka Penelitian

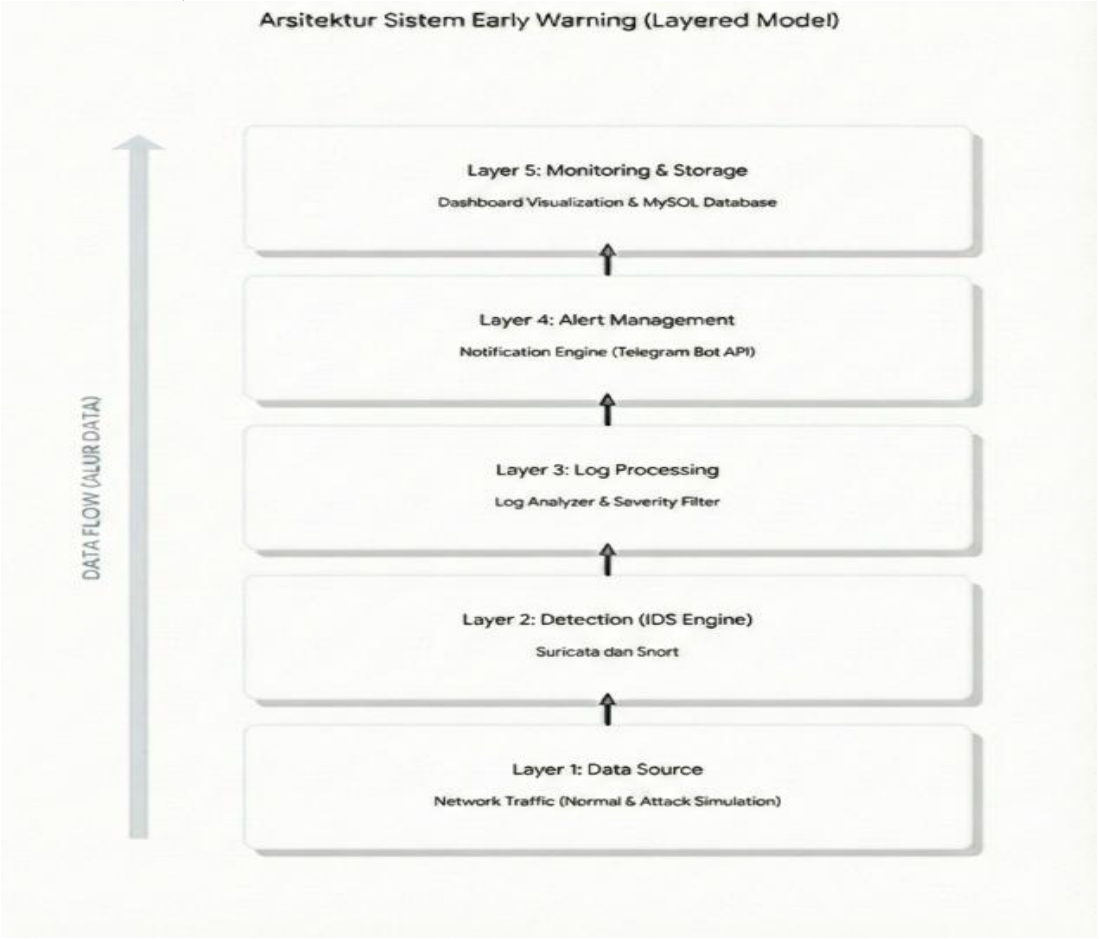
Penelitian ini menggunakan pendekatan Design Science Research karena tujuan utamanya adalah merancang, membangun, dan mengevaluasi artefak berupa aplikasi Early Warning System untuk mendukung deteksi dini serangan pada web server. Design Science Research relevan untuk penelitian sistem informasi yang menghasilkan artefak teknologis, seperti model, metode, instansiasi perangkat lunak, atau arsitektur sistem, dengan evaluasi terhadap utilitas artefak dalam konteks masalah yang jelas. Tahapan penelitian mengikuti enam aktivitas utama: identifikasi masalah dan motivasi, penentuan tujuan solusi, desain dan pengembangan, demonstrasi, evaluasi, dan komunikasi hasil (Hevner et al., 2004; Peffers et al., 2007).

Tabel 1. Operasionalisasi tahapan Design Science Research

Tahapan DSR	Implementasi dalam penelitian	Luaran
Identifikasi masalah	Kebutuhan deteksi dini kebocoran data pada web server dan keterbatasan alert mentah IDS.	Rumusan masalah dan gap penelitian.
Tujuan solusi	Merancang EWS ringan dengan dual-IDS, Log Analyzer, notifikasi, dan dashboard.	Spesifikasi kebutuhan fungsional.
Desain dan pengembangan	Membangun arsitektur Suricata–Snort–Python–MySQL–Flask–Telegram.	Prototipe sistem.
Demonstrasi	Menjalankan prototipe pada topologi VM dengan simulasi SQLi, Brute Force, dan Port Scanning.	Bukti fungsional sistem.
Evaluasi	Mengukur DR, FPR, dan TTD serta membatasi interpretasi sesuai cakupan pengujian.	Ringkasan kinerja awal.
Komunikasi	Menyajikan arsitektur, algoritma korelasi, hasil evaluasi, pembahasan, dan batasan.	Artikel ilmiah revisi.

Arsitektur Sistem Early Warning

Arsitektur sistem dirancang dalam lima lapisan agar aliran data keamanan dapat ditelusuri dari sumber trafik hingga peringatan yang diterima administrator. Lapisan tersebut meliputi Data Source Layer, Detection Layer, Log Processing Layer, Alert Management Layer, serta Monitoring and Storage Layer. Pembagian lapisan ini mengikuti prinsip IDPS yang memisahkan fungsi sensor, analisis, penyimpanan, dan manajemen alert sehingga sistem lebih mudah dipelihara, diuji ulang, dan dikembangkan menjadi sistem pencegahan pada tahap berikutnya (Scarfone & Mell, 2007; NIST, 2025).



Gambar 1. Arsitektur sistem Early Warning System berbasis dual-IDS

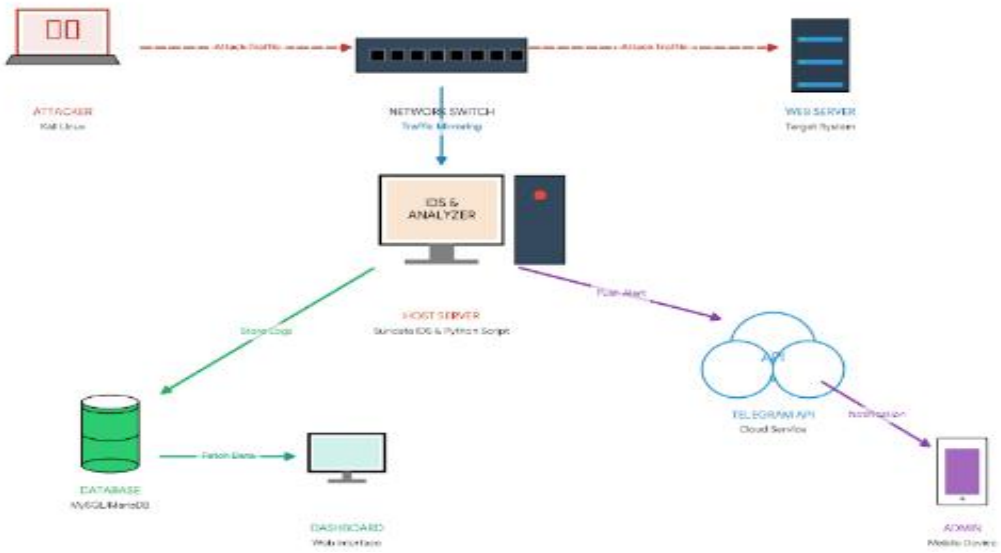
Pada Detection Layer, Suricata dan Snort dikonfigurasi untuk memantau salinan lalu lintas yang sama agar keduanya dapat menghasilkan perspektif deteksi yang saling melengkapi. Suricata digunakan karena mendukung EVE JSON yang memuat alert, anomali, metadata, dan informasi protokol dalam format terstruktur, sedangkan Snort digunakan karena memiliki basis signature yang luas dan telah lama digunakan sebagai IDS/IPS open-source. Penggabungan dua engine ini tidak diperlakukan sebagai jaminan otomatis peningkatan akurasi, tetapi sebagai desain redundansi deteksi yang perlu dievaluasi melalui korelasi dan pembandingan kinerja (OISF, 2026; Cisco Talos, n.d.; Shah & Issac, 2017; Ghazi et al., 2024).

Tabel 2. Fungsi lapisan arsitektur sistem

Lapisan	Komponen	Fungsi utama
Data Source Layer	Trafik normal dan trafik serangan simulatif	Menyediakan data jaringan yang dipantau oleh IDS.
Detection Layer	Suricata dan Snort	Mendeteksi signature atau pola anomali dan menghasilkan alert.
Log Processing Layer	Python Log Analyzer	Normalisasi log, filtering severity, deduplikasi, dan korelasi alert.
Alert Management Layer	Telegram Bot API	Mengirim peringatan kondisional untuk alert High dan Critical.
Monitoring & Storage Layer	MySQL dan Flask Dashboard	Menyimpan riwayat alert dan menyediakan visualisasi historis.

Topologi dan Lingkungan Pengujian

Pengujian dilakukan dalam lingkungan laboratorium virtual terkontrol yang mensimulasikan jaringan produksi skala kecil. Topologi terdiri atas mesin attacker berbasis Kali Linux, router atau bridge yang melakukan port mirroring, web server target yang menjalankan aplikasi dummy dengan data pribadi sintetis, serta server EWS yang menjalankan Suricata, Snort, Log Analyzer, MySQL, Flask, dan Telegram integration. Desain ini dipilih agar serangan dapat dikendalikan, lalu lintas dapat direkam, dan hasil deteksi dapat ditelusuri ulang tanpa mengekspos data pribadi nyata (OWASP Foundation, 2021a; OWASP Foundation, 2021b; Scarfone & Mell, 2007).



Gambar 2. Topologi jaringan pengujian dalam lingkungan virtual terkontrol

Untuk meningkatkan replikasi, penelitian ini mendokumentasikan konfigurasi teknis inti yang digunakan dalam prototipe dan pengujian awal, meliputi sistem operasi, engine IDS, komponen backend, framework dashboard, database, skenario trafik, dan jumlah serangan simulatif. Detail seperti ruleset, durasi trafik normal, variasi payload, serta konfigurasi sumber daya komputasi perlu dijaga konsisten dalam setiap pengujian ulang karena validitas evaluasi IDS sangat dipengaruhi oleh lingkungan eksperimen dan karakteristik trafik yang digunakan (Ring et al., 2019; Khraisat et al., 2019; NIST, 2025).

Tabel 3. Parameter lingkungan pengujian prototipe

Parameter	Nilai pada penelitian
Sistem operasi server EWS	Ubuntu 22.04 LTS pada Virtual Machine.
IDS Engine	Suricata dan Snort pada versi stabil yang digunakan saat pengujian.
Backend dan Log Analyzer	Python 3.10 dengan modul json, requests, dan mysql-connector.
Web framework dan database	Flask dan MySQL Server 8.0.
Ruleset IDS	Ruleset signature IDS yang aktif pada saat pengujian.
Trafik normal	Trafik normal tanpa serangan untuk pengukuran FP/TN.
Serangan simulatif	SQL Injection, Brute Force, dan Port Scanning.
Jumlah serangan awal	180 paket/skenario serangan simulatif berdasarkan naskah awal.

Mekanisme Log Analyzer dan Korelasi Alert

Log Analyzer dikembangkan sebagai daemon Python yang membaca keluaran Suricata dan Snort secara periodik atau near real-time, kemudian mengubah struktur log yang berbeda menjadi skema alert terpadu. Skema terpadu mencakup timestamp, engine, source IP, destination IP, destination port, protocol, signature identifier, signature message, attack class, severity, raw event, dan correlation key. Normalisasi ini diperlukan karena alert dari dua IDS tidak selalu memiliki penamaan field, format severity, atau struktur metadata yang sama (Scarfone & Mell, 2007; OISF, 2026; Cisco Talos, n.d.).

Korelasi alert dilakukan dalam tiga tahap. Pertama, sistem melakukan filtering awal dengan hanya memprioritaskan event yang memenuhi severity Medium, High, atau Critical, sedangkan alert informasional tetap disimpan tanpa memicu notifikasi. Kedua, sistem melakukan deduplikasi berdasarkan kombinasi source IP, destination IP, destination port, attack class, dan jendela waktu tertentu, misalnya 5–10 detik. Ketiga, alert Suricata dan Snort yang memiliki correlation key serupa ditandai sebagai confirmed cross-engine alert, sedangkan alert yang hanya muncul pada satu engine diberi label single-engine alert agar dapat dianalisis ulang oleh administrator. Mekanisme ini membatasi penggunaan istilah korelasi agar tidak sekadar berarti penggabungan log, melainkan proses pengelompokan, validasi silang, dan prioritisasi alert (Axelsson, 2000; Scarfone & Mell, 2007; Khraisat et al., 2019).

Skenario Pengujian dan Metrik Evaluasi

Pengujian menggunakan simulasi serangan yang umum pada aplikasi web. SQL Injection diuji melalui SQLMap dan payload manual pada parameter URL atau form login; Brute Force diuji menggunakan Hydra terhadap form autentikasi; sedangkan Port Scanning digunakan sebagai skenario rekognisi terhadap layanan web server.

Pemilihan ketiga skenario didasarkan pada relevansinya terhadap risiko aplikasi web, kategori OWASP Top 10, dan praktik umum evaluasi IDS berbasis signature (OWASP Foundation, 2021a; OWASP Foundation, 2021b; Ramot et al., 2025).

Untuk memperkuat validitas, evaluasi idealnya tidak hanya dilakukan terhadap sistem dual-IDS final, tetapi juga melalui uji ablation atau komparatif: Snort saja, Suricata saja, dual-IDS tanpa korelasi, dan dual-IDS dengan Log Analyzer. Naskah awal belum menyediakan data komparatif tersebut, sehingga hasil yang dilaporkan dalam artikel ini dibatasi sebagai evaluasi awal terhadap konfigurasi sistem terintegrasi. Pengujian komparatif perlu ditambahkan apabila penulis ingin menyatakan bahwa dual-IDS secara empiris lebih unggul daripada single-IDS (Shah & Issac, 2017; Ghazi et al., 2024; Khraisat et al., 2019).

Tabel 4. Metrik evaluasi kinerja sistem

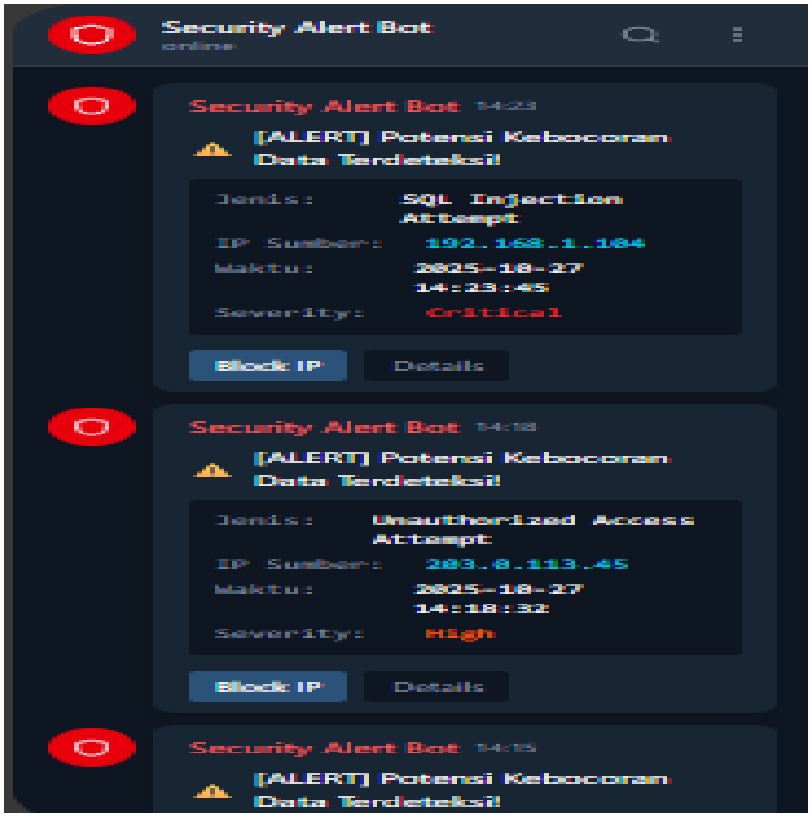
Metrik	Definisi	Formula	Target awal
Detection Rate / Recall	Persentase serangan yang berhasil dideteksi.	DR = TP / (TP + FN) × 100%	> 95%
False Positive Rate	Persentase trafik normal yang salah diklasifikasikan sebagai serangan.	FPR = FP / (FP + TN) × 100%	< 5%
Precision	Proporsi alert positif yang benar-benar merupakan serangan.	Precision = TP / (TP + FP) × 100%	Dilaporkan bila FP tersedia
Time-to-Detect	Selisih waktu antara serangan dimulai dan notifikasi diterima.	TTD = t_notifikasi − t_serangan	< 10 detik
Mean Time-to-Detect	Rata-rata TTD seluruh skenario.	Mean TTD = ΣTTD_i / n	Semakin rendah semakin baik

HASIL DAN PEMBAHASAN

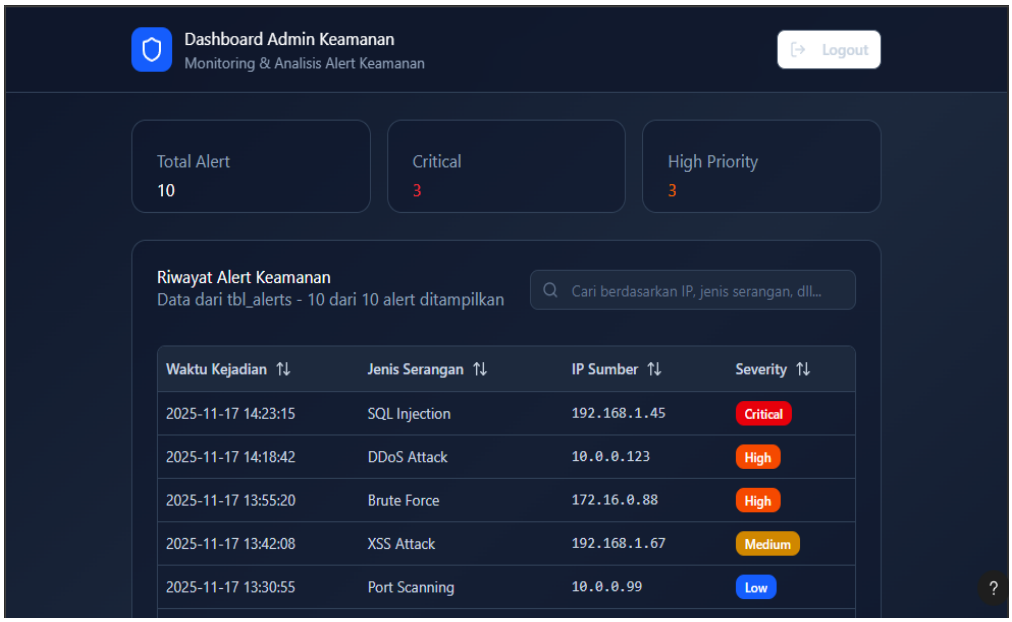
Implementasi Sistem EWS Berbasis Dual-IDS

Prototipe EWS berhasil diimplementasikan dengan mengintegrasikan Suricata, Snort, Python Log Analyzer, MySQL, Telegram Bot API, dan dashboard Flask. Sistem bekerja dengan memantau lalu lintas menuju web server target, membaca alert dari kedua IDS, menormalkan format log, mengklasifikasikan severity, menyimpan riwayat alert, dan mengirimkan notifikasi ketika alert memenuhi kriteria prioritas. Dari sisi desain artefak, implementasi ini menjawab kebutuhan sistem deteksi dini yang tidak hanya menghasilkan log, tetapi juga menyediakan alur eskalasi informasi bagi administrator (Hevner et al., 2004; Peffers et al., 2007; NIST, 2025).

Notifikasi Telegram digunakan sebagai kanal peringatan cepat karena memungkinkan administrator menerima alert tanpa harus terus-menerus membuka dashboard. Namun, notifikasi hanya dikirim untuk alert High dan Critical agar tidak menimbulkan alert fatigue. Strategi ini sejalan dengan prinsip incident response yang menekankan prioritisasi kejadian, komunikasi cepat, dan dokumentasi insiden agar organisasi dapat merespons ancaman secara proporsional (NIST, 2025; OWASP Foundation, 2021c; CIS, 2024).



Gambar 3. Tampilan notifikasi Telegram



Gambar 4. Tampilan dashboard analisis keamanan

Hasil Normalisasi, Filtering Severity, dan Korelasi Alert

Log Analyzer mengklasifikasikan alert berdasarkan jenis serangan dan tingkat keparahan. SQL Injection ditempatkan sebagai Critical karena berpotensi langsung mengakses atau memanipulasi basis data, sedangkan Brute Force ditempatkan sebagai High karena berkaitan dengan upaya kompromi kredensial. Port Scanning ditempatkan sebagai Medium karena berfungsi sebagai indikasi rekognisi awal yang belum tentu menghasilkan kompromi langsung, tetapi tetap penting sebagai indikator tahap awal rantai serangan (OWASP Foundation, 2021a; OWASP Foundation, 2021b; Scarfone & Mell, 2007).

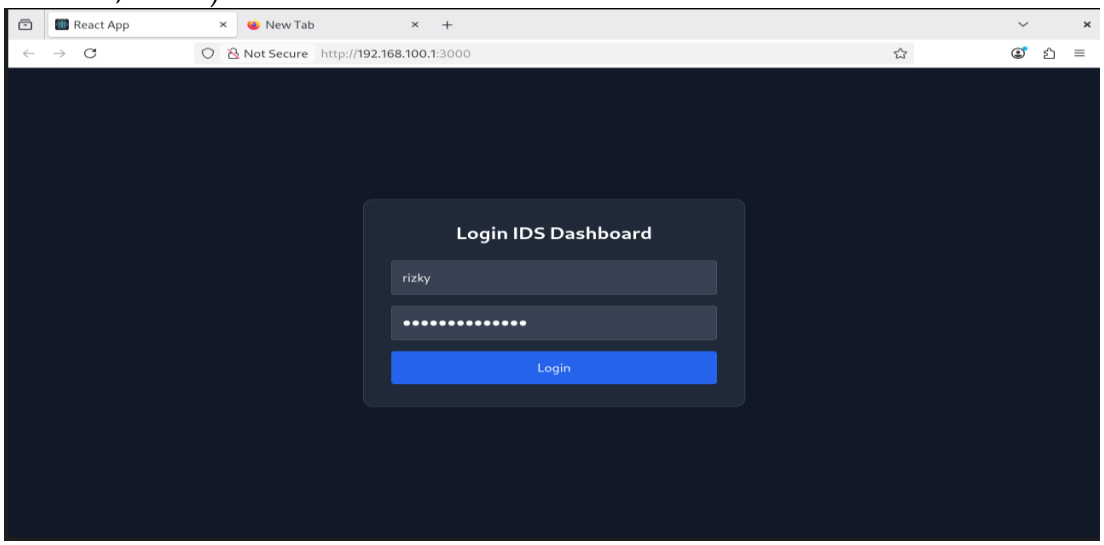
Tabel 5. Sampel hasil deteksi alert pada pengujian awal

Waktu Serangan	Jenis Serangan	Severity	Status Deteksi	Interpretasi
2026-04-17 16:50:50	SQL Injection	Critical	Terdeteksi	Ancaman langsung terhadap query dan data basis data.
2026-04-17 17:01:07	Brute Force	High	Terdeteksi	Upaya akses tidak sah melalui percobaan kredensial berulang.
2026-04-17 17:01:08	Port Scanning	Medium	Terdeteksi	Indikasi rekognisi layanan sebelum eksploitasi.

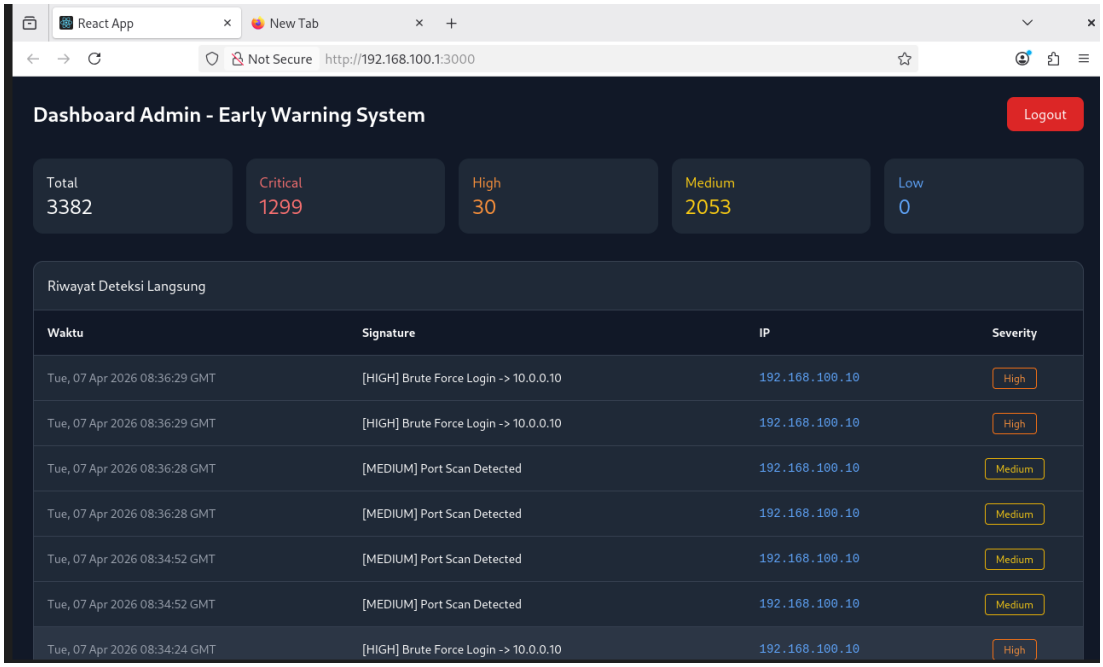
Tabel 5 menunjukkan bahwa sistem mampu mencatat timestamp, jenis serangan, severity, dan status deteksi sebagai dasar prioritisasi peringatan. Namun, istilah "akurasi tinggi" tidak boleh disimpulkan hanya dari tiga contoh alert, karena validitas deteksi IDS sangat dipengaruhi oleh variasi payload, volume trafik normal, dan kondisi jaringan. Oleh karena itu, data sampel harus dibaca sebagai bukti fungsional bahwa pipeline deteksi berjalan, bukan sebagai pembuktian final bahwa sistem bebas dari false positive pada lingkungan produksi (Axelsson, 2000; Ring et al., 2019; Khraisat et al., 2019).

Hasil Pengembangan Antarmuka Pengguna

Dashboard dikembangkan menggunakan Flask untuk menyediakan akses terpusat terhadap riwayat alert, statistik severity, dan tabel interaktif. Fitur login ditambahkan untuk membatasi akses terhadap informasi keamanan yang sensitif, sementara tabel alert dapat digunakan untuk pencarian, pengurutan, dan analisis historis. Dalam konteks keamanan operasional, dashboard tidak hanya berfungsi sebagai antarmuka visual, tetapi juga sebagai media audit yang mendukung dokumentasi insiden dan pelacakan respons (NIST, 2025; CIS, 2024; OWASP Foundation, 2021c).



Gambar 5. Halaman login dashboard



Gambar 6. Dashboard pemantauan alert keamanan

Fitur dashboard yang paling relevan bagi administrator adalah ringkasan alert berdasarkan severity, riwayat event, pencarian berdasarkan IP sumber atau jenis serangan, serta pembaruan data yang mendekati real-time. Fitur-fitur ini memperkuat aspek actionable intelligence karena alert yang telah difilter dan dikelompokkan lebih mudah diprioritaskan daripada log mentah yang tersebar di file sistem. Meski demikian, keamanan dashboard itu sendiri masih perlu diuji lebih lanjut, terutama pada autentikasi, manajemen sesi, pembatasan akses, dan perlindungan terhadap serangan web umum (OWASP Foundation, 2021b; OWASP Foundation, 2021c; NIST, 2025).

Evaluasi Kinerja Awal Sistem

Pengujian awal dilakukan pada 180 paket atau skenario serangan simulatif dalam lingkungan virtual terkontrol. Berdasarkan data naskah awal, seluruh serangan yang dilancarkan berhasil menghasilkan alert sehingga Detection Rate mencapai 100% pada konfigurasi dual-IDS dengan Log Analyzer. Karena tidak terdapat false negative pada skenario uji tersebut, hasil ini menunjukkan bahwa ruleset dan pipeline pemrosesan log mampu mengenali pola serangan yang disimulasikan. Namun, hasil ini tidak boleh digeneralisasi ke seluruh jenis serangan web atau lalu lintas produksi tanpa uji lanjutan yang lebih beragam (Axelsson, 2000; Ring et al., 2019; Khraisat et al., 2019).

False Positive Rate pada pengujian awal dilaporkan 0%, tetapi interpretasinya perlu dibatasi karena naskah awal belum menjelaskan jumlah dan keragaman trafik normal yang digunakan sebagai pembagi FPR. Dalam evaluasi IDS, trafik normal seharusnya mencakup aktivitas login valid, request halaman, query basis data yang sah, unggah berkas, crawling, request API, aktivitas admin, dan variasi user-agent agar sistem diuji terhadap kondisi yang mendekati produksi. Tanpa informasi tersebut, FPR 0% lebih tepat disebut sebagai hasil awal pada trafik normal terbatas, bukan bukti final bahwa sistem minim noise pada lingkungan nyata (Axelsson, 2000; Ring et al., 2019; Khraisat et al., 2019).

Rata-rata Time-to-Detect sebesar 2 detik menunjukkan bahwa proses pembacaan log, filtering severity, penyimpanan database, dan pengiriman notifikasi Telegram

berjalan cepat dalam lingkungan uji. Kecepatan ini merupakan aspek penting karena respons insiden membutuhkan waktu deteksi yang pendek agar administrator dapat melakukan containment lebih awal. Akan tetapi, TTD perlu diuji kembali pada beban trafik lebih tinggi, ukuran log yang lebih besar, latensi jaringan berbeda, serta kondisi API Telegram yang tidak selalu stabil agar klaim kecepatan sistem lebih kuat secara empiris (NIST, 2025; Scarfone & Mell, 2007; CIS, 2024).

Tabel 6. Rekapitulasi hasil evaluasi awal sistem

Parameter	Target	Hasil pengujian awal	Interpretasi revisi
Detection Rate	> 95%	100% pada 180 serangan simulatif	Menjanjikan pada skenario terbatas; perlu uji payload variatif.
False Positive Rate	< 5%	0% pada trafik normal yang belum dijelaskan rinci	Belum cukup untuk klaim bebas false positive pada produksi.
Mean Time-to-Detect	< 10 detik	2 detik	Cepat pada VM terkontrol; perlu uji beban dan stabilitas.

Perbandingan dengan Penelitian Terdahulu

Posisi kebaruan penelitian menjadi lebih jelas ketika dibandingkan dengan studi terdahulu. Studi Snort dengan Telegram umumnya kuat pada aspek notifikasi, tetapi tidak selalu menyediakan dashboard historis, normalisasi lintas-engine, atau mekanisme korelasi alert. Studi berbasis machine learning menawarkan potensi deteksi anomali yang lebih adaptif, tetapi memerlukan dataset, pelatihan, dan validasi model yang lebih kompleks. Dibandingkan pendekatan tersebut, penelitian ini menekankan artefak EWS berbasis open-source yang ringan dan dapat direplikasi, dengan kontribusi pada desain integrasi, prioritisasi, dan visualisasi alert (Erlansari et al., 2020; Purnama et al., 2023; Safitri et al., 2024; Buczak & Guven, 2016; Nikhlis & Ghofur, 2025).

Tabel 7. Perbandingan posisi penelitian dengan studi terdahulu

Studi	IDS/Metode	Notifikasi	Filtering severity	Dashboard	Metrik evaluasi	Keterbatasan utama
Erlansari et al. (2020)	Snort	Telegram	Terbatas	Tidak dominan	Deteksi awal	Alert cenderung mentah.
Purnama et al. (2023)	Snort	WhatsApp/Telegram	Terbatas	Terbatas	Implementasi IDS	Fokus notifikasi, bukan korelasi.
Safitri et al. (2024)	Snort	Telegram	Terbatas	Tidak dominan	Syn Flood/Ping of Death	Jenis serangan berbeda dari web attack.
Nikhlis & Ghofur (2025)	Machine learning	Tidak utama	Model-based	Tidak utama	Akurasi model	Butuh dataset dan komputasi.
Penelitian ini	Suricata + Snort + Log Analyzer	Telegram kondisional	Ada	Ada	DR, FPR, TTD	Masih perlu uji komparatif dan trafik produksi.

Pembahasan Kritis terhadap Validitas Hasil

Hasil awal menunjukkan bahwa arsitektur EWS yang dikembangkan mampu bekerja sebagai pipeline deteksi, korelasi, notifikasi, dan visualisasi. Akan tetapi,

klaim bahwa sistem "optimal", "andal", atau "akurat" perlu diganti dengan istilah yang lebih hati-hati, misalnya "menunjukkan performa awal yang menjanjikan dalam lingkungan virtual terkontrol". Perubahan diksi ini penting karena penelitian IDS sangat sensitif terhadap cakupan data uji, base-rate serangan, variasi trafik normal, dan kompleksitas payload yang digunakan (Axelsson, 2000; Ring et al., 2019; Khraisat et al., 2019).

Kontribusi dual-IDS juga perlu dibedakan antara kontribusi implementatif dan kontribusi ilmiah. Secara implementatif, integrasi Suricata dan Snort memperluas sumber alert dan memungkinkan validasi silang antar-engine. Secara ilmiah, keunggulan dual-IDS atas single-IDS baru dapat dibuktikan apabila tersedia eksperimen komparatif yang mengukur Snort saja, Suricata saja, dual-IDS tanpa korelasi, dan dual-IDS dengan Log Analyzer pada dataset dan skenario yang sama. Karena naskah awal belum menyediakan data tersebut, penelitian ini sebaiknya menyatakan dual-IDS sebagai rancangan korelatif yang potensial, bukan sebagai pendekatan yang telah terbukti lebih unggul secara kausal (Shah & Issac, 2017; Ghazi et al., 2024; Scarfone & Mell, 2007).

Mekanisme korelasi alert menjadi bagian penting yang memperkuat novelty. Apabila sistem hanya menggabungkan dua file log, maka istilah korelasi terlalu kuat. Dalam revisi ini, korelasi didefinisikan sebagai normalisasi, pencocokan correlation key, deduplikasi dalam jendela waktu, validasi silang lintas-engine, dan penetapan severity tertinggi. Definisi ini membuat kontribusi sistem lebih eksplisit dan dapat diuji ulang oleh peneliti lain melalui dataset atau topologi yang berbeda (Scarfone & Mell, 2007; OISF, 2026; Cisco Talos, n.d.).

Keterbatasan penelitian perlu dinyatakan secara terbuka. Pertama, pengujian masih menggunakan lingkungan VM dan belum menggunakan trafik produksi nyata. Kedua, variasi payload serangan belum dijelaskan secara rinci, termasuk apakah terdapat payload obfuscated atau evasive. Ketiga, belum ada uji overhead CPU, RAM, throughput, latency, dan packet drop untuk membuktikan klaim ringan. Keempat, belum ada uji skalabilitas, keamanan dashboard, dan usability oleh administrator. Keterbatasan ini tidak mengurangi nilai artefak, tetapi membatasi generalisasi temuan dan menjadi dasar agenda riset berikutnya (Ring et al., 2019; Khraisat et al., 2019; NIST, 2025).

Dari perspektif kepatuhan UU PDP, EWS tidak dapat diperlakukan sebagai satu-satunya kontrol pelindungan data pribadi. Sistem ini lebih tepat diposisikan sebagai kontrol deteksi dan dukungan respons insiden yang melengkapi kontrol pencegahan, seperti validasi input, hardening server, manajemen akses, enkripsi, patching, backup, dan prosedur incident response. Dengan posisi tersebut, kontribusi penelitian menjadi lebih proporsional: EWS membantu mempercepat kesadaran administrator terhadap indikasi serangan dan menyediakan dokumentasi alert untuk tindak lanjut keamanan (Republik Indonesia, 2022; NIST, 2024; NIST, 2025; CIS, 2024).

KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan aplikasi Early Warning System berbasis dual-IDS yang mengintegrasikan Suricata, Snort, Python Log Analyzer, MySQL, Telegram Bot API, dan dashboard Flask untuk mendeteksi dini serangan pada web server. Kontribusi utama penelitian terletak pada rancangan pipeline yang menormalkan log dari dua IDS, melakukan filtering severity, deduplikasi, korelasi alert lintas-engine, dan menyajikan informasi keamanan dalam

bentuk notifikasi serta dashboard historis. Evaluasi awal terhadap 180 serangan simulatif menunjukkan Detection Rate 100%, False Positive Rate 0%, dan rata-rata Time-to-Detect 2 detik, tetapi temuan tersebut harus dibatasi pada lingkungan virtual terkontrol dengan skenario yang belum merepresentasikan kompleksitas trafik produksi. Oleh karena itu, sistem ini lebih tepat disimpulkan sebagai prototipe EWS yang menunjukkan performa awal menjanjikan, bukan sebagai sistem yang telah terbukti optimal pada seluruh kondisi operasional. Penelitian ini berkontribusi pada pengembangan solusi keamanan siber berbasis open-source untuk organisasi dengan sumber daya terbatas, khususnya dalam mendukung deteksi dan dokumentasi potensi insiden pelindungan data pribadi. Penelitian selanjutnya perlu menambahkan pengujian komparatif Snort saja, Suricata saja, dual-IDS tanpa korelasi, dan dual-IDS dengan Log Analyzer; memperluas variasi trafik normal dan payload serangan; serta mengukur overhead CPU, RAM, throughput, packet drop, keamanan dashboard, dan usability administrator (Republik Indonesia, 2022; NIST, 2025; Ring et al., 2019).

REKOMENDASI

Berdasarkan keterbatasan penelitian, pengembangan berikutnya disarankan menambahkan modul Intrusion Prevention System untuk pemblokiran otomatis yang tetap dikontrol oleh kebijakan administrator agar tidak menimbulkan dampak operasional akibat false positive. Sistem juga perlu diintegrasikan dengan SIEM atau log management platform seperti ELK Stack atau Wazuh agar korelasi dapat diperluas ke log aplikasi, autentikasi, sistem operasi, dan firewall. Pengukuran overhead sistem, packet drop, latency dashboard, serta stabilitas notifikasi perlu dilakukan untuk membuktikan klaim ringan dan skalabel (Scarfone & Mell, 2007; NIST, 2025; CIS, 2024).

Dari sisi riset lanjutan, penulis perlu melakukan pengujian komparatif dan variasi payload yang lebih luas, termasuk SQL Injection termodifikasi, credential stuffing, brute force lambat, port scanning stealth, dan serangan yang mencoba menghindari signature IDS. Evaluasi berbasis dataset atau trafik semi-riil juga direkomendasikan agar nilai Detection Rate dan False Positive Rate tidak hanya mencerminkan skenario laboratorium yang sederhana. Pendekatan machine learning dapat dipertimbangkan untuk anomali dan zero-day, tetapi harus disertai validasi dataset, explainability, dan kontrol terhadap false positive (Buczak & Guven, 2016; Ring et al., 2019; Khraisat et al., 2019).

REFERENSI

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. ACM Transactions on Information and System Security, 3(3), 186–205. https://doi.org/10.1145/357830.357849

Azeez, K., Al-Husseini, O., & Obaid, A. H. (2018). Usage of prototyping in software testing. Multi-Knowledge Electronic Comprehensive Journal for Education and Science Publications, 14.

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

Cisco Talos. (n.d.). Snort users manual and documentation. https://www.snort.org/documents

Center for Internet Security. (2024). CIS Critical Security Controls Version 8.1. https://www.cisecurity.org/controls/v8-1

ENISA. (2024). ENISA Threat Landscape 2024. European Union Agency for Cybersecurity.

Erlansari, A., Coastera, F. F., & Husamudin, A. (2020). Early Intrusion Detection System (IDS) using Snort and Telegram approach. SISFORMA: Journal of Information Systems, 7(1). https://doi.org/10.24167/sisforma

Furqan, N., Suandi, I., & Teknologi Rekayasa Jaringan Telekomunikasi Jurusan Teknik Elektro Politeknik Negeri Lhokseumawe. (2023). Implementasi Intrusion Detection System (IDS) pada sistem keamanan jaringan menggunakan Telegram sebagai media notifikasi. Jurnal Tektro, 7(1), 44.

Ghazi, D. S., Hamid, H. S., Zaiter, M. J., & Behadili, A. S. G. (2024). Performance and efficacy of Snort versus Suricata in intrusion detection: A benchmark analysis. AIP Conference Proceedings, 3232(1), 020024. https://doi.org/10.1063/5.0236936

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75–105.

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2, 20. https://doi.org/10.1186/s42400-019-0038-7

National Institute of Standards and Technology. (2024). The NIST Cybersecurity Framework (CSF) 2.0. NIST.

National Institute of Standards and Technology. (2025). Incident Response Recommendations and Considerations for Cybersecurity Risk Management: A CSF 2.0 Community Profile (NIST SP 800-61 Rev. 3). NIST.

Nikhlis, N., & Ghofur, M. J. U. (2025). Deteksi dini pencurian data pada perangkat seluler menggunakan machine learning. Jurnal Ilmiah Sistem Informasi, 4(2), 110–129. https://doi.org/10.51903/fz0xzt26

Open Information Security Foundation. (2026). Suricata User Guide 8.0.5. https://docs.suricata.io/

OWASP Foundation. (2021a). A03:2021 Injection. OWASP Top 10:2021.

OWASP Foundation. (2021b). A07:2021 Identification and Authentication Failures. OWASP Top 10:2021.

OWASP Foundation. (2021c). A09:2021 Security Logging and Monitoring Failures. OWASP Top 10:2021.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Purnama, T., Muhyidin, Y., & Singasatia, D. (2023). Implementasi Intrusion Detection System (IDS) Snort sebagai sistem keamanan menggunakan WhatsApp dan Telegram sebagai media notifikasi. Jurnal Teknologi Informasi dan Komunikasi, 14(2), 358–369.

Ramot, F., Pasaribu, A., & Maslan, A. (2025). Analisis performa Intrusion Detection System Snort dan Suricata terhadap serangan SQL Injection. Jurnal Comasie, 13(2).

Republik Indonesia. (2022). Undang-Undang Republik Indonesia Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147–167. https://doi.org/10.1016/j.cose.2019.06.005

Safitri, Z. A., Haerani, E., Muzawi, R., Affandes, M., & Pizaini. (2024). Intrusion Detection System (IDS) pada Snort dengan Bot Telegram sebagai sistem notifikasi terhadap serangan Syn Flood dan Ping of Death. SATIN - Sains dan Teknologi Informasi, 9(2), 1–11. https://doi.org/10.33372/stn.v9i2.1000

Scarfone, K., & Mell, P. (2007). Guide to Intrusion Detection and Prevention Systems (IDPS) (NIST SP 800-94). National Institute of Standards and Technology.

Shah, S. A. R., & Issac, B. (2017). Performance comparison of intrusion detection systems and application of machine learning to Snort system. arXiv:1710.04843.

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

Sufardy, D. B., Widiasari, & Indrastanti, R. (2024). The use of pfSense and Suricata as a network security attack detection and prevention tool on web servers. 9(2).

Utami, T. K., Putri, K. A., Suryanto, S. O., & Asriani, F. (2025). Personal data breach cases in Indonesia: Perspective of Personal Data Protection Law. Journal Customary Law, 2(2), 21. https://doi.org/10.47134/jcl.v2i2.3742

Verizon. (2025). 2025 Data Breach Investigations Report. Verizon Business.

Wahyuriyanto, A. G., & Yulianton, H. (2024). Implementasi sistem monitoring jaringan berbasis Simple Network Management Protocol dengan notifikasi Telegram.