PENERAPAN APLIKASI *MOBILE SIEM* BERBASIS WAZUH UNTUK NOTIFIKASI *REAL-TIME* DAN REKOMENDASI PENANGANAN AWAL

Ilman Sholihin, Nurmi Hidayasari, Mansur
Teknik Informatika, Politeknik Negeri Bengkalis
Jalan Bathin Alam Desa Sungai Alam, Kab. Bengkalis, Indonesia
ilmansholihin1@gmail.com

ABSTRAK

Keamanan jaringan merupakan aspek vital dalam menjaga stabilitas sistem informasi di era digital. Sistem *Security Information and Event Management* (SIEM) seperti Wazuh telah banyak digunakan untuk mendeteksi ancaman. Namun, sistem ini memiliki keterbatasan pada aspek mobilitas karena *dashboard* hanya tersedia berbasis web dan notifikasi bergantung pada platform pihak ketiga, menyebabkan administrator kesulitan merespons insiden keamanan dengan cepat. Penelitian ini bertujuan mengembangkan aplikasi *mobile* SIEM berbasis Wazuh yang mampu memberikan notifikasi keamanan jaringan secara *real-time* serta rekomendasi penanganan awal terhadap ancaman yang terdeteksi. Metode penelitian menggunakan pendekatan eksperimental dengan integrasi Wazuh API, pengembangan *middleware* berbasis Python-Flask, implementasi *Firebase Cloud Messaging* (FCM), dan pengembangan aplikasi *mobile* Flutter. Pengujian dilakukan dengan tiga skenario serangan (*SQL Injection*, *XSS*, dan *Web Directory Scanning*) dengan masing-masing 5 kali percobaan. Hasil penelitian menunjukkan sistem mampu mendeteksi serangan dengan rata-rata *Detection Latency* 1,69 detik, *Notification Latency* 0,82 detik, dan *End-to-End Latency* 2,51 detik. Sistem berhasil menampilkan notifikasi keamanan secara *real-time* melalui perangkat *mobile* serta menyediakan rekomendasi penanganan awal yang dapat digunakan sebagai panduan mitigasi oleh administrator jaringan.

Kata kunci : *SIEM, Wazuh, Mobile Security, Real-time Notification, Flutter, Network Security*

1. PENDAHULUAN

Dalam era digital saat ini, keamanan jaringan merupakan aspek vital dalam menjaga stabilitas dan keberlangsungan sistem informasi pada berbagai organisasi. Serangan siber seperti *Distributed Denial-of-Service* (DDoS), *malware*, dan *brute-force attack* dapat menyebabkan gangguan serius terhadap ketersediaan layanan dan kebocoran informasi sensitif [1]. Data dari *Chainalysis* melaporkan bahwa total pembayaran *ransomware* mencapai 813 juta USD pada tahun 2024, menunjukkan bahwa serangan menjadi semakin masif dan variatif [2]. *Security Information and Event Management* (SIEM) merupakan pendekatan yang digunakan untuk mengintegrasikan pemantauan keamanan dari berbagai sumber log dan memberikan analisis insiden secara *real-time* [3]. Wazuh sebagai platform SIEM *open-source* menawarkan fitur analisis log, deteksi ancaman, dan *active response* yang powerful [4].

Meskipun Wazuh telah terbukti efektif dalam mendeteksi ancaman keamanan, sistem ini masih memiliki keterbatasan signifikan pada aspek mobilitas. *Dashboard* pemantauan Wazuh hanya tersedia berbasis web yang mengharuskan administrator mengakses komputer atau laptop untuk memantau kondisi keamanan jaringan [5]. Penelitian sebelumnya menunjukkan bahwa Wazuh efektif dalam mendeteksi ratusan *alert* dalam hitungan menit, namun hambatan utama terletak pada bagaimana informasi tersebut diterima dan ditindaklanjuti secara cepat oleh administrator [4].

Penelitian terkini menunjukkan implementasi prototipe *SIEM* berbasis Wazuh pada website dengan pengujian *File Integrity Monitoring (FIM)* dan *threat hunting* mampu mendeteksi *XSS* melalui rule 31105 dan *SQL injection* melalui rule 31106, namun masih menggunakan *dashboard* web bawaan Wazuh untuk visualisasi [6]. Penelitian lain menerapkan sistem deteksi dini serangan siber menggunakan Wazuh yang terintegrasi dengan *dashboard* kustom dan notifikasi Telegram, yang terbukti mampu mendeteksi serangan *DDoS*, *brute force*, dan *SQL injection* secara *real-time*, namun administrator tetap bergantung pada platform pihak ketiga untuk menerima notifikasi [7]. Keterbatasan ini menjadi kendala bagi pengelola TI yang membutuhkan respons cepat terhadap insiden keamanan, terutama ketika tidak berada di depan *dashboard* monitoring.

Penelitian ini bertujuan mengembangkan aplikasi *mobile-native* berbasis *Flutter* yang terintegrasi langsung dengan sistem Wazuh untuk memberikan notifikasi keamanan jaringan secara *real-time* serta rekomendasi penanganan awal terhadap ancaman yang terdeteksi. Aplikasi dirancang untuk memberikan kemudahan akses dalam melakukan pemantauan visual terhadap serangan dan menerima notifikasi keamanan tanpa ketergantungan pada platform eksternal.

Untuk mencapai tujuan tersebut, penelitian ini menggunakan metode eksperimental dengan pendekatan kuantitatif melalui pengembangan sistem yang mengintegrasikan Wazuh dengan aplikasi mobile berbasis *Flutter* dan *Firebase Cloud Messaging* untuk notifikasi *real-time*. Pengujian dilakukan terhadap salinan website Jurusan Teknik Informatika Politeknik Negeri Bengkalis menggunakan tiga skenario

serangan untuk mengukur performa deteksi dan pengiriman notifikasi sistem.

2. TINJAUAN PUSTAKA

2.1. *Security Information and Event Management (SIEM)*

SIEM merupakan sistem yang menggabungkan fungsi *Security Information Management (SIM)* dan *Security Event Management (SEM)* untuk mengumpulkan, menganalisis, dan menampilkan log keamanan dari berbagai sumber secara terpusat [8]. SIEM bekerja dengan mengumpulkan log dari *firewall*, *server*, dan *endpoint*, melakukan korelasi event untuk mendeteksi pola serangan kompleks, menghasilkan *alert* saat ancaman terdeteksi, dan menyajikan visualisasi melalui *dashboard* interaktif

2.2. Arsitektur Wazuh

Wazuh merupakan platform *SIEM open-source* yang terdiri dari komponen *Wazuh Agent*, *Wazuh Manager*, *Wazuh API*, dan *Dashboard*. Agent memantau sistem dan mengirimkan log ke *Manager*, *Manager* menganalisis log dan menghasilkan *alert*, sedangkan *API* memungkinkan aplikasi eksternal mengambil data *alert* secara terprogram [4]. Wazuh mendukung fitur *File Integrity Monitoring (FIM)*, *Rootkit Detection*, dan *Vulnerability Assessment*.

2.3. *Mobile* Application dan *Real-time* Notification

Framework Flutter mendukung pengembangan aplikasi lintas platform dengan kinerja tinggi melalui pendekatan widget-*driven* dan kompilasi *ahead-of-time (AOT)* [9]. Notifikasi *real-time* merupakan proses penyampaian pesan peringatan yang terjadi segera setelah sistem mendeteksi ancaman. *Firebase Cloud messaging (FCM)* mampu mengirimkan pesan secara cepat, andal, dan lintas platform dengan tingkat keterlambatan yang sangat rendah [10].

2.4. Penelitian Terkait

Alanda dkk. membuktikan efektivitas Wazuh dalam memantau *server* dan mendeteksi anomali dengan menghasilkan lebih dari 1.700 *alert* dalam satu minggu, namun belum menyediakan mekanisme notifikasi langsung [4]. Paramaputra dkk. menunjukkan sistem Wazuh mampu mendeteksi serangan *DoS* dalam rata-rata 10,36 detik dan memblokir IP dalam 50,36 detik, namun masih berfokus pada pemblokiran otomatis tanpa integrasi *mobile* [11]. Damanik dan Anggraeni mengembangkan sistem *hybrid Wazuh-Suricata* yang efektif namun masih berbasis desktop [5]. Ismail dkk. mengintegrasikan Wazuh dengan *Retrieval-Augmented Generation* untuk rekomendasi cerdas, namun belum difokuskan untuk platform *mobile* [12].

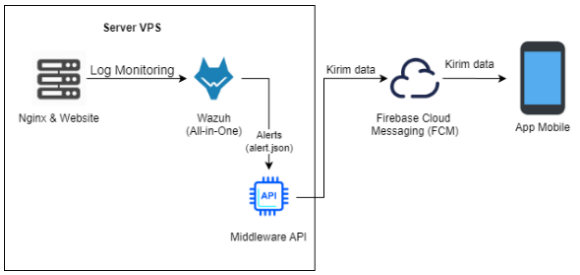
Berbeda dari penelitian sebelumnya, penelitian ini mengembangkan *SIEM* berbasis *mobile-native* yang terintegrasi langsung dengan Wazuh melalui *Middleware API*, mengirimkan notifikasi *real-time* disertai rekomendasi tindakan perbaikan dalam satu sistem terpadu.

3. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimental dengan pendekatan kuantitatif untuk mengukur performa sistem SIEM *mobile* dalam mendeteksi dan merespons ancaman jaringan secara *real-time*. Metode eksperimental merupakan metode untuk mengetahui pengaruh dari perlakuan tertentu dalam kondisi yang terkendali. Penelitian ini mengkaji efektivitas sistem dalam mendeteksi tiga jenis serangan siber yaitu *SQL Injection*, *Cross-Site Scripting (XSS)*, dan *Web Directory Scanning*.

3.1. Arsitektur Sistem

Sistem yang dikembangkan terdiri dari enam komponen utama yang saling terintegrasi untuk membentuk solusi pemantauan keamanan jaringan berbasis *mobile*. Komponen pertama adalah *Server VPS* yang menjalankan web *server* Nginx dan website sebagai sumber log aktivitas. Komponen kedua adalah *Wazuh (All-in-One)* yang berfungsi sebagai *Security Information and Event Management (SIEM)* untuk memantau log Nginx secara *real-time* dan menghasilkan *alert* dalam format JSON. Komponen ketiga adalah *Middleware API* yang terdiri dari *alert_reader.py* untuk membaca file *alerts.json* secara *real-time* dan *push_api.py* berbasis Flask untuk memproses *alert*, menentukan rekomendasi perbaikan, dan mengirimkan notifikasi. Komponen keempat adalah *Firebase Cloud messaging (FCM)* yang berperan sebagai perantara pengiriman notifikasi push ke perangkat *mobile*. Komponen kelima adalah Aplikasi *Mobile Flutter* yang menampilkan *dashboard* serangan, menerima notifikasi *real-time*, dan menampilkan rekomendasi perbaikan kepada administrator.



Gambar 1. arsitektur sistem siem mobile

Berdasarkan Gambar 1, dapat dijelaskan bahwa alur kerja sistem dimulai dari *Server* VPS yang menjalankan Nginx dan Website sebagai sumber log. Log yang dihasilkan dipantau secara *real-time* oleh *Wazuh (All-in-One)* melalui konfigurasi pemantauan log Nginx. Ketika Wazuh mendeteksi pola serangan seperti *SQL Injection*, *XSS*, atau *Web Directory Scanning* berdasarkan rule yang telah dikonfigurasi, sistem akan menghasilkan *alert* dalam format JSON yang disimpan di file *alerts.json*.

Alert tersebut kemudian dibaca oleh *Middleware API* melalui komponen *alert_reader.py* menggunakan teknik *file tailing*, sehingga setiap *alert* baru langsung terdeteksi tanpa delay. Setelah *alert* diterima, komponen *push_api.py* memproses data tersebut dengan mengekstrak informasi penting seperti rule ID, *severity level*, *timestamp*, dan *source IP*. *Middleware* juga melakukan severity *mapping* untuk mengkategorikan ancaman ke dalam level *Low*, *Medium*, *High*, atau *Critical* berdasarkan Wazuh rule level, serta mengambil rekomendasi perbaikan yang sesuai dari database *recommendations.yaml* berdasarkan rule ID yang terdeteksi.

Selanjutnya, *Middleware* mengirimkan notifikasi melalui *Firebase Cloud messaging (FCM)* yang berisi informasi serangan dan rekomendasi perbaikan. *FCM* kemudian meneruskan notifikasi tersebut ke aplikasi *mobile* pengguna secara *real-time*, bahkan ketika aplikasi berada di latar belakang. Aplikasi *mobile* akan menampilkan notifikasi push dan memperbarui *dashboard* dengan informasi serangan terbaru, sehingga administrator dapat segera melihat detail ancaman dan mengambil tindakan perbaikan yang direkomendasikan oleh sistem.

3.2. Lingkungan Pengujian

Penelitian dilakukan pada *Virtual Private Server (VPS)* dengan spesifikasi Ubuntu 22.04 LTS, RAM 8GB, Storage 100GB, Processor 2 vCPU. Perangkat lunak yang digunakan meliputi Wazuh v4.14 sebagai SIEM platform, Nginx v1.18.0 sebagai web *server*, Python 3.10.12 dengan Flask 3.1.2 untuk *middleware*, Firebase Admin SDK 7.1.0 untuk *push notification*, Flutter 3.32.4 dengan Dart 3.8.1 untuk *mobile* application, dan curl sebagai *HTTP client* untuk simulasi serangan. Seluruh perangkat disinkronisasi menggunakan timezone Asia/Jakarta (WIB, UTC+7) untuk memastikan konsistensi *timestamp*.

3.3. Desain Eksperimen

Desain eksperimen menggunakan dua variabel utama yaitu variabel bebas berupa jenis serangan (*SQL Injection*, *XSS*, *Web Directory Scanning*) dan variabel terikat berupa waktu respons sistem. Indikator pengukuran meliputi *Detection Latency* (LD) yang dihitung dengan rumus LD = T_*alert* - T_start, *Notification Latency* (LN) dengan rumus LN = T_app - T_*alert*, dan *End-to-End Latency* (LE2E) dengan rumus LE2E = T_app - T_start. Setiap jenis serangan dijalankan sebanyak 5 kali untuk memperoleh data yang konsisten.

3.4. Implementasi Sistem

Implementasi dilakukan dalam beberapa tahapan untuk membangun sistem *SIEM mobile* yang fungsional. Tahapan pertama adalah instalasi Wazuh mode all-in-one pada VPS Ubuntu 22.04 dengan konfigurasi pemantauan log Nginx melalui file *ossec.conf*. Konfigurasi ini mencakup penentuan lokasi file log yang akan dipantau, format log yang digunakan, dan rule deteksi yang akan diaktifkan untuk mengenali pola serangan.

Tahapan kedua adalah pengembangan *Middleware* menggunakan *Python* dengan dua komponen yaitu *alert_reader.py* yang membaca file *alerts.json* secara real-time menggunakan teknik file tailing, dan *push_api.py* yang menerima data *alert*, melakukan *severity mapping*, membaca *recommendations.yaml* untuk mendapatkan saran perbaikan, dan mengirim notifikasi melalui *FCM*.

Tahapan ketiga adalah pembuatan database rekomendasi dalam format *YAML* yang berisi mapping antara Wazuh rule ID dengan daftar saran perbaikan untuk setiap jenis serangan. Struktur *YAML* dipilih karena kemudahan dalam *maintenance* dan *readability*, sehingga administrator dapat dengan mudah menambahkan atau memodifikasi rekomendasi tanpa perlu memahami kode program secara mendalam. Setiap rekomendasi dirancang berdasarkan *best practice* keamanan siber dan disesuaikan dengan konteks serangan yang terdeteksi.

Tahapan keempat adalah pengembangan aplikasi *mobile* menggunakan *Flutter* dengan fitur *dashboard real-time*, *push notification handler*, detail *alert view* dengan rekomendasi, *timestamp* logging untuk pengukuran performa, dan *color-coded severity*.

3.5. Skenario Pengujian

Pengujian dilakukan dengan tiga skenario serangan terhadap website wazuhtipol.site yang dihosting pada VPS penelitian. Setiap jenis serangan dijalankan sebanyak 5 kali menggunakan *curl* sebagai *HTTP client*. Pengujian mencakup *SQL Injection* menggunakan *payload time-based*, *XSS* dengan lima varian script injection, dan *Web Directory Scanning* melalui batch request ke path sensitif. Setiap eksekusi serangan dicatat menggunakan tiga timestamp yaitu T_start saat serangan dimulai, T_alert dari file alerts.json saat Wazuh mendeteksi ancaman, dan T_app saat notifikasi diterima aplikasi mobile. Data timestamp ini digunakan untuk menghitung Detection Latency, Notification Latency, dan End-to-End Latency guna mengevaluasi kecepatan deteksi dan efektivitas pengiriman notifikasi sistem SIEM mobile.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem

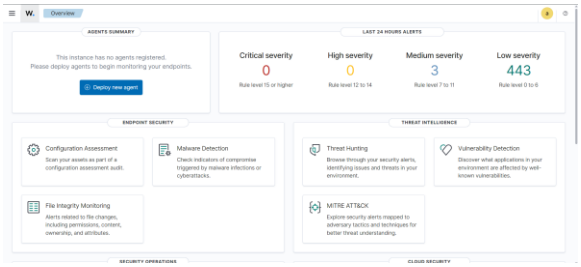
Sistem berhasil diimplementasikan pada VPS dengan spesifikasi Ubuntu 22.04 LTS, RAM 8GB, Storage 100GB, dan Processor 2 vCPU. Implementasi melibatkan konfigurasi beberapa komponen utama yang saling terintegrasi untuk membentuk sistem SIEM *mobile* yang fungsional. Proses implementasi dimulai dengan instalasi Wazuh versi 4.14 dalam mode all-in-one, dilanjutkan dengan konfigurasi pemantauan log Nginx, pengembangan *middleware* berbasis Python-Flask, integrasi dengan Firebase *Cloud messaging*, dan deployment aplikasi *mobile* menggunakan Flutter.

Tabel 1. Ringkasan Konfigurasi Sistem

Komponen	Versi/Spesifikasi	Status	Keterangan
Wazuh server	v4.14 All-in-One	Berhasil	Monitoring log Nginx via *ossec.conf*
Nginx Web *Server*	v1.18.0	Berhasil	Sumber log aktivitas website
Middleware API	Python 3.10.12, Flask 3.1.2	Berhasil	*alert_reader.py + push_api.py*
Firebase *Admin SDK*	v7.1.0	Berhasil	*Push notification* via *FCM*
Mobile Application	Flutter 3.32.4, Dart 3.8.1	Berhasil	Android deployment
Recommendations DB	YAML format	Berhasil	*Mapping* rule ID ke saran perbaikan

Dari Tabel 1 dapat dilihat bahwa seluruh komponen sistem berhasil dikonfigurasi dan berfungsi dengan baik. Konfigurasi Wazuh dilakukan dengan memodifikasi file *ossec.conf* untuk memantau log Nginx secara *real-time*, sehingga setiap aktivitas mencurigakan dapat segera terdeteksi.

Middleware API dikembangkan dengan dua komponen utama yaitu *alert_reader.py* yang berfungsi membaca file *alerts.json* menggunakan teknik *file tailing* untuk memastikan setiap *alert* baru langsung diproses tanpa delay, dan *push_api.py* yang bertugas memproses *alert*, melakukan severity *mapping*, mengambil rekomendasi dari *recommendations.yaml*, dan mengirimkan notifikasi melalui *FCM*. Integrasi dengan *Firebase Cloud messaging* berhasil dilakukan dan terbukti mampu mengirimkan notifikasi push secara *real-time* ke perangkat *mobile* meskipun aplikasi berada di latar belakang atau dalam kondisi tidak aktif.

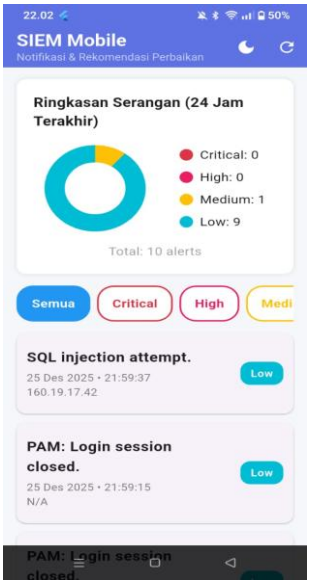


Gambar 2. Dashboard wazuh setelah konfigurasi

Gambar 2 menunjukkan *dashboard* Wazuh yang telah berhasil dikonfigurasi untuk memantau aktivitas keamanan pada web *server* Nginx. *Dashboard* menampilkan statistik *alert* yang terdeteksi, tingkat severity serangan, serta informasi tentang agent yang terhubung. Dari gambar tersebut dapat dilihat bahwa Wazuh mampu mengidentifikasi berbagai jenis ancaman berdasarkan rule yang telah dikonfigurasi, termasuk *SQL Injection*, *XSS*, dan *Web Directory Scanning*. Setiap *alert* yang muncul di *dashboard* secara otomatis dicatat dalam file *alerts.json* yang kemudian dibaca oleh *middleware* untuk diproses lebih lanjut.

4.2. Antarmuka Aplikasi *Mobile*

Aplikasi *mobile* dikembangkan menggunakan Flutter *framework* dengan bahasa pemrograman Dart untuk menciptakan antarmuka yang responsif dan user-friendly. Aplikasi dirancang dengan beberapa fitur utama yaitu *dashboard* yang menampilkan daftar serangan secara *real-time*, detail *alert* yang menampilkan informasi lengkap tentang ancaman beserta rekomendasi perbaikan, dan *push notification* yang memberikan peringatan ketika serangan terdeteksi.

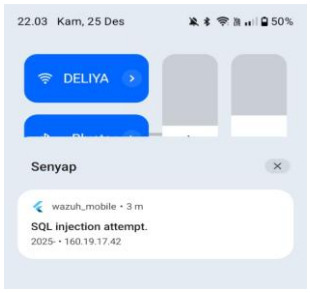


Gambar 3. Tampilan dashboard aplikasi mobile

Gambar 3 menampilkan halaman *dashboard* aplikasi *mobile* yang menjadi antarmuka utama bagi administrator untuk memantau ancaman keamanan. *Dashboard* menampilkan daftar serangan dalam bentuk card yang dilengkapi dengan *color-coded severity* untuk memudahkan identifikasi tingkat urgensi ancaman. Setiap card menampilkan informasi rule description, *timestamp* serangan, *source IP*, dan level severity dengan warna yang berbeda yaitu biru untuk Low, kuning untuk Medium, pink untuk High, dan merah untuk Critical. Desain ini memungkinkan administrator untuk dengan cepat mengidentifikasi serangan yang memerlukan penanganan prioritas tanpa harus membaca detail lengkap terlebih dahulu.

Gambar 4 memperlihatkan halaman detail *alert* yang muncul ketika administrator memilih salah satu serangan dari *dashboard*. Halaman ini menampilkan informasi lengkap tentang ancaman yang terdeteksi meliputi rule ID, *severity*, *timestamp*, *source IP*, dan rekomendasi perbaikan untuk jenis serangan tersebut. Rekomendasi disajikan dalam bentuk list yang mudah dipahami dan dapat langsung diimplementasikan oleh administrator. Fitur ini membedakan sistem yang dikembangkan dari penelitian sebelumnya yang hanya menampilkan *alert* tanpa memberikan panduan tindakan perbaikan.

Gambar 4. Tampilan detail alert aplikasi mobile

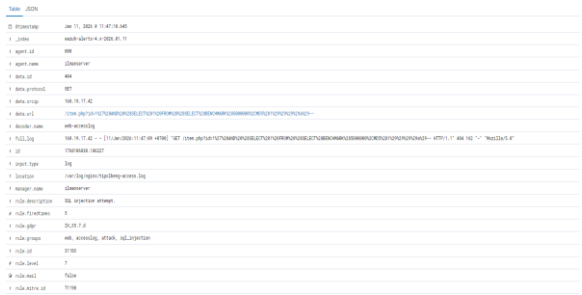


Gambar 5. Push notification real-time di aplikasi mobile

Gambar 5 menunjukkan *push notification* yang diterima oleh aplikasi *mobile* secara *real-time* ketika sistem mendeteksi ancaman. Notifikasi muncul langsung di layar perangkat Android bahkan ketika aplikasi berada di latar belakang atau tidak aktif. Setiap notifikasi memuat informasi ringkas tentang jenis serangan dan *source IP*. Ketika administrator mengetuk notifikasi, aplikasi akan langsung membuka halaman utama aplikasi *mobile*. Mekanisme *push notification* ini memastikan bahwa administrator dapat segera mengetahui dan merespons ancaman tanpa harus terus-menerus memantau *dashboard*, sehingga meningkatkan efektivitas respons keamanan jaringan.

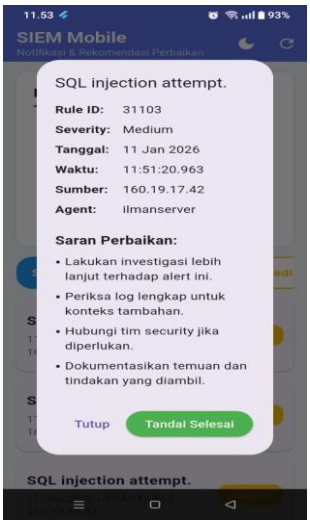
4.3. Pengujian Sistem

Pengujian dilakukan terhadap tiga jenis serangan dengan hasil deteksi yang berhasil dicatat oleh sistem Wazuh dan diteruskan ke aplikasi mobile. Setiap jenis serangan menghasilkan alert yang unik berdasarkan rule ID yang telah dikonfigurasi dalam Wazuh, dan sistem mampu memproses serta mengirimkan notifikasi dengan konsisten pada seluruh percobaan. Berikut adalah hasil pengujian untuk setiap jenis serangan:



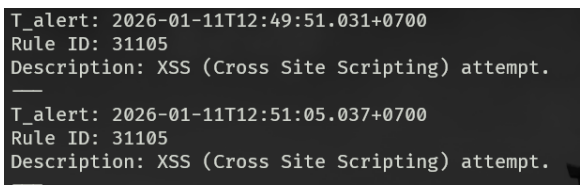
Gambar 6. Tampilan wazuh berhasil deteksi SQL Injection

Gambar 6 memperlihatkan *alert* yang dihasilkan oleh Wazuh ketika mendeteksi serangan *SQL Injection* dengan rule ID 31103. *Alert* menampilkan informasi detail termasuk *timestamp* deteksi, *source IP* penyerang, full log yang berisi *payload* serangan, dan tingkat severity. Dari gambar dapat dilihat bahwa Wazuh berhasil mengidentifikasi pola *SQL Injection* melalui analisis log Nginx yang mengandung string yang berbahaya. *Alert* ini kemudian disimpan dalam file *alerts.json* dan langsung dibaca oleh *middleware* untuk diproses.



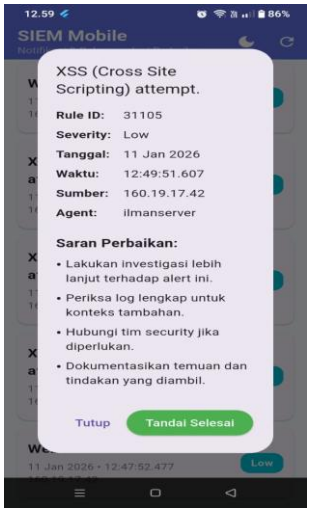
Gambar 7. Detail serangan SQLi pada aplikasi mobile

Gambar 7 menunjukkan detail serangan yang diterima aplikasi *mobile* untuk serangan *SQL Injection* yang sama dengan yang terdeteksi pada Gambar 6. Terdapat korespondensi langsung antara *alert* Wazuh dengan aplikasi *mobile*, menunjukkan bahwa sistem *middleware* berhasil memproses dan meneruskan informasi serangan dengan cepat dan akurat.



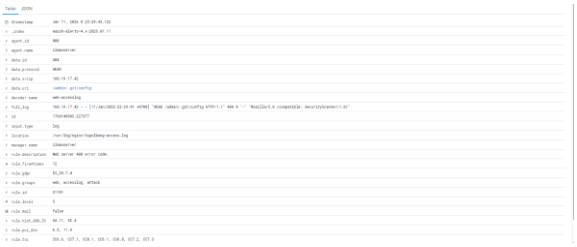
Gambar 8. wazuh deteksi serangan XSS

Gambar 8 memperlihatkan *alert* yang dihasilkan oleh Wazuh pada file *alert*.json ketika mendeteksi serangan *Cross-Site Scripting(XSS)* dengan rule ID 31105. Wazuh berhasil mengidentifikasi pola *XSS* melalui deteksi tag HTML berbahaya seperti script, img onerror, dan svg onload dalam log HTTP request.



Gambar 9. Detail serangan XSS `pada aplikasi mobile

Gambar 9 menunjukkan detail yang diterima aplikasi *mobile* untuk serangan *XSS*. Informasi detail tentang serangan, memungkinkan administrator untuk segera mengetahui adanya upaya *script injection* terhadap website.



Gambar 10. Tampilan wazuh berhasil deteksi WebDirScan

Gambar 10 memperlihatkan *alert* yang dihasilkan oleh Wazuh ketika mendeteksi aktivitas *Web Directory Scanning* dengan rule ID 31101 yang mengindikasikan "Web *server* 400 error code". *Alert* menampilkan upaya akses berulang ke path sensitif seperti /admin dan /.git/config yang menghasilkan HTTP error code 400 (Bad Request). Wazuh mengidentifikasi pola ini melalui analisis log akses yang menunjukkan permintaan HTTP berulang ke direktori yang seharusnya tidak dapat diakses publik atau tidak tersedia, yang merupakan karakteristik dari aktivitas reconnaissance penyerang. Tingkat severity dikategorikan sebagai rendah karena meskipun belum terjadi eksploitasi langsung, aktivitas ini mengindikasikan upaya pemetaan struktur direktori website sebagai tahap awal serangan yang lebih kompleks.

Gambar 11. Detail serangan pada aplikasi mobile

Gambar 11 menunjukkan notifikasi yang diterima aplikasi *mobile* untuk serangan *Web Directory Scanning*. Meskipun tingkat severity untuk jenis serangan ini umumnya lebih rendah dibanding *SQL Injection*, namun deteksi dini terhadap aktivitas scanning sangat penting karena merupakan fase awal dari serangan yang lebih kompleks.

Tabel 2. Hasil Pengukuran Waktu Respons Sistem

Jenis Serangan	Detection (s)	Notification (s)	End-to-End (s)
SQL Injection	1.88	**0.56**	2.45
XSS	1.68	**0.84**	2.52
Web Directory Scanning	1.50	**1.06**	2.57
Rata-rata	1.69	**0.82**	2.51

Dari Tabel 2 dapat dilihat bahwa sistem mampu mendeteksi serangan dengan rata-rata waktu respons (detection latency) sebesar 1,69 detik. *Web Directory Scanning* memiliki waktu respons tercepat, yaitu 1,50 detik, karena pola akses berulang lebih mudah diidentifikasi, sedangkan *SQL Injection* memiliki waktu respons paling lama, yaitu 1,88 detik, karena memerlukan analisis *payload* yang lebih kompleks. Selain itu, sistem mampu mengirimkan notifikasi ke perangkat *mobile* dengan rata-rata *Notification Latency* sebesar 0,82 detik, sedangkan rata-rata waktu end-to-end dari serangan terjadi hingga notifikasi diterima pengguna adalah 2,51 detik. Nilai *Notification Latency* yang relatif konsisten pada ketiga jenis serangan menunjukkan bahwa *middleware* dan *Firebase Cloud messaging (FCM)* bekerja dengan performa yang stabil dalam mendistribusikan notifikasi secara *real-time*.

4.4. Analisis Peforma

Hasil pengujian menunjukkan bahwa sistem mampu memberikan respons yang cepat dalam mendeteksi dan mengirimkan notifikasi ancaman jaringan secara *real-time*. *Detection Latency* tercepat

diperoleh pada *Web Directory Scanning*, yaitu sebesar 1,50 detik, karena pola akses berulang lebih mudah diidentifikasi. *SQL Injection* memiliki *Detection Latency* tertinggi, yaitu 1,88 detik, karena memerlukan analisis *payload* yang lebih kompleks.

Dari sisi pengiriman notifikasi, *Notification Latency* paling cepat terjadi pada serangan *SQL Injection* dengan nilai 0,56 detik, yang menunjukkan efisiensi *middleware* dalam memproses *alert* dengan rule ID tunggal dan mendistribusikannya ke perangkat *mobile*. Rata-rata *Notification Latency* untuk seluruh jenis serangan adalah 0,82 detik, menunjukkan bahwa proses pengiriman notifikasi melalui *middleware* dan *Firebase Cloud messaging (FCM)* berjalan stabil dan konsisten.

Dibandingkan dengan penelitian Paramaputra dkk. yang melaporkan waktu deteksi sebesar 10,36 detik pada serangan DoS, sistem yang dikembangkan memiliki *Detection Latency* rata-rata sebesar 1,69 detik, atau sekitar enam kali lebih cepat. Selain itu, rata-rata waktu end-to-end, yaitu dari serangan terjadi hingga notifikasi diterima oleh administrator, adalah 2,51 detik, yang menunjukkan bahwa sistem bersifat *real-time* dan responsif dalam mendukung proses penanganan insiden keamanan.

4.5. Fitur Rekomendasi Perbaikan

Sistem berhasil memberikan rekomendasi perbaikan untuk setiap jenis serangan berdasarkan rule ID yang terdeteksi oleh Wazuh. Rekomendasi disimpan dalam file *recommendations.yaml* yang berisi *mapping* antara Wazuh rule ID dengan daftar saran perbaikan yang telah dikurasi berdasarkan best practice keamanan siber.

Database rekomendasi menggunakan format YAML yang mudah dipelihara dan dapat diperluas sesuai kebutuhan. Administrator sistem dapat menambahkan rule baru atau memodifikasi rekomendasi yang ada tanpa perlu mengubah kode program. Setiap rekomendasi dirancang untuk dapat diimplementasikan oleh administrator sebagai panduan awal dalam proses mitigasi. Kelebihan pendekatan ini adalah rekomendasi yang diberikan bersifat kontekstual terhadap jenis serangan yang terdeteksi, sehingga administrator tidak perlu mencari solusi secara manual dari berbagai sumber dokumentasi. Sistem ini membedakan penelitian yang dikembangkan dari studi sebelumnya yang hanya berfokus pada deteksi ancaman tanpa memberikan panduan tindakan perbaikan.

Integrasi fitur rekomendasi dengan aplikasi *mobile* memungkinkan administrator untuk segera mengetahui langkah-langkah mitigasi yang perlu dilakukan setelah menerima notifikasi serangan. Hal ini mempercepat proses respons insiden keamanan dan mengurangi waktu yang dibutuhkan untuk mencari solusi yang tepat. Kelebihan sistem yang dikembangkan adalah kemampuan memberikan notifikasi *real-time* tanpa ketergantungan pada platform pihak ketiga seperti Telegram, rekomendasi perbaikan yang kontekstual untuk setiap jenis serangan, mobilitas tinggi melalui aplikasi *mobile* yang dapat diakses kapan saja dan di mana saja, dan integrasi langsung dengan Wazuh melalui API yang aman dan efisien tanpa memerlukan konfigurasi tambahan yang kompleks.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengembangkan sistem SIEM mobile berbasis Flutter yang terintegrasi dengan Wazuh untuk memberikan notifikasi *real-time* dan rekomendasi perbaikan terhadap ancaman jaringan. Sistem mampu mendeteksi tiga jenis serangan (*SQL Injection*, *XSS*, dan *Web Directory Scanning*) dengan rata-rata *Detection Latency* 1,69 detik, *Notification Latency* 0,82 detik, dan *End-to-End Latency* 2,51 detik. Aplikasi *mobile* berhasil menampilkan daftar serangan dengan *color-coded severity* dan memberikan rekomendasi perbaikan yang spesifik melalui integrasi Firebase *Cloud messaging*. Sistem ini memberikan solusi praktis bagi administrator jaringan untuk melakukan pemantauan keamanan secara *mobile* tanpa ketergantungan pada platform pihak ketiga.

Untuk penelitian selanjutnya, disarankan untuk mengintegrasikan teknologi AI untuk menghasilkan rekomendasi yang lebih dinamis dan kontekstual, menambahkan fitur automatic response untuk mitigasi otomatis terhadap ancaman tertentu, melakukan pengujian pada skala serangan yang lebih besar untuk mengevaluasi skalabilitas sistem, dan mengembangkan sistem analitik untuk memberikan insight tren keamanan berdasarkan historical data.

DAFTAR PUSTAKA

[1] D. Shankar, G. V. S. George, J. N. J. N. S. S, and P. S. Madhuri, “Deep Analysis of Risks and Recent Trends Towards Network Intrusion Detection System,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023, doi: 10.14569/IJACSA.2023.0140129.

[2] A. Breland, “Ransomware payments plunge after law enforcement actions,” 2025, *Axios*. [Online]. Available: https://www.axios.com/2025/02/07/chainalysis-ransomware-payments-hackers

[3] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, “Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures,” *Sensors*, vol. 21, no. 14, p. 4759, Jul. 2021, doi: 10.3390/s21144759.

[4] A. Alanda, H. A. Mooduto, and R. Hadi, “Real-time Defense Against Cyber Threats: Analyzing Wazuh’s Effectiveness in Server Monitoring,” *JITCE (Journal of Information Technology and Computer Engineering)*, vol. 7, no. 2, pp. 56–62, Sep. 2023, doi: 10.25077/jitce.7.2.56-62.2023.

[5] H. A. Damanik and M. Anggraeni, "Sistem Deteksi Intrusi Hybrid dan Mitigasi Kerentanan Infrastruktur Jaringan Menggunakan Teknik Active Response (XDR) Wazuh dan Suricata," *Jurnal Pekommas*, vol. 9, no. 2, pp. 309–322, Dec. 2024, doi: 10.56873/jpkm.v9i2.5829.

[6] N. Hidayasari, Mansur, Kasmawi, and Z. Efendi, "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, doi: 10.62527/jitsi.6.4.523.

[7] Haq Fatullazi Abdul, Mansur, and Hidayasari Nurmi, "DETEKSI DINI SERANGAN SIBER PADA SISTEM INFORMASI AKADEMIK BERBASIS WEB MENGGUNAKAN WAZUH," *Jurnal Studi Komputer dan Informatika*, vol. 9, no. 12, pp. 1–12, Dec. 2025.

[8] A. Anggraeni, J. G. A. Ginting, and S. Ikhwan, "Implementation of intrusion prevention system (IPS) to analysis triad cia on network security attacks on web server," *JURNAL INFOTEL*, vol. 14, no. 4, pp. 277–286, Nov. 2022, doi: 10.20895/infotel.v14i4.813.

[9] S. Stošović, D. Stefanović, M. Bogdanović, and N. Vukotić, "THE USE OF THE FLUTTER FRAMEWORK IN THE DEVELOPMENT PROCESS OF HYBRID MOBILE APPLICATIONS," *KNOWLEDGE - International Journal*, vol. 54, no. 3, pp. 477–483, Sep. 2022, doi: 10.35120/kij5403477s.

[10] A. Kristian, N. Nuryuliani, and A. R. Hakim, "ALERT SYSTEM DESIGN MENGGUNAKAN TEKNOLOGI FIREBASE CLOUD MESSAGING (FCM)," *Jurnal Ilmiah Multidisiplin*, vol. 3, no. 02, pp. 103–112, Mar. 2024, doi: 10.56127/jukim.v3i02.1601.

[11] A. P. Paramaputra, G. M. Suranegara, and E. Setyowati, "MITIGATION OF MULTI TARGET DENIAL OF SERVICE (DOS) ATTACKS USING WAZUH ACTIVE RESPONSE," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 7, no. 2, pp. 483–493, Apr. 2025, doi: 10.47709/cnahpc.v7i2.5755.

[12] 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, doi: 10.3390/s25030870