

Implementasi dan Analisis Sistem Keamanan Wazuh-Graylog di Diskominfo Cimahi

Taviv Sutisna¹, Fajar Gilang Trifebriyanto², Ratu Fayyaza Ayu Humayra³, Maya Rahayu⁴

^{1,2,3,4}Jurusan Teknik Elektro, Politeknik Negeri Bandung, Indonesia

¹tavivsutisna@polban.ac.id, ²fajar.gilang.tkom422@polban.ac.id, ³ratufayyazaah11@gmail.com,

⁴mayarahayu@polban.ac.id

Abstrak

Transformasi digital pada instansi pemerintahan menuntut standar keamanan siber yang tangguh. Namun, sistem seringkali luput dari pemantauan intensif, sehingga rentan terhadap eksploitasi yang dapat menurunkan kualitas pelayanan publik. Penelitian ini bertujuan untuk merancang dan menganalisis sistem keamanan terpadu berbasis *Security Information and Event Management* (SIEM) dan *Security Operations Center* (SOC). Sistem ini dibangun menggunakan platform *open-source* Wazuh dan Graylog di atas infrastruktur *Virtual Private Server* (VPS) guna memetakan kerentanan serta mengeksekusi mitigasi serangan secara otonom. Implementasi dilakukan pada peladen *development* dengan membaginya menjadi dua skema: pengawasan aset secara pasif pada server target di Dinas Komunikasi dan Informatika (Diskominfo) Kota Cimahi, serta simulasi serangan aktif di lingkungan laboratorium terisolasi. Metode pengujian komprehensif mencakup injeksi *ransomware*, serangan *brute force* pada *Secure Shell* (SSH), eksploitasi *Denial of Service* (DoS) menggunakan taktik *HTTP flood*, dan pemindaian kerentanan aplikasi web menggunakan *Dynamic Application Security Testing* (DAST). Parameter kinerja utama yang dievaluasi meliputi tingkat akurasi deteksi ancaman, waktu tunda mitigasi (*response time*), dan ketahanan sistem terhadap masifnya volume lalu lintas jaringan. Hasil pengujian membuktikan bahwa sistem terintegrasi ini berhasil mendeteksi anomali siber dengan akurasi tinggi. Modul *Active Response* mampu memblokir entitas penyerang secara otomatis dengan rata-rata waktu tunda di bawah lima detik. Sistem juga terbukti tangguh dalam memutus koneksi peretas dan menghapus berkas *ransomware* sebelum layanan peladen mengalami kelumpuhan akibat penumpukan antrean permintaan (*request queue*), sehingga sangat layak diterapkan sebagai lapisan pelindung infrastruktur pemerintahan.

Kata kunci: *Security Operations Center, Virtual Private Server, Deteksi Dini, Respon Otomatis*

1. Pendahuluan

Transformasi digital di sektor pemerintahan daerah membawa perubahan masif pada efisiensi layanan, namun turut memperluas area permukaan serangan (*attack surface*) yang rentan terhadap ancaman siber. Insiden kelumpuhan Pusat Data Nasional Sementara (PDNS) pada tahun 2024 dan sistem perbankan terdistribusi pada tahun 2023 akibat *ransomware* menunjukkan fatalnya ketiadaan kapabilitas deteksi dini (*early detection*) dan respon insiden pada jaringan infrastruktur kritis.

Implementasi *Security Information and Event Management* (SIEM) yang terintegrasi dengan *Security Operations Center* (SOC) menjadi kebutuhan krusial, namun adopsi platform komersial seringkali terkendala oleh tingginya biaya lisensi bagi instansi daerah.

Beberapa penelitian terdahulu telah mengeksplorasi pemanfaatan platform *open-source* untuk mengatasi kendala tersebut. Gilang Trifebriyanto *et al.* (2026) merancang sistem pemantauan keterse-

diaan peladen (*uptime*) instansi pemerintah menggunakan Uptime Kuma berbasis VPS, namun arsitektur tersebut masih sebatas pemantauan pasif tanpa mitigasi keamanan. Nas *et al.* (2023) menggunakan Wazuh sebagai SIEM untuk sentralisasi log di Diskominfo Sulawesi Selatan, tetapi sistem tersebut belum mengimplementasikan modul pertahanan aktif. Sementara itu, Farrel (2024) membuktikan efektivitas fitur *Active Response* Wazuh dalam memblokir serangan *brute force*, namun pengujiannya hanya dieksekusi pada jaringan area lokal (LAN) konvensional. Untuk mengelola telemetri yang masif, Khotimah *et al.* (2022) menegaskan bahwa agregasi log berbasis arsitektur *Elasticsearch* (seperti Graylog) mampu memberikan waktu tunda (*latency*) kueri pencarian yang sangat rendah.

Keterbatasan utama dari berbagai studi sebelumnya adalah pengujian arsitektur SIEM yang mayoritas masih mengandalkan dataset sekunder statis yang kurang merepresentasikan lalu lintas jaringan *cloud* asinkron. Selain itu, belum ada integrasi per-

tahanan otonom multivektor yang komprehensif pada infrastruktur peladen pemerintahan. Mengatasi celah tersebut, penelitian ini berkontribusi untuk merancang bangun sistem keamanan hibrida SIEM-SOC menggunakan Wazuh dan Graylog secara terpusat pada *Virtual Private Server* (VPS).

Sebagai pembeda utama dan kebaruan (*novelty*) dari penelitian sebelumnya, penelitian ini tidak sekadar melakukan instalasi standar, melainkan merancang arsitektur kustom *Hybrid-SIEM* yang mengkombinasikan peringatan deteksi dari agen (*edge-level*) dengan agregasi pencarian log berlatensi rendah di Graylog (*core-level*). Jika dibandingkan dengan pendekatan SIEM tunggal komersial seperti Splunk atau arsitektur dasar ELK Stack yang cenderung memonopoli penggunaan *resource* komputasi lokal, arsitektur terdistribusi Wazuh-Graylog berbasis VPS pada penelitian ini dirancang spesifik agar agen pemantau di titik akhir (seperti pada peladen Diskominfo) beroperasi sangat ringan (*lightweight*), memindahkan seluruh beban komputasi analisis (*analysis engine*) murni ke *cloud*. Pendekatan ini mengatasi masalah skalabilitas yang sering dialami instansi daerah saat mengelola ribuan log harian.

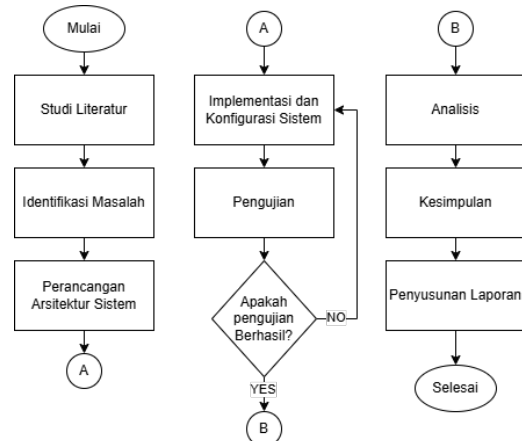
Penelitian ini dilakukan dengan pendekatan eksperimental menggunakan data primer di dalam laboratorium virtual yang terisolasi. Fokus utama penelitian adalah mengevaluasi keandalan audit pasif pada *Server Development* Diskominfo Kota Cimahi, sekaligus mengukur efisiensi *Active Response* dalam memitigasi serangan agresif multivektor, meliputi *SSH Brute Force*, injeksi *ransomware*, *HTTP Flood Denial of Service* (DoS), dan pemindaian *Dynamic Application Security Testing* (DAST) menggunakan OWASP ZAP. Melalui sistem ini, metrik akurasi deteksi, keandalan penanganan lonjakan lalu lintas jaringan, dan waktu tunda mitigasi dapat terukur secara presisi, yang hasilnya langsung ditransmisikan sebagai peringatan *real-time* kepada administrator melalui bot Telegram.

2. Metode

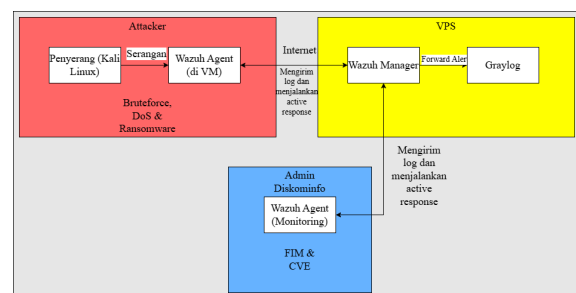
Penelitian ini menerapkan metode kuantitatif eksperimental yang dipadukan dengan studi kasus lapangan. Subjek penelitian berfokus pada infrastruktur *Server Development* milik Dinas Komunikasi dan Informatika (Diskominfo) Kota Cimahi. Pengumpulan data dilakukan melalui dua pendekatan utama, yakni akuisisi data sekunder berupa log audit pasif dari *server* Diskominfo, dan data primer melalui simulasi serangan multivektor di dalam lingkungan laboratorium virtual yang terisolasi.

2.1 Tahapan dan Alur Penelitian

Penyelesaian penelitian ini menempuh langkah-langkah terstruktur yang diawali dari tahap perencanaan hingga evaluasi akhir, sebagaimana diilustrasikan pada Gambar 1.



Gambar 1. Diagram Alur Penelitian



Gambar 2. Diagram Blok Rancangan Sistem

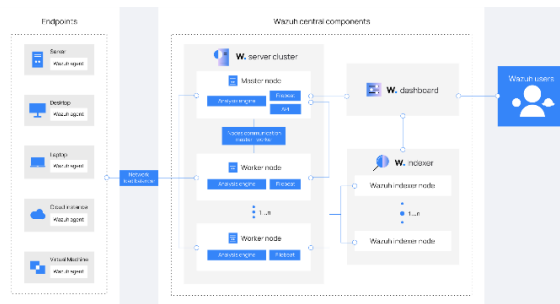
trasikan pada Gambar 1.

Tahapan tersebut dimulai dengan studi literatur dan identifikasi masalah untuk menganalisis urgensi pengamanan *server development* instansi pemerintah serta mengkaji literatur terkait arsitektur *Security Information and Event Management* (SIEM). Selanjutnya, dilakukan perancangan arsitektur sistem dengan menyusun desain topologi *Hybrid Cloud* yang memisahkan zona pusat komando (VPS), zona laboratorium virtual, dan zona implementasi lapangan. Langkah berikutnya adalah implementasi dan konfigurasi, yaitu menyiapkan peladen VPS serta melakukan instalasi dan integrasi *Wazuh Manager* beserta *Graylog Server* sebagai pusat agregasi log. Setelah itu, pengujian sistem dilaksanakan dengan melakukan pemantauan pasif dan peluncuran simulasi serangan agresif untuk menguji keandalan deteksi dan respon otomatis (*Active Response*). Penelitian ini diakhiri dengan tahap analisis dan kesimpulan untuk mengevaluasi metrik kinerja sistem berupa akurasi deteksi, waktu tunda mitigasi (*response time*), dan keandalan sistem dalam menangani lonjakan beban lalu lintas log jaringan.

2.2 Rancangan Sistem

Arsitektur keamanan yang dibangun terdistribusi ke dalam tiga zona lingkungan utama yang terintegrasi secara asinkron, seperti yang ditunjukkan pada Gambar 2.

Zona pertama bertindak sebagai pusat komando VPS yang menampung *Wazuh Manager* untuk anal-



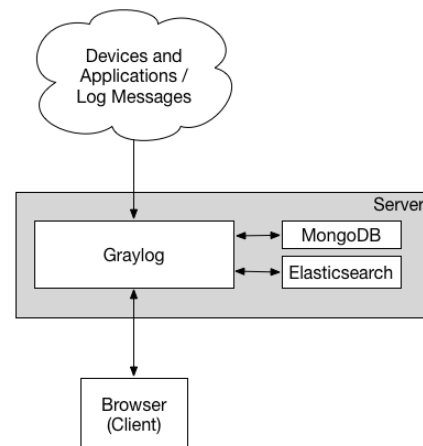
Gambar 3. Alur Subsistem Wazuh

isis log berdasarkan aturan keamanan (*ruleset*), serta platform *Graylog* sebagai antarmuka visualisasi yang terhubung melalui jalur *data pipeline*. Zona kedua merupakan lingkungan *sandbox* laboratorium virtual yang difungsikan untuk pengujian pertahanan aktif, yang di dalamnya terdapat mesin penyerang berbasis Kali Linux dan mesin target yang dipasang agen *Wazuh*. Sementara itu, zona ketiga adalah lingkungan implementasi nyata pada *Server Development* Diskominfo Kota Cimahi. Pada zona ketiga ini, agen keamanan beroperasi dalam mode audit pasif (*read-only*) yang difokuskan pada pengawasan *File Integrity Monitoring* (FIM) dan pemindaian kerentanan *Common Vulnerabilities and Exposures* (CVE).

Secara lebih spesifik, operasional pada subsistem *Wazuh* mengandalkan interaksi yang terenkripsi antara titik akhir (*endpoints*) dan komponen pusat, sebagaimana divisualisasikan pada Gambar 3.

Titik akhir yang dipasang agen bertugas mengumpulkan log aktivitas, memantau perubahan berkas, memindai kerentanan, dan mengeksekusi modul pertahanan aktif. Data telemetri tersebut kemudian dikirimkan menuju klaster peladen pusat, di mana mesin analisis (*Analysis Engine*) pada simpul utama (*Master Node*) melakukan pembongkaran data (*decoding*) dan pencocokan aturan keamanan (*ruleset matching*). Apabila anomali terdeteksi, sistem membangkitkan peringatan yang kemudian diteruskan oleh utilitas kurir log *Filebeat* menuju subsistem *Graylog*.

Aliran data (*data pipeline*) selanjutnya bermuara pada subsistem *Graylog* yang diilustrasikan pada Gambar 4. *Graylog Server* berperan sebagai pengumpul log tingkat lanjut yang menerima aliran pesan berformat JSON dari *Wazuh*. Setelah pesan log diurai, rute penyimpanannya didukung oleh dua basis data utama: *MongoDB* untuk menyimpan metadata operasional seperti tata letak dasbor, serta *Elasticsearch* yang berfungsi sebagai mesin pencari dan pengindeksan data (*search and indexing engine*). Arsitektur ini memungkinkan pencarian log historis berkapasitas besar dengan latensi waktu tunda yang sangat rendah dan memvisualisasikannya melalui antarmuka klien peramban (*browser client*) bagi administrator *Security Operations Center* (SOC).



Gambar 4. Alur Subsistem Graylog

2.3 Skenario Pengujian

Guna mendapatkan data kinerja yang presisi, pengujian dibagi menjadi skema pemantauan pasif di lapangan dan simulasi serangan di laboratorium. Simulasi serangan pada lingkungan laboratorium difokuskan pada empat vektor ancaman yang mewakili berbagai eskalasi siber. Pengujian diawali dengan pemindaian kerentanan otomatis menggunakan alat *Dynamic Application Security Testing* (DAST) *OWASP ZAP* untuk mengevaluasi kecepatan agen dalam membaca anomali pada log akses peladen web. Selanjutnya, sistem diuji menggunakan injeksi *ransomware* melalui eksekusi skrip perulangan yang mengenkripsi kumpulan berkas secara paksa, dengan tujuan menguji ketajaman modul pemantauan integritas berkas (FIM). Evaluasi juga mencakup simulasi intrusi akses jaringan melalui *SSH Brute force* menggunakan kamus kata sandi secara berulang untuk mengukur latensi pemutusan koneksi peretas oleh utilitas *iptables*. Terakhir, pengujian dilakukan dengan mengeksplorasi peladen web menggunakan lalu lintas *HTTP Flood* untuk mengevaluasi kemampuan peladen dalam memproses lonjakan paket secara bersamaan, serta memastikan antrean layanan tidak mengalami kelumpuhan (*service timeout*) sebelum mekanisme perlindungan aktif berhasil tereksekusi.

3. Hasil dan Pembahasan

Bagian ini memaparkan hasil pengujian fungsionalitas dan analisis kinerja arsitektur SIEM-SOC (*Wazuh-Graylog*) yang telah diimplementasikan pada infrastruktur *Virtual Private Server* (VPS). Pembahasan pada bab ini mencakup evaluasi audit keamanan pasif pada peladen *development* Dinas Komunikasi dan Informatika (Diskominfo) Kota Cimahi, serta analisis komprehensif terkait metrik respons sistem dalam memitigasi berbagai simulasi serangan siber aktif di lingkungan laboratorium. Lebih lanjut, bagian ini juga menguraikan implikasi temuan

Tabel 1. Hasil Pengujian Serangan *Ransomware*

Pengujian ke-	Total Injeksi	Berhasil Terdeteksi	Waktu Tunda
1	10 Aksi	10	2.40 detik
2	10 Aksi	10	2.85 detik
3	10 Aksi	10	2.55 detik
4	10 Aksi	10	3.10 detik
5	10 Aksi	10	2.60 detik
Rata-rata	10 Aksi	10 (0 false negatif)	2.70 detik

penelitian, perbandingan efektivitas mitigasi dengan studi terdahulu, serta identifikasi keterbatasan sistem berdasarkan pengujian *confusion matrix*.

3.1 Hasil Pengujian *Ransomware*

Pengujian simulasi serangan *ransomware* dieksekusi menggunakan skrip perulangan (*loop script*) pada lingkungan target untuk memanipulasi dan mengenkripsi sekumpulan berkas secara paksa. Skrip tersebut secara agresif menimpa isi dokumen asli dan mengubah ekstensi berkas menjadi *.locked*, yang secara langsung merepresentasikan perilaku destruktif perangkat perusak (*malware*). Tindakan manipulasi ini secara otomatis memicu sensor *File Integrity Monitoring* (FIM) pada agen Wazuh untuk mengirimkan log peringatan menuju peladen pusat.

Hasil rekaman pengujian yang diulang sebanyak lima kali iterasi disajikan pada Tabel 1.

Berdasarkan Tabel 1, dari total 50 aksi injeksi yang dilakukan pada 25 berkas target, agen Wazuh sukses mendeteksi seluruh anomali penghapusan dan penambahan berkas (*File deleted/added*). Rata-rata waktu tunda (*response time*) yang dibutuhkan oleh agen sejak berkas mulai dienkripsi hingga peringatan keamanan muncul di dasbor pusat adalah sebesar 2,70 detik. Waktu tunda ini konsisten berada di bawah ambang batas toleransi mitigasi (*Service Level Agreement* / SLA internal) yang ditetapkan pada penelitian ini, yakni kurang dari 5 detik. Batasan ini ditetapkan sebagai target respons kritis guna memastikan rantai serangan diputus sebelum peretas berhasil menimbulkan kerusakan struktural atau mengekstraksi data dalam jumlah masif.

Kecepatan deteksi yang presisi ini dikarenakan modul FIM pada agen tidak sekadar mengandalkan pemindaian berkala (*scheduled scan*), melainkan dikonfigurasi secara spesifik untuk terintegrasi dengan fungsi *inotify* pada tingkat *kernel* sistem operasi Linux. Mekanisme *inotify* bekerja secara reaktif untuk memantau perubahan atribut berkas secara seketika (*real-time*). Ketika skrip *ransomware* memodifikasi berkas, *inotify* secara instan memicu agen untuk menghitung ulang nilai acakan kriptografi (*hash*) dan melaporkannya ke *server* VPS, sehingga latensi deteksi dapat ditekan secara maksimal. Selain itu, notifikasi deteksi juga berhasil ditransmisikan secara in-

Tabel 2. Hasil Pengujian Serangan Bruteforce

Pengujian ke-	Pemblokiran pada Request ke-	Waktu Tunda
1	13	4 detik
2	14	5 detik
3	15	4 detik
4	16	3 detik
5	17	2 detik
Rata-rata	15	3.6 detik

stan kepada administrator melalui integrasi bot Telegram.

3.2 Hasil Pengujian Bruteforce

Pengujian simulasi serangan intrusi *Secure Shell* (SSH) *brute force* dieksekusi dengan menembakkan kombinasi kata sandi secara paksa dan berulang-ulang menuju *port* 22 pada peladen target menggunakan utilitas *Hydra*. Serangan kamus kata (*dictionary attack*) ini secara agresif mengirimkan permintaan masuk (*login request*) secara bertubi-tubi, yang secara otomatis memicu lonjakan log kegagalan autentikasi pada sistem. Pengujian ini bertujuan untuk mengevaluasi keandalan mesin korelasi dalam mengidentifikasi anomali serta mengukur kecepatan eksekusi modul *Active Response* dalam menjatuhkan koneksi peretas.

Hasil pengujian yang dilakukan sebanyak lima kali iterasi dirangkum pada Tabel 2.

Berdasarkan Tabel 2, sistem berhasil memblokir alamat IP penyerang pada setiap iterasi pengujian. Rata-rata pemblokiran terjadi pada percobaan permintaan masuk ke-15, dengan rata-rata waktu tunda (*response time*) sebesar 3,6 detik sejak serangan pertama kali diluncurkan hingga alamat IP penyerang dimasukkan ke dalam daftar hitam (*blacklist*).

Waktu tunda tersebut terjadi karena modul *Active Response* tidak langsung memblokir alamat IP pada saat percobaan masuk gagal yang pertama. Mesin analisis korelasi log (*analysis engine*) pada *Wazuh Manager* dikonfigurasi untuk memantau frekuensi log autentikasi, dan pemblokiran baru akan dieksekusi secara otonom di lapisan *kernel* target (menggunakan *iptables*) apabila sistem mendeteksi akumulasi kegagalan autentikasi yang menembus ambang batas aturan (*frequency ruleset*), yakni sekitar 6 hingga 8 kegagalan dalam kurun waktu singkat. Selama proses pengumpulan log, transmisi jaringan ke manajer, hingga pengiriman perintah blokir kembali ke agen target berlangsung (sekitar 3,6 detik), utilitas penyerang yang bekerja secara asinkron sempat meloloskan beberapa permintaan tambahan sehingga pemblokiran baru efektif pada kisaran *request* ke-13 hingga ke-17. Meskipun demikian, waktu tunda tersebut terbukti sangat optimal dalam mencegah peretas mendapatkan hak akses peladen yang sesungguhnya.

Tabel 3. Hasil Pengujian Serangan DoS

Pengujian ke-	Total Permintaan	Permintaan Lolos	Permintaan Digugurkan (Drop)
1	100,000	18,464	81,536
2	100,000	21,192	78,808
3	100,000	30,733	69,267
4	100,000	5,472	94,528
5	100,000	7,412	92,588
Rata-rata	100,000	16,654	83,345

3.3 Hasil Pengujian DoS

Pengujian simulasi serangan lapisan aplikasi berupa *HTTP Flood Denial of Service (DoS)* dieksekusi dengan membombardir peladen web menggunakan lalu lintas permintaan HTTP secara serentak. Pengujian ini memanfaatkan utilitas *ApacheBench* untuk mengirimkan 100.000 paket permintaan dengan tingkat konkurensi 1.000 tembakan bersamaan.

Rincian rekaman pengujian yang diulang sebanyak lima kali iterasi dirangkum pada Tabel 3.

Berdasarkan Tabel 3, dari total 100.000 tembakan per iterasi, sistem secara otonom berhasil menggugurkan rata-rata 83.345 permintaan lalu lintas berbahaya. Sebaliknya, rata-rata permintaan yang sempat lolos masuk ke dalam peladen sebelum pemblokiran terjadi adalah sebesar 16.654 permintaan. Terdapat fluktuasi jumlah *request* yang lolos antar iterasi; pengujian ketiga meloloskan 30.733 permintaan dengan waktu tunda mitigasi 10 detik, sementara pengujian keempat hanya meloloskan 5.472 permintaan dengan waktu tunda 4 detik.

Fluktuasi pengeksekusian mitigasi oleh agen Wazuh ini tidak terlepas dari masifnya pengiriman paket asinkron dalam hitungan milidetik. Ketika utilitas *ApacheBench* menembakkan gelombang permintaan *HTTP GET* secara bersamaan, antrian pembacaan log lokal dan transmisi telemetri menuju peladen VPS pusat sempat mengalami penumpukan antrian lalu lintas jaringan (*network queue*). Rentang keterlambatan transmisi selama beberapa detik inilah yang memberikan celah bagi masuknya ribuan permintaan HTTP awal. Meskipun waktu tunda eksekusi sempat berfluktuasi akibat masifnya beban trafik jaringan, mekanisme perlindungan *Active Response* terbukti tetap berhasil dieksekusi secara otomatis guna memutus rantai serangan DoS.

3.4 Hasil Pengujian Pemindaian DAST (Zaproxy)

Pengujian simulasi ancaman pada lapisan aplikasi dieksekusi menggunakan perangkat lunak OWASP ZAP (*Zed Attack Proxy*). Pengujian ini meniru perilaku peretas yang meluncurkan pemindaian kerentanan (*vulnerability scanning*) secara otomatis dengan menembakkan berbagai muatan eksploitasi ke arah peladen web target. Tujuan pengujian ini

Tabel 4. Hasil Pengujian Pemindaian OWASP ZAP

Pengujian ke-	Waktu Tunda	Status Pemblokiran
1	4 detik	Berhasil
2	2 detik	Berhasil
3	2 detik	Berhasil
4	1 detik	Berhasil
5	4 detik	Berhasil
Rata-rata	2.6 detik	Berhasil

adalah untuk mengevaluasi kecepatan agen keamanan dalam membaca anomali pada log akses peladen web dan memutus akses proksi penyerang.

Hasil rekaman pengujian pemindaian yang diulang sebanyak lima kali iterasi dirangkum pada Tabel 4.

Berdasarkan Tabel 4, sistem berhasil menggagalkan seluruh iterasi pemindaian kerentanan yang diluncurkan oleh mesin penyerang.

Secara keseluruhan, rata-rata latensi waktu pemblokiran adalah sebesar 2,6 detik. Waktu tunda mitigasi ini juga telah memenuhi standar *Service Level Agreement (SLA internal)* yang membatasi toleransi respons maksimal di angka 5 detik. Justifikasi batasan SLA ini sangat krusial pada serangan lapisan aplikasi guna memastikan pemutusan akses alamat IP perangkat penyerang (*timeout*) tereksekusi sebelum alat pemindai (ZAP) mendapatkan cukup waktu untuk merayapi (*crawling*) dan memetakan seluruh celah kerentanan secara mendalam pada peladen target.

Respons mitigasi yang sangat cepat ini dapat dicapai karena agen Wazuh dikonfigurasi untuk membaca dan menganalisis berkas log akses peladen web secara seketika (*real-time*). Ketika OWASP ZAP menembakkan ribuan injeksi *payload* yang tidak valid (seperti simulasi *SQL Injection* atau penelusuran direktori paksa), peladen web target secara otomatis menolak format tersebut dan meresponsnya dengan kode galat, seperti *error code 400*. Mesin korelasi Wazuh langsung menangkap lonjakan log galat tersebut sebagai anomali pemindaian berbahaya. Arsitektur pemantauan yang terhubung langsung dengan log layanan aplikasi ini memungkinkan modul *Active Response* mengeksekusi pemblokiran alamat IP dengan sangat responsif, sehingga perangkat pemindai ZAP langsung kehilangan koneksi (*timeout*) sebelum berhasil memetakan seluruh celah kerentanan pada peladen target.

3.5 Hasil Penilaian Konfigurasi Keamanan (*Security Configuration Assessment*)

Pengujian postur keamanan dilakukan menggunakan modul *Security Configuration Assessment (SCA)* pada agen Wazuh untuk memetakan profil keamanan peladen instansi target secara pasif. Proses audit dicocokkan terhadap standar kebijakan *CIS (Cen-*

Tabel 5. Hasil Pengujian *Confusion Matrix*

Parameter	Skenario Deteksi	Jumlah
<i>True Positive</i> (TP)	Serangan (Ransomware/Brute force) yang sukses diblokir	48
<i>True Negative</i> (TN)	Aktivitas normal yang sukses dibiarkan system	45
<i>False Positive</i> (FP)	Skrip <i>backup</i> administrator yang keliru diblokir	5
<i>False Negative</i> (FN)	Serangan <i>backdoor low-and-slow</i> yang lolos deteksi	2

ter for Internet Security) Benchmark.

Hasil evaluasi menunjukkan adanya sejumlah indikator konfigurasi sistem peladen yang belum memenuhi standar keamanan penuh. Namun, untuk alasan keamanan dan anonimisasi infrastruktur riil instansi, persentase metrik kepatuhan (*compliance score*) serta jumlah celah spesifik pada Diskominfo Kota Cimahi tidak dipublikasikan secara mendetail di dalam naskah ini. Kerentanan yang teridentifikasi secara umum didominasi oleh kelemahan pada kebijakan kontrol akses pengelola (seperti ketiadaan persyaratan kompleksitas kata sandi) dan tidak aktifnya sejumlah aturan pemfilteran pada *firewall* bawaan. Perlu ditekankan bahwa seluruh temuan kerentanan hasil audit pasif ini telah dilaporkan kepada pihak administrator instansi terkait, dan proses remediasi (*patching*) serta penguatan sistem (*hardening*) telah berstatus selesai diimplementasikan saat penelitian ini ditulis.

3.6 Hasil Pengujian *Confusion Matrix*

Meskipun arsitektur Wazuh-Graylog terbukti tangguh memitigasi vektor serangan, sistem tetap diuji untuk memetakan sejumlah keterbatasan (*blind spot*). Evaluasi ini dilakukan menggunakan metrik *confusion matrix* kuantitatif yang divalidasi melalui 100 sampel skenario log (50 log serangan aktif dan 50 log aktivitas normal/sah). Rangkuman metrik kuantitatif disajikan pada Tabel 5.

Berdasarkan Tabel 5, mesin korelasi menghasilkan nilai Akurasi sistem sebesar 93%. *False Positive* memicu nilai Presisi sebesar 90,5%. Secara teknis, positif palsu ini terjadi karena aturan pemantauan integritas berkas (FIM) secara bawaan mendeteksi perubahan atau penghapusan berkas massal dalam interval waktu sempit sebagai anomali *ransomware*. Saat skrip *backup* administrator menghapus log lama secara instan, ambang batas heuristik ini terlampaui. Strategi mitigasi untuk meminimalkan hal ini adalah dengan menerapkan *Directory Exclusion* berbasis *Regex* (*Regular Expression*) serta melakukan *whitelisting* pada *User/Process-ID* spesifik yang menjalankan tugas *cronjob* pencadan-

gan. Di sisi lain, *False Negative* mencatatkan nilai *Recall* sebesar 96%. Anomali ini berhasil lolos karena penyerang menggunakan taktik *Low-and-Slow*, yaitu memodifikasi *backdoor* dengan jeda waktu yang sangat lambat sehingga gagal memicu peringatan batas frekuensi waktu (*time frequency threshold*) dari mesin *ruleset*. Strategi praktis untuk mengatasi celah ini adalah dengan tidak hanya mengandalkan perhitungan frekuensi waktu, tetapi juga memperketat aturan deteksi berbasis anomali ukuran berkas dan perubahan *hash* secara lebih granular.

Sebagai bentuk perluasan evaluasi dan perbandingan dengan pendekatan pada lingkungan yang berbeda, penelitian Farrel (2024) mencatatkan waktu tunda pemblokiran brute force rata-rata sekitar 2 hingga 3 detik yang sepenuhnya diuji di jaringan lokal (LAN). Pada penelitian ini, implementasi *hybrid cloud* berbasis VPS mencatatkan waktu rata-rata 3,6 detik. Selisih latensi ini sangat wajar mengingat adanya proses penjaluran (*routing*) internet asinkron, namun terbukti tetap beroperasi optimal dan andal. Jika dikomparasikan dengan studi Nas et al. (2023) pada infrastruktur instansi pemerintahan yang hanya menerapkan *monitoring* pasif (tanpa blokir otomatis), arsitektur pada penelitian ini mengungguli *baseline* dengan menawarkan kapabilitas *Active Response* yang mengeksekusi aksi mitigasi pada seluruh anomali siber yang terdeteksi secara *real-time*.

3.7 Analisis Efisiensi Sumber Daya dan Skalabilitas

Selain metrik kecepatan deteksi, efisiensi arsitektur dievaluasi berdasarkan observasi tingkat deplesi sumber daya (*resource depletion*). Pada kondisi diam (*idle*), agen Wazuh di peladen target terbukti beroperasi sangat efisien karena hanya bertugas melakukan pembacaan log lokal tanpa melakukan *decoding* yang berat. Namun, pada pengujian eskalasi ekstrem seperti serangan *HTTP Flood DoS* (100.000 *request* simultan), observasi menunjukkan adanya peningkatan utilisasi CPU yang signifikan pada peladen target. Peningkatan ini tidak diakibatkan oleh beban agen keamanan, melainkan oleh beban peladen web (*Apache*) yang merespons trafik asinkron. Meskipun beban CPU melonjak dan *throughput log* mengalami penumpukan antrean sementara (*network queue*), agen tetap berhasil menjaga kestabilan penggunaan memori (RAM) dan mentransmisikan instruksi blokir ke subsistem *IPtables*. Terkait skalabilitas, untuk implementasi pada jumlah *endpoint* instansi yang lebih masif di masa mendatang, disarankan penggunaan penyeimbang beban (*Load Balancer*) pada simpul VPS agar *throughput log* yang besar tidak memicu *bottleneck* pada *Master Node* Wazuh.

4. Kesimpulan

Berdasarkan hasil pengujian dan analisis, implementasi arsitektur keamanan *Security Information and Event Management* (SIEM) berbasis Wazuh dan Graylog pada infrastruktur *Virtual Private Server* (VPS) terbukti efektif dalam memetakan dan melindungi peladen. Implementasi ini dilakukan dengan batasan penelitian yang tegas: pengawasan pada infrastruktur produksi (*Server Development* riil instansi Diskominfo) beroperasi hanya pada tahap audit pasif. Sementara itu, seluruh eksekusi mitigasi otomatis terhadap serangan siber aktif sepenuhnya dijalankan di dalam batasan lingkungan laboratorium virtual yang terisolasi.

Pada pengujian serangan agresif di lingkungan laboratorium, sistem mampu mengeksekusi mitigasi otonom (*Active Response*) secara presisi; memblokir intrusi *SSH Brute Force* dalam waktu rata-rata 3,6 detik, memutus pemindaian *Dynamic Application Security Testing* (DAST) OWASP ZAP dalam 2,6 detik, serta menghentikan manipulasi *ransomware* dalam waktu 2,70 detik. Lebih lanjut, sistem terbukti mampu menggugurkan rata-rata 83.345 anomali permintaan *HTTP Flood Denial of Service* (DoS), meskipun jumlah permintaan awal yang lolos sempat berfluktuasi akibat padatnya beban trafik lalu lintas jaringan dan penumpukan antrean pembacaan log (*network queue*) dalam hitungan milidetik.

Meskipun kinerja mitigasi otomatis menunjukkan hasil yang andal, sistem ini masih memiliki keterbatasan yang membuka peluang bagi pengembangan di masa mendatang. Sebagai arah pengembangan penelitian (*future works*), arsitektur SIEM-SOC ini perlu diekspansi agar lebih relevan dengan lanskap ancaman keamanan siber modern. Disarankan agar sistem tidak hanya berfokus pada pemindaian struktur muatan (*payload inspection*) dan penyetelan FIM, tetapi juga diintegrasikan dengan teknologi Kecerdasan Buatan (AI) atau *Machine Learning*. Integrasi ini akan menghadirkan kapabilitas deteksi anomali adaptif (*User and Entity Behavior Analytics / UEBA*), sehingga sistem tidak lagi bergantung secara kaku pada ambang batas aturan statis (*ruleset*). Selain itu, pengadopsian kerangka kerja *Security Orchestration, Automation, and Response* (SOAR) yang dipadukan dengan *Threat Intelligence* eksternal sangat direkomendasikan. Pendekatan ini akan memperkaya konteks data log, memperluas cakupan otomatisasi mitigasi, serta secara efektif menekan angka *False Negative* pada ancaman tingkat lanjut (APT) yang berkarakteristik *low-and-slow*.

Daftar Pustaka

Artha Kusuma, G. H. (2022): *Sistem Firewall untuk Pencegahan DDOS ATTACK di Masa Pandemi*

- Covid-19*, Journal of Informatics and Advanced Computing (JIAC), vol. 3, no. 1, pp. 53-58.
- B. V. Labs (2023): *SIEM Stack: Ingest Wazuh alert ke dalam Graylog*, <https://bayuskylabs.com/siem-stack-ingest-wazuh-alert-ke-dalam-graylog/>
- CNN Indonesia (2024): *PDN Diserang Ransomware, Apa Itu dan Cara Kerjanya?*, <https://www.cnnindonesia.com/teknologi/20240625102506-192-1114068/pdn-diserang-ransomware-apa-itu-dan-cara-kerjanya>
- Farrel, F. I. (2024): *Implementation of Security Information & Event Management (SIEM) Wazuh with Active Response and Telegram Notification for Mitigating Brute Force Attacks on the GT-12TI USAKTI Information System*, INTEL MAT-ICS, vol. 4, no. 1, pp. 1-11.
- Gilang Trifebriyanto, F., Syani, M., Alpiyas, F., Laia, F. & Parga Zen, B. (2026): *Implementasi Monitoring Website Pemkot Cimahi Menggunakan Uptime Kuma Berbasis VPS*, JSIG, vol. 4, no. 1, pp. 71-78.
- Granadillo, G. G., Zarzosa, S. G. & Diaz, R. (2021): *Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures*, Sensors, vol. 21, no. 17, p. 5935.
- Graylog, Inc. (2025): *Graylog Documentation*, <https://go.graylog.org/docs>
- Khotimah, H., Bimantoro, F. & Kabanga, R. S. (2022): *Implementasi Security Information and Event Management (SIEM) Pada Aplikasi SMS Center Pemerintah Daerah Provinsi Tenggara Barat*, J-BEGATI: Jurnal Teknologi Informasi, vol. 3, no. 3, pp. 213-219.
- Kompas.com (2023): *BSI Diduga Kena Serangan Ransomware, Apa Itu?*, <https://www.kompas.com/tren/read/2023/05/13/183000165/bsi-diduga-kena-serangan-ransomware-apa-itu->
- Nas, M., Ulfiah, F. & Putri, U. (2023): *Analisis Sistem Security Information and Event Management (SIEM) Aplikasi Wazuh pada Dinas Komunikasi Informatika Statistik dan Persandian Sulawesi Selatan*, Jurnal Teknologi Elektroika, vol. 20, no. 2, pp. 29-34.
- Oz, H., Aris, A., Levi, A. & Uluagac, A. S. (2022): *A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions*, ACM Computing Surveys, vol. 54, no. 11s, Art. no. 238.
- Pfleeger, C. P., Pfleeger, S. L. & Margulies, J. (2015): *Security in Computing*, 5th ed. Upper Saddle River, NJ: Prentice Hall.
- Pratama, M. D., Nova, F. & Prayama, D. (2022): *Wazuh sebagai Log Event Management dan Deteksi Celah Keamanan pada Server dari Serangan Dos*, Jurnal Ilmiah Teknologi Sistem Informasi, vol. 3, no. 1, pp. 1-7.
- Pratama, Y., Fathurohman, M. A. & Gobel, Y. P. (2023): *Implementation of MISP Threat Intelligence with Wazuh as a Security Information and Event Management to Detect Malware Attacks*,

- Jurnal Teknik Informatika C.I.T Medicom, vol. 15, no. 1, pp. 27-34.
- Stankovic, S., Gajin, S. & Petrovic, R. (2022): *A Review Of Wazuh Tool Capabilities For Detecting Attacks Based on Log Analysis*, Proc. 2022 30th Telecommunications Forum (TELFOR), Belgrade, Serbia, pp. 1-5.
- Wazuh, Inc. (2025): *Wazuh Documentation*, <https://documentation.wazuh.com/current/index.html>