PENGEMBANGAN DASBOR *FILTERING LOG* TERPUSAT DAN OTOMASI RESPONS SERANGAN SIBER UNTUK REDUKSI *MEAN TIME TO RESPOND* (MTTR)

SKRIPSI

PRATAMA VARIAN ANDIKA PARULIAN
2207421040

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
TAHUN 2026

PENGEMBANGAN DASBOR *FILTERING LOG* TERPUSAT DAN OTOMASI RESPONS SERANGAN SIBER UNTUK REDUKSI *MEAN TIME TO RESPOND* (MTTR)

SKRIPSI

Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk Memperoleh Diploma Empat Politeknik

PRATAMA VARIAN ANDIKA PARULIAN
2207421040

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK NEGERI JAKARTA
TAHUN 2026

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Pratama Varian Andika Parulian

NIM : 2207421040

Jurusan/ProgramStudi : T.Informatika dan Komputer / T. Multimedia dan Jaringan

Judul skripsi : PENGEMBANGAN DASBOR FILTERING LOG TERPUSAT DAN OTOMASI RESPONS SERANGAN SIBER UNTUK REDUKSI MEAN TIME TO RESPOND (MTTR)

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung cirri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatantersebut.

Depok, 7, Juni, 2026

Yang membuat pernyataan

(Pratama Varian Andika Parulian)

NIM. 2207421040

LEMBAR PENGESAHAN

Skripsi diajukan oleh :
Nama : Christian Nataniel Yosua Purba
NIM : 2207421043
Program Studi : TMJ
Judul Skripsi : OPTIMALISASI PROSES KEAMANAN PROAKTIF MELALUI HUB MANAJEMEN MULTI-WAZUH TERINTEGRASI *THREAT INTELLIGENCE* DAN NOTIFIKASI *REAL-TIME*

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Senin Tanggal 22 Bulan Juni Tahun 2026 dan dinyatakan **LULUS**.

Disahkan oleh:

Pembimbing I : Ayu Rosyida Zain, S.ST., M.T ( )

Penguji I : Dr. Defiana Arnaldy, S.Tp., M.Si. ( )

Penguji II : Iik Muhamad Malik Matin, S.Kom., M.T. ( )

Penguji III : Andika Riyadi Jasril, S.Pd., M.Pd.T ( )

Mengetahui:
Jurusan Teknik Informatika dan Komputer
Ketua

Dr. Anita Hidayati, S.Kom., M.Kom.
NIP. 197908032003122003

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan Skripsi yang berjudul "Pengembangan Dasbor Filtering Log Terpusat dan Otomasi Respons Serangan Siber untuk Reduksi Mean Time To Respond (MTTR)". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Terapan pada Program Studi Teknik Multimedia dan Jaringan, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta.

Penulis menyadari bahwa penyusunan skripsi ini tidak akan terwujud tanpa bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Ibu Ayu Rosyida Zain, S.ST., M.T., selaku dosen pembimbing yang telah meluangkan banyak waktu, tenaga, dan pikiran untuk memberikan bimbingan, arahan, serta masukan yang sangat berharga selama proses penyusunan penelitian ini.
2. Ibunda tercinta, yang senantiasa memberikan doa yang tiada henti, kasih sayang, serta dukungan moril dan materil yang menjadi kekuatan bagi penulis untuk terus melangkah.
3. Rekan satu tim skripsi, Abian dan Christian, atas kerja sama yang solid, diskusi panjang, dan perjuangan bersama dalam merancang serta mengembangkan sistem ini.
4. Teman-teman PBL (Azzuri, Ayu, Caca, Izza, dan Rei), yang selalu memberikan semangat, warna, dan dukungan dalam setiap proses perkuliahan dan pengerjaan proyek.
5. Seluruh teman-teman seperjuangan Teknik Multimedia dan Jaringan angkatan 2022 (TMJ 22), atas kebersamaan, bantuan, dan kenangan indah selama menempuh pendidikan di Politeknik Negeri Jakarta.

Penulis menyadari bahwa penelitian ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi

penyempurnaan penelitian ini ke depannya. Semoga penelitian ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya di bidang keamanan siber.

Depok, 07 Juni, 2026

Pratama Varian Andika Parulian

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya bertanda tangan dibawah ini :

Nama : Pratama Varian Andika Parulian

NIM : 2207421040

Jurusan ProgramStudi : T.Informatika dan Komputer / T. Multimedia dan Jaringan

Demi pengembangan ilmu pengetahuan , menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul :

Pengembangan Dasbor Filtering Log Terpusat Dan Otomasi Respons Serangan Siber Untuk Reduksi Mean Time To Respond (MTTR)

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta Berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyatan ini saya buat dengan sebenarnya.

Depok, 07, Juni, 2026
Yang menyatakan

Pratama Varian Andika Parulian

PENGEMBANGAN DASBOR *FILTERING LOG* TERPUSAT DAN OTOMASI RESPONS SERANGAN SIBER UNTUK REDUKSI *MEAN TIME TO RESPOND* (MTTR)

ABSTRAK

Fragmentasi infrastruktur keamanan siber dan prosedur mitigasi manual lintas platform sering kali memicu hambatan operasional (context switching) bagi analis SOC, sehingga memperlambat penanganan insiden dan meningkatkan Mean Time to Respond (MTTR). Penelitian ini bertujuan merancang dan mengimplementasikan sistem middleware SOAR terpusat berupa dasbor filtering log dan mesin otomasi respons (autoblocking) untuk meminimalkan waktu penanganan serangan. Metodologi rekayasa sistem yang digunakan adalah prototyping, dengan mengintegrasikan basis data OLAP ClickHouse untuk agregasi log analitik skala besar dan Redis Sorted Set untuk kalkulasi intensitas serangan berbasis sliding window secara real-time. Pengujian dilakukan di lingkungan staging terisolasi menggunakan simulasi serangan SSH Brute Force dan DoS *HTTP Flood. Hasil penelitian menunjukkan bahwa sistem berhasil mereduksi MTTR sebesar 68,27% pada serangan Brute Force (dari 76,67 menjadi 24,33 detik) dan 66,23% pada serangan* DoS *(dari 77 menjadi 26 detik). Dasbor terpusat berhasil menyederhanakan langkah investigasi analis dari 7–10 langkah menjadi 3–4 langkah, yang memotong waktu analisis sebesar 81,55%. Kecepatan kueri ClickHouse mencapai 110,4 ms pada volume 5 juta log, membuktikan efisiensi performa pencarian log. Disimpulkan bahwa sistem ini efektif meningkatkan efisiensi operasional SOC dan mempercepat containment ancaman secara terkoordinasi.*

Kata kunci: autoblocking, ClickHouse, dasbor terpusat, Mean Time to Respond (MTTR), Redis, Security Orchestration*, Automation, and Response (SOAR), Wazuh.*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Untuk menghadapi serangan siber yang semakin terdistribusi dan otomatis, organisasi umumnya menerapkan arsitektur keamanan berlapis yang terdiri dari berbagai perangkat, seperti *Security Information and Event Management* (SIEM) untuk deteksi, *Cyber Threat Intelligence* (CTI) untuk pengayaan, serta *firewall* untuk pencegahan. Pendekatan ini mendorong fenomena *tool sprawl*, di mana organisasi *enterprise* harus mengelola puluhan hingga ratusan alat keamanan secara bersamaan, dengan rata-rata 60–80 alat dan dapat mencapai 140 alat pada lingkungan yang lebih kompleks (*WisdomTree*, 2022).

Fragmentasi antar alat tersebut menimbulkan inefisiensi operasional yang signifikan. Arif et al. (2025) mengidentifikasi fenomena ini sebagai *Time Tax*, yaitu beban waktu akibat perpindahan antar aplikasi (*Context Switching*) dan notifikasi yang terduplikasi, yang dapat menghabiskan hingga 27,6 jam kerja per analis per tahun. Kondisi ini meningkatkan risiko *alert fatigue* dan memperlambat respons insiden, karena keterhubungan antar sistem masih sangat bergantung pada proses manual, sehingga berdampak langsung pada meningkatnya *Mean Time to Respond* (MTTR).

Pada lingkungan tanpa orkestrasi otomatis seperti di PT. XYZ, bottleneck muncul akibat prosedur manual yang mengharuskan administrator melakukan langkah repetitif lintas platform, yaitu (1) memantau da=n memfilter dasbor SIEM untuk memvalidasi serangan, (2) menyalin indikator serangan seperti IP *address*, (3) beralih antarmuka untuk melakukan autentikasi ke sistem *firewall*, dan (4) menginput aturan pemblokiran secara manual pada setiap perangkat. Ketergantungan pada alur kerja ini mengonsumsi waktu yang signifikan, meningkatkan latensi respons, serta memperbesar risiko *human error* akibat kelelahan operator, terutama pada infrastruktur berskala besar dengan arsitektur terdistribusi yang menjalankan banyak *instance* SIEM (Wazuh) dan *firewall* di berbagai data center atau kantor cabang. Fragmentasi ini menyebabkan data dan kontrol mitigasi terisolasi, sehingga analis kesulitan mengidentifikasi *instance*

Wazuh yang mendeteksi serangan dan *firewall* yang relevan untuk dikonfigurasi dalam skenario serangan terkoordinasi. Kondisi ini menunjukkan ketiadaan otomasi respons yang terpadu, yang pada akhirnya meningkatkan *Mean Time to Respond* (MTTR) secara eksponensial seiring bertambahnya jumlah perangkat dan membuka celah waktu bagi penyerang untuk melakukan penetrasi lanjutan atau *lateral movement*.

Untuk mengatasi inefisiensi operasional akibat fragmentasi alat keamanan, industri mengadopsi pendekatan *Security* Orchestration, Automation, *and* Response (SOAR) sebagai solusi standar untuk mengintegrasikan proses deteksi dan respons, namun tinjauan literatur menunjukkan bahwa solusi yang ada masih memiliki keterbatasan pada aspek antarmuka dan skalabilitas. Aljahdali dan Alsulami (2025) mengusulkan kerangka kerja SOAR berbasis Splunk yang berfokus pada otomatisasi di sisi backend, tetapi tidak menyediakan dasbor terpusat dengan kemampuan *searching* dan *filtering log* terpusat, serta tidak menyediakan antarmuka manajemen untuk melakukan pembukaan blokir (*unblocking*) secara praktis, sehingga kurang mendukung operasional harian yang melibatkan intervensi analis (*human-in-the-loop*). Di sisi lain, Farrel et al. (2024) mengimplementasikan mekanisme *Active Response* Wazuh yang mampu mengirimkan perintah pemblokiran alamat IP hanya ke satu *firewall* yang terhubung langsung dengan satu manajer Wazuh. Arsitektur ini tidak memiliki mekanisme untuk mendistribusikan kebijakan pemblokiran yang sama ke beberapa *firewall* secara simultan ketika organisasi menjalankan banyak *instance* Wazuh dan *firewall* di *server* berbeda, sehingga solusi tersebut tidak mampu menyatukan visibilitas maupun kontrol mitigasi pada infrastruktur terdistribusi berskala besar seperti di PT. XYZ.

Penelitian ini mengusulkan pengembangan sebuah sistem orkestrasi terpusat yang berfungsi sebagai jembatan untuk mengisi celah tersebut. Sistem ini dirancang untuk: (1) Menyediakan satu dasbor terpusat dengan kemampuan *searching* dan *filtering* cepat dari berbagai sumber *instance* Wazuh menggunakan teknologi basis data OLAP (*ClickHouse*), dan (2) Menyediakan otomasi respons (*autoblocking*) yang secara otomatis mendeteksi ambang batas serangan dan mendistribusikan

perintah blokir ke seluruh *instance firewall* terkait. Solusi ini diharapkan dapat menghilangkan hambatan heterogenitas data pada fase *Identification* ke *Containment* dan mereduksi MTTR secara signifikan.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana merancang arsitektur sistem yang memungkinkan pencarian dan penyaringan (*searching* dan *filtering*) data *log* dari berbagai *instance server* Wazuh yang terpisah melalui satu dasbor *filtering log* terpusat?
2. Bagaimana mengimplementasikan fitur otomasi respons (*autoblocking*) menggunakan algoritma *sliding window* yang memanfaatkan aliran data *log* dari berbagai *instance server* Wazuh, untuk membantu menurunkan *Mean Time to Respond* (MTTR)?
3. Seberapa besar efektivitas sistem terpusat ini dalam menurunkan *Mean Time to Respond* (MTTR) dibandingkan dengan prosedur manual yang melibatkan perpindahan antar banyak konsol perangkat?

1.3 Batasan Masalah

Penelitian ini dibatasi pada ruang lingkup sebagai berikut:

1. Sistem yang dibangun dibatasi pada pengembangan logika otomasi respons (*autoblocking*), dan mekanisme *searching* & *filtering log* terpusat untuk mendukung investigasi insiden, dengan asumsi bahwa aliran data *log* yang diterima telah tervalidasi dan disediakan oleh modul konektor Wazuh *Management Hub*.
2. Data yang diolah dikhususkan pada *log security alerts* yang dihasilkan Wazuh, bukan keseluruhan *log* sistem operasi.
3. Logika otomasi respons (*autoblocking*) dirancang menggunakan pendekatan deterministik berbasis aturan dengan ambang batas statis melalui Redis, untuk memastikan eksekusi cepat, keputusan yang jelas, dan kemampuan audit yang transparan.
4. Pengujian efektivitas sistem (penurunan MTTR) dilakukan pada lingkungan *staging* terisolasi yang merepresentasikan topologi jaringan

PT. XYZ, dengan simulasi serangan yang dibatasi pada vektor serangan volumetrik umum (*Brute Force* dan DOS) untuk membuktikan konsep respons otomatis

1.4 Tujuan dan Manfaat

Tujuan Penelitian:

1. Merancang arsitektur sistem dengan satu dasbor terpusat yang memungkinkan pencarian (*searching*) dan penyaringan (*filtering*) *log* dari berbagai *instance server* Wazuh yang terpisah, sehingga analis dapat menelusuri insiden secara cepat dan konsisten.
2. Mengimplementasikan algoritma sliding window pada sistem otomasi respons *(autoblocking)* yang memanfaatkan *data log* dari berbagai *instance* Wazuh yang telah dikelola secara terpusat melalui Wazuh *Management Hub* untuk mendeteksi pola serangan dan mengeksekusi pemblokiran IP secara otomatis.
3. Mengukur secara kuantitatif bahwa pendekatan orkestrasi terpusat dapat mereduksi waktu respons insiden (MTTR) dibandingkan metode manual.

Manfaat Penelitian:

1. Mempercepat investigasi insiden dengan dasbor terpusat yang memungkinkan *searching* dan *filtering log* dari berbagai *instance* Wazuh tanpa perlu berpindah konsol.
2. Menyediakan eksekusi respons otomatis yang jelas dan dapat diaudit melalui algoritma otomasi respons (*autoblocking)* berbasis aturan untuk deteksi pola serangan.z
3. Mengurangi *Mean Time to Respond* (MTTR) dibandingkan prosedur manual, meningkatkan efektivitas operasional dan keamanan jaringan.

1.5 Sistematika Penulisan

Penyusunan laporan skripsi ini dibagi menjadi lima bab dengan sistematika sebagai berikut:

- **BAB I PENDAHULUAN**

  Membahas latar belakang masalah inefisiensi operasional keamanan,

rumusan dan batasan masalah, tujuan penelitian, serta sistematika penulisan laporan.

- **BAB II TINJAUAN PUSTAKA**

  Menguraikan landasan teori Manajemen Insiden, konsep SOAR, teknologi pendukung (Wazuh, ClickHouse, Redis), metrik MTTR, dan analisis celah penelitian terdahulu.

- **BAB III METODE PENELITIAN**

  Menjelaskan metode eksperimental dengan *model Prototyping*, mencakup perancangan arsitektur sistem, skenario simulasi serangan, serta teknik analisis data untuk mengukur penurunan MTTR.

- **BAB IV HASIL DAN PEMBAHASAN**

  Menyajikan hasil implementasi Dasbor *Filtering Log* Terpusat dan sistem otomasi respons *(Autoblocking)*, diikuti analisis pengujian fungsional dan perbandingan kecepatan respons antara metode manual dan otomatis.

- **BAB V PENUTUP**

  Berisi kesimpulan dari hasil penelitian dan saran untuk pengembangan sistem di masa mendatang.

BAB V
PENUTUP

5.1 Simpulan

Rancang bangun arsitektur sistem orkestrasi terpusat yang mengintegrasikan basis data OLAP ClickHouse untuk memproses data dari beberapa *instance* Wazuh *Indexer* menghasilkan waktu kueri rata-rata sebesar 20,1 ms pada *volume* 100.000 *log*, dan meningkat menjadi 110,4 ms pada *volume* data ekstrem 5.000.000 baris *log*, yang berarti tetap berada di bawah ambang batas waktu respons kueri 1 detik. Untuk menjawab kebutuhan otomatisasi respons, sistem otomasi respons (*autoblocking*) berbasis aturan (*rule-based thresholding*) diimplementasikan dengan menerapkan mekanisme *sliding window* pada Redis *Sorted Set* serta integrasi dengan OPNsense *firewall*. Pengujian kinerja penulisan (*ingestion*) *log* mencatatkan tingkat keberhasilan penulisan data sebesar 100% dengan waktu pemrosesan rata-rata 4,3 detik (puncak 5,0 detik) pada skenario beban kerja ekstrem 1.000 EPS (30.000 *log* per siklus scheduler 30 detik), sehingga seluruh pemrosesan selesai sebelum siklus berikutnya dimulai tanpa adanya penumpukan antrean data (*queue backlog*).

Pengukuran kuantitatif terhadap kinerja sistem orkestrasi terpusat ini mencatatkan reduksi *Mean Time to Respond* (MTTR) yang terukur dibandingkan dengan metode penanganan manual oleh analis. Pada mitigasi serangan SSH Brute Force, nilai MTTR berkurang dari rata-rata 76,67 detik pada metode manual menjadi 24,33 detik menggunakan sistem otomatis (mengalami reduksi sebesar 68,27%). Sementara itu, pada mitigasi serangan *Denial of Service* (DoS) jenis HTTP *Flood*, nilai MTTR berkurang dari rata-rata 77 detik pada metode manual menjadi 26 detik pada metode otomatis (mengalami reduksi sebesar 66,23%). Berdasarkan analisis langkah kerja (*step reduction analysis*), sistem ini memotong jumlah proses penelusuran analis dari rata-rata 7 hingga 10 langkah manual menjadi hanya 3 hingga 4 langkah terpusat melalui satu dasbor, yang menghasilkan penurunan waktu pengerjaan tugas (*time on task*) rata-rata sebesar 81,55% di seluruh skenario pengujian.

5.2 Saran

Untuk pengembangan sistem ke lingkungan produksi dengan skala yang lebih besar, beberapa rekomendasi pengembangan dapat dilakukan. Pendekatan deteksi berbasis ambang batas statis disarankan untuk ditingkatkan menjadi metode yang lebih adaptif, seperti penerapan *multi-tiered threshold* atau integrasi model deteksi anomali berbasis *machine learning*. Modifikasi ini diperlukan untuk memperkuat kemampuan identifikasi terhadap pola serangan *low and slow* sekaligus meminimalkan tingkat *false positive* pada aktivitas pengguna yang sah.

Selain penyesuaian pada mekanisme deteksi, optimalisasi pada infrastruktur penyimpanan data juga diperlukan, khususnya pada komponen ClickHouse yang menjadi titik beban utama saat lalu lintas data tinggi. Peningkatan efisiensi sistem ini dapat dicapai melalui penyesuaian *index granularity*, pemanfaatan fitur *projections*, serta penerapan arsitektur *clustering* guna mendistribusikan beban komputasi secara lebih merata pada volume data yang besar.

Daftar Pustaka

Aljahdali, A.O. dan Alsulami, R. 2025. "Streamlining Threat Response and Automating Critical Use Cases with Security Orchestration, Automation and Response (SOAR)". *Journal of Digital Security and Forensics*, 2 (1). https://doi.org/10.29121/digisecforensics.v2.i1.2025.45.

Arif, I., Tariq, A. dan Barchuk, B. 2025. "The Cost of Fragmentation: Measuring Time, Spend and Risk in Personal Cybersecurity Tool Stacks". *World Journal of Advanced Engineering Technology and Sciences*, 16 (3), 334–363. https://doi.org/10.30574/wjaets.2025.16.3.1350.

Ariyanto, Y., Syaifudin, Y.W., Ratsanjani, M.H., Muladawila, A.R., Fatmawati, T., Saputra, P.Y. dan Setiadi, C. 2025. "Cyber Threat Detection and Automated Response Using Wazuh and Telegram API". *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 25 (1), 173–188. https://doi.org/10.30812/matrik.v25i1.5610.

Barnard-Brak, L., Watkins, L. dan Richman, D.M. 2021. "Optimal Number of Baseline Sessions Before Changing Phases Within Single-Case Experimental Designs". *Behavioural Processes*, 191, 104461. https://doi.org/10.1016/j.beproc.2021.104461.

Cornejo, O., Briola, D., Micucci, D. et al. 2024. "A Family of Experiments About How Developers Perceive Delayed System Response Time". *Software Quality Journal*. https://doi.org/10.1007/s11219-024-09660-w.

Duras, R. 2024. *A Comparative Analysis of the Ingestion and Storage Performance of Log Aggregation Solutions: Elastic Stack & SigNoz*. [Thesis] DIVA. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1838164&dswid=9788. [17 January 2026].

Eddelbuettel, D. 2022. "A Brief Introduction to Redis". *arXiv preprint arXiv:2203.06559*. https://doi.org/10.48550/arXiv.2203.06559.

Farrel, F.I., Mardianto dan Qamar, A.S. 2024. "Implementation of Security Information & Event Management (SIEM) Wazuh with Active Response and Telegram Notification for Mitigating Brute Force Attacks on the GT-I2TI USAKTI Information System". *Intelmatics*, 4 (1), 1–7. https://doi.org/10.25105/itm.v4i1.18529.

Gauthama Raman, M.R., Khandekar, S., Murarishetti, R., Chew, Z.Y.C., Ng, G.F.E. dan Zhou, J. 2025. "Network-Based Real-Time Detection of Data Manipulation Attacks in Industrial Control Systems". *International Journal of Information Security*, 25 (12). https://doi.org/10.1007/s10207-025-01172-3.

Harahap, F.A., Ikhwan, A. dan Irawan, M.D. 2025. "Pemilihan Abang Kakak Tebing Tinggi dengan Metode Profile Matching Berbasis Website". *METIK JURNAL*, 9 (2), 247–255. https://doi.org/10.47002/metik.v9i2.1081.

Islam, M. dan Raisa, R. 2024. "Wazuh SIEM for Cyber Security and Threat Mitigation in Apparel Industries". *International Journal of Engineering Materials and Manufacture*, 9 (4), 136–144. https://doi.org/10.26776/ijemm.09.04.2020.02.

Kurniawan, A., Nursyabani, N. dan Matin, I.M.M. 2025. "Analisis Keamanan Website Kota Depok Menggunakan Metode Vulnerability Assesment". *Multinetics*, 11 (1). https://doi.org/10.32722/multinetics.v11i1.7226.

Lee, M., Jang-Jaccard, J. dan Kwak, J. 2022. "Novel Architecture of Security Orchestration, Automation and Response Internet of Blended Environment". *Computers, Materials & Continua*, 73 (1), 199–223. https://doi.org/10.32604/cmc.2022.028495.

Lumbantoruan, J., Makarim, B., Trianto, N. dan Setiawan, K.W. 2025. "Optimization of Audit Daemon for Preventing Ransomware Attacks on Linux Servers". *Proceedings of the 1st International Conference on Research and Innovations in Information and Engineering Technology*, 114–121. https://doi.org/10.5220/0014266200004928.

Radview. 2026. *Benchmark Testing: How to Measure & Compare Performance*. Radview. https://www.radview.com/blog/benchmark-testing-how-to-measure-compare-performance/. [27 May 2026].

Saarinen, A., Farooqi, M.Z., Pärssinen, M. dan Manner, J. 2025. "Understanding and Mitigating Webpage Data Bloat: Causes and Preventive Measures". *Energy Informatics Review*. https://doi.org/10.1145/3727200.3727216.

Schulze, R., Schreiber, T., Yatsishin, I., Dahimene, R. dan Milovidov, A. 2024. "ClickHouse - Lightning Fast Analytics for Everyone". *Proceedings of the VLDB Endowment*, 17 (12), 3731–3744. https://doi.org/10.14778/3685800.3685802.

Shankar, A. dan Madisetti, V. 2024. "A Framework for Cybersecurity Alert Distribution and Response Network (ADRIAN)". *Journal of Software Engineering and Applications*, 17 (05), 396–420. https://doi.org/10.4236/jsea.2024.175022.

Sheeraz, M., Paracha, M.A., Haque, M.U., Durad, M.H., Mohsin, S.M., Band, S.S. dan Mosavi, A. 2023. "Effective Security Monitoring Using Efficient SIEM Architecture". *Human-centric Computing and Information Sciences*, 13 (1), 17. https://doi.org/10.22967/HCIS.2023.13.017.

Sujan Kumar, K. 2025. "Next-Generation Security Operations: Leveraging Automation for Proactive Threat Mitigation". *INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 09 (04), 1–9. https://doi.org/10.55041/ijsrem43432.

Verwiebe, J., Grulich, P.M., Traub, J. dan Markl, V. 2023. "Survey of Window Types for Aggregation in Stream Processing Systems". *The VLDB Journal*, 32 (5), 985–1011. https://doi.org/10.1007/s00778-022-00778-6.

WisdomTree dan Gannatti, C. 2022. *Have You Experienced 'Tool Sprawl' in Cybersecurity?* WisdomTree. https://www.wisdomtree.com/api/sitecore/pdf/getblogpdf?id=efb270d1-68e5-45b9-a022-faf200cfedd8. [6 January 2026].

DAFTAR RIWAYAT HIDUP PENULIS

Pratama Varian Andika Parulian Siregar, lahir di bandung, 2004, lulus dari SMK Ananda Bekasi dengan keahlian di Jurusan Teknik Komputer dan Jaringan pada tahun 2022. Dan melanjutkan pendidikan akademik sebagai mahasiswa di Program Studi Teknik Multimedia dan Jaringan, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta (PNJ).

1. LAMPIRAN 1: Hasil pengujian Kecepatan Query

A. Script Pengujian query

```
#!/bin/bash

# Configuration
CH_USER="<REDACTED>"
CH_PASS="<REDACTED>"
CH_CONTAINER="56c6f93bf78a_lamis_clickhouse"
WZ_USER="<REDACTED>"
WZ_PASS="<REDACTED>"
WZ_URL="https://localhost:9200"

# Test sizes
LIMITS=(100000 500000 1000000 5000000)

ITERATIONS=10
REGEX_PATTERN=".*fail.*"

run_clickhouse() {
    local limit=$1

    docker exec $CH_CONTAINER clickhouse-client \
        --user $CH_USER \
        --password $CH_PASS \
        --query="SYSTEM DROP MARK CACHE; SYSTEM DROP UNCOMPRESSED CACHE;" > /dev/null 2>&1


    sleep 1

    sql="SELECT count() FROM (SELECT full_log FROM wazuh_alerts LIMIT $limit) WHERE match(full_log, '$REGEX_PATTERN') FORMAT Null;"

    elapsed=$(docker exec $CH_CONTAINER clickhouse-client \
        --time \
        --user $CH_USER \
        --password $CH_PASS \
        --query="$sql" 2>&1 | tr -d '\r' | grep -E '^[0-9.]+$')

    if [[ -z "$elapsed" ]]; then
        echo "0"
    else
        echo "$elapsed * 1000" | bc -l | cut -d. -f1
    fi
}

run_wazuh() {
    local limit=$1

    curl -s -k -u "$WZ_USER:$WZ_PASS" -X POST "$WZ_URL/_cache/clear" > /dev/null 2>&1
    sleep 1
```

```
47
48      dsl='{
49          "size": 0,
50          "track_total_hits": true,
51          "query": {
52              "regexp": {
53                  "full_log": "'$REGEX_PATTERN'"
54              }
55          }
56      }'
57
58      res=$(curl -s -k -u "$WZ_USER:$WZ_PASS" \
59          -X GET "$WZ_URL/wazuh-alerts-*/_search" \
60          -H 'Content-Type: application/json' \
61          -d"$dsl")
62
63      took=$(echo "$res" | grep -o '"took":[0-9]*' | cut -d: -f2)
64
65      if [[ -z "$took" ]]; then
66          echo "0"
67      else
68          echo "$took"
69      fi
70  }
71
72  echo "============================================================"
73  echo "     FILTER SCALING TEST (ITERATIVE LIMITS x10 RUNS)      "
74  echo "============================================================"
75
76  for limit in "${LIMITS[@]}"; do
77      echo ""
78      echo "LIMIT = $limit"
79      echo "Iter | CH Filter (ms) | WZ Filter (ms)"
80      echo "----------------------------------------"
81
82      for i in $(seq 1 $ITERATIONS); do
83          ch_f=$(run_clickhouse $limit)
84          wz_f=$(run_wazuh $limit)
85
86          printf "%-4d | %-15d | %-15d\n" $i $ch_f $wz_f
87      done
88  done
```

B. Hasil running Script

```
Problems  Output  Debug Console  Terminal  Ports 8
lstmagang@ubuntu-magang-02:~/Practice_Ahead$ /home/lstmagang/Practice_Ahead/scratch/benchmark.sh
lstmagang@ubuntu-magang-02:~/Practice_Ahead$ /home/lstmagang/Practice_Ahead/scratch/benchmark.sh
=========================================================
     FILTER SCALING TEST (ITERATIVE LIMITS x10 RUNS)
=========================================================

LIMIT = 100000
Iter | CH Filter (ms) | WZ Filter (ms)
----------------------------------------
1    | 44             | 253
2    | 18             | 258
3    | 19             | 265
4    | 16             | 257
5    | 17             | 267
6    | 17             | 254
7    | 15             | 269
8    | 20             | 261
9    | 18             | 269
10   | 17             | 262

LIMIT = 500000
Iter | CH Filter (ms) | WZ Filter (ms)
----------------------------------------
1    | 29             | 246
2    | 29             | 258
3    | 25             | 252
4    | 34             | 270
5    | 24             | 263
6    | 28             | 251
7    | 26             | 266
8    | 26             | 261
9    | 23             | 256
10   | 26             | 331
```

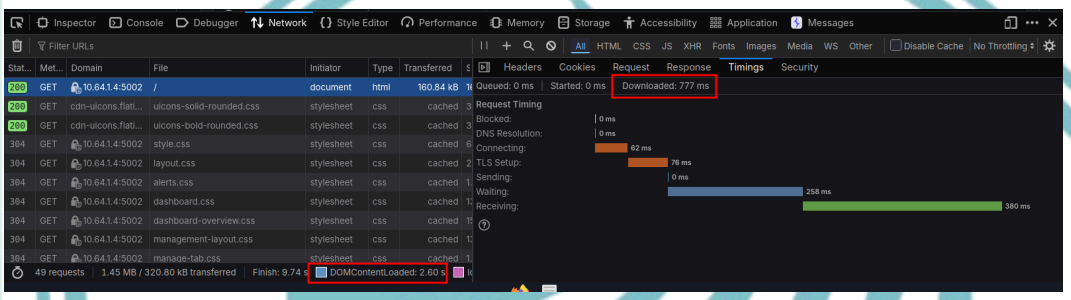
```
lstmagang@ubuntu-magang-02:~/Practice_Ahead$ /home/lstmagang/Practice_Ahead/scratch/benchmark.sh
4    | 34             | 270
5    | 24             | 263
6    | 28             | 251
7    | 26             | 266
8    | 26             | 261
9    | 23             | 256
10   | 26             | 331

LIMIT = 1000000
Iter | CH Filter (ms) | WZ Filter (ms)
----------------------------------------
1    | 36             | 255
2    | 32             | 268
3    | 43             | 307
4    | 33             | 261
5    | 39             | 292
6    | 33             | 256
7    | 52             | 248
8    | 35             | 253
9    | 34             | 253
10   | 40             | 251

LIMIT = 5000000
Iter | CH Filter (ms) | WZ Filter (ms)
----------------------------------------
1    | 107            | 286
2    | 105            | 260
3    | 110            | 249
4    | 149            | 257
5    | 96             | 252
6    | 112            | 303
7    | 111            | 263
8    | 101            | 239
9    | 98             | 301
10   | 115            | 250
lstmagang@ubuntu-magang-02:~/Practice_Ahead$
```

LAMPIRAN 2: HASIL PENGUJIAN KEBUTUHAN *NON*-FUNGSIONAL

A. Pengujian Performa Kueri Basis Data Dibanding Wazuh *Indexer* (NF-D1)
Pengujian ini dibuktikan pada Lampiran 1.

B. Pengujian Performa Keuri Basis Data Bervolume Masif (NF-D2)
Pengujian ini dibuktikan pada Lampiran 1.

C. Pengujian Latensi Pemuatan Antarmuka (NF-D3)
Membuktikan bahwa halaman Dasbor utama dapat dimuat sepenuhnya dalam waktu di bawah 3 detik menggunakan alat ukur *Browser Developer Tools*.

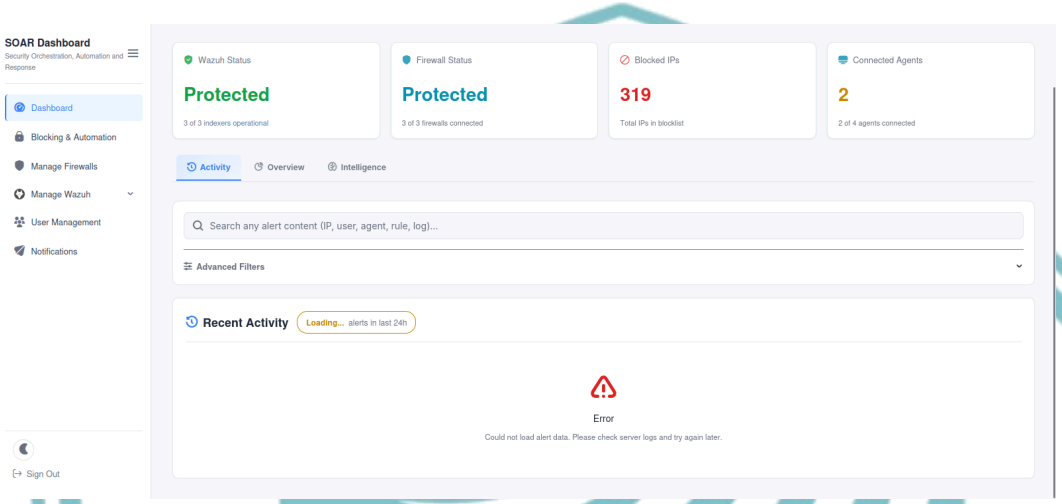


Gambar 2.1 Hasil Pengukuran Load *Time* Dasbor pada Tab Network (Firefox *DevTools*).

D. Pengujian Toleransi Kesalahan / Graceful Error (NF-D4)
Menunjukkan kemampuan aplikasi untuk tetap berjalan dan memberikan informasi yang jelas kepada pengguna saat koneksi ke basis data terputus.

```
tama@lst-thread-intel-risk:~/LAMIS/Practice_Ahead$ docker compose stop clickhouse
WARN[0000] /home/tama/LAMIS/Practice_Ahead/docker-compose.yaml: the attribute `vers
[+] Stopping 1/1
 ✔ Container lamis_clickhouse  Stopped
tama@lst-thread-intel-risk:~/LAMIS/Practice_Ahead$
```

Gambar 2.2 *Container* Clickhouse dimatikan



Gambar 2.3 Tampilan Peringatan error pada Dasbor.

E. Pengujian Latensi Respons Otomatis / *Autoblocking* (NF-A1)
Menunjukkan sinkronisasi waktu antara saat serangan terdeteksi oleh wazuh hingga perintah pemblokiran berhasil diterima oleh API *Firewall*.

- Pengujian ini dibuktikan pada lampiran 4

F. Pengujian Responsivitas UI di Bawah Beban Tinggi (NF-A2)
Membuktikan bahwa proses *backend* yang sedang memproses banyak *log* tidak mengganggu kenyamanan pengguna dalam menavigasi menu dasbor.

- Pengujian ini dibuktikan pada lampiran 6

LAMPIRAN 3: HASIL PENGUJIAN EFISIENSI ANALIS

Link Video Bukti Pengujian efisiensi analsis:
https://drive.google.com/drive/folders/1WA505c0lJFlDXjjR0Enq80pVT1VUmfae?usp=sharing

Tabel L.1 — Skenario A: Pencarian IoC Lintas *Server*

Sistem / Metode	Jam Mulai	Jam Selesai	Durasi (detik)
Sistem Usulan	15:28:14	15:28:30	16
Metode Eksisting (Manual - Wazuh)	15:28:31	15:30:26	115

Tabel L.2 — Skenario B: *Filter* Waktu & *Level* (08:00–09:00, *Level* 10–15)

Sistem / Metode	Jam Mulai	Jam Selesai	Durasi (detik)
Sistem Usulan	15:41:44	15:42:06	22
Metode Eksisting (Manual - Wazuh)	15:42:07	15:43:55	108

Tabel L.3 — Skenario C: *GeoIP* "China" (24 Jam)

Sistem / Metode	Jam Mulai	Jam Selesai	Durasi (detik)
Sistem Usulan	15:45:12	15:45:26	14
Metode Eksisting (Manual - Wazuh)	15:45:28	15:46:36	68

Tabel L.4 — Skenario D: *Sorting Severity* (*Descending*)

Sistem / Metode	Jam Mulai	Jam Selesai	Durasi (detik)
Sistem Usulan	15:49:44	15:50:02	18
Metode Eksisting (Manual - Wazuh)	15:50:03	15:51:38	95

LAMPIRAN 4: HASIL PENGUJIAN REDUKSI MTTR

```
ember
quartz
falcon
ripple
cinder
velvet
thorn
lumen
drift
sable
glyph
harbor
nova
spire
echo
bramble
orbit
flint
cascade
zephyr
```

Gambar Wordlist yang digunakan pada command hydra

Link Video Pengujian MTTR MANUAL untuk SSH Bruteforce:
https://drive.google.com/drive/folders/1kx2pB4W6f99cY4YqgicE7tRhf1gYb4pf?usp=sharing

Link Video Pengujian MTTR MANUAL untuk DoS:
https://drive.google.com/drive/folders/1kx2pB4W6f99cY4YqgicE7tRhf1gYb4pf?usp=sharing

Logs untuk Pengujian MTTR Automatis SSH Bruteforce:

```
--- ITERATION 1 EVIDENCE LOGS ---
2026-06-04 11:09:58,258 - app - INFO - IP 10.10.10.4 matched rule 'Default Level 10+'. Count within 1m window is now 1/1.
2026-06-04 11:09:58,260 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4. Cleared Redis key autoblock:ip:10.10.10.4:rule:level_10.
2026-06-04 11:09:58,261 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4 on rule 'Default Level 10+'. Adding to batch.
2026-06-04 11:09:58,564 - app - INFO - Starting concurrent blocklist push to 3 firewall(s)...
2026-06-04 11:09:58,567 - app - INFO - Starting blocklist update for OPNsense instance 'Main Opnsense'
2026-06-04 11:09:58,571 - app - INFO - Starting blocklist update for OPNsense instance 'FW Copy QA 1'
2026-06-04 11:09:58,573 - app - INFO - Starting blocklist update for 'pfsense test'
2026-06-04 11:09:59,030 - app - INFO - Successfully pushed blocklist to 'pfsense test' (pfsense).
2026-06-04 11:10:00,025 - app - INFO - Successfully pushed blocklist to 'FW Copy QA 1' (opnsense).
2026-06-04 11:10:01,059 - app - INFO - Successfully pushed blocklist to 'Main Opnsense' (opnsense).

--- ITERATION 2 EVIDENCE LOGS ---
2026-06-04 11:10:13,327 - app - INFO - Starting concurrent blocklist push to 3 firewall(s)...
2026-06-04 11:10:13,330 - app - INFO - Starting blocklist update for OPNsense instance 'Main Opnsense'
2026-06-04 11:10:13,333 - app - INFO - Starting blocklist update for OPNsense instance 'FW Copy QA 1'
2026-06-04 11:10:13,337 - app - INFO - Starting blocklist update for 'pfsense test'
2026-06-04 11:10:13,760 - app - INFO - Successfully pushed blocklist to 'pfsense test' (pfsense).
2026-06-04 11:10:15,550 - app - INFO - Successfully pushed blocklist to 'FW Copy QA 1' (opnsense).
2026-06-04 11:10:16,290 - app - INFO - Successfully pushed blocklist to 'Main Opnsense' (opnsense).
2026-06-04 11:10:53,066 - app - INFO - IP 10.10.10.4 matched rule 'Default Level 10+'. Count within 1m window is now 1/1.
2026-06-04 11:10:53,068 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4. Cleared Redis key autoblock:ip:10.10.10.4:rule:level_10.
2026-06-04 11:10:53,069 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4 on rule 'Default Level 10+'. Adding to batch.
2026-06-04 11:10:53,240 - app - INFO - Starting concurrent blocklist push to 3 firewall(s)...
2026-06-04 11:10:53,244 - app - INFO - Starting blocklist update for OPNsense instance 'Main Opnsense'
2026-06-04 11:10:53,247 - app - INFO - Starting blocklist update for OPNsense instance 'FW Copy QA 1'
2026-06-04 11:10:53,249 - app - INFO - Starting blocklist update for 'pfsense test'
2026-06-04 11:10:53,730 - app - INFO - Successfully pushed blocklist to 'pfsense test' (pfsense).
2026-06-04 11:10:55,473 - app - INFO - Successfully pushed blocklist to 'FW Copy QA 1' (opnsense).
2026-06-04 11:10:56,310 - app - INFO - Successfully pushed blocklist to 'Main Opnsense' (opnsense).

--- ITERATION 3 EVIDENCE LOGS ---
2026-06-04 11:11:43,331 - app - INFO - Starting concurrent blocklist push to 3 firewall(s)...
2026-06-04 11:11:43,335 - app - INFO - Starting blocklist update for OPNsense instance 'Main Opnsense'
2026-06-04 11:11:43,337 - app - INFO - Starting blocklist update for OPNsense instance 'FW Copy QA 1'
2026-06-04 11:11:43,340 - app - INFO - Starting blocklist update for 'pfsense test'
2026-06-04 11:11:43,791 - app - INFO - Successfully pushed blocklist to 'pfsense test' (pfsense).
2026-06-04 11:11:45,152 - app - INFO - Successfully pushed blocklist to 'FW Copy QA 1' (opnsense).
2026-06-04 11:11:45,828 - app - INFO - Successfully pushed blocklist to 'Main Opnsense' (opnsense).
2026-06-04 11:11:44,316 - app - INFO - IP 10.10.10.4 matched rule 'Default Level 10+'. Count within 1m window is now 1/1.
2026-06-04 11:11:44,318 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4. Cleared Redis key autoblock:ip:10.10.10.4:rule:level_10.
2026-06-04 11:11:44,320 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4 on rule 'Default Level 10+'. Adding to batch.
2026-06-04 11:11:44,439 - app - INFO - Starting concurrent blocklist push to 3 firewall(s)...
2026-06-04 11:11:44,444 - app - INFO - Starting blocklist update for OPNsense instance 'Main Opnsense'
2026-06-04 11:11:44,446 - app - INFO - Starting blocklist update for OPNsense instance 'FW Copy QA 1'
2026-06-04 11:11:44,449 - app - INFO - Starting blocklist update for 'pfsense test'
2026-06-04 11:11:45,032 - app - INFO - Successfully pushed blocklist to 'pfsense test' (pfsense).
2026-06-04 11:11:45,828 - app - INFO - Successfully pushed blocklist to 'FW Copy QA 1' (opnsense).
2026-06-04 11:11:46,557 - app - INFO - Successfully pushed blocklist to 'Main Opnsense' (opnsense).
 Ctrl+L for Agent
```

Logs untuk Pengujian MTTR Automatis DoS:

NGUJIAN 5 > logs > evidence_dos.log

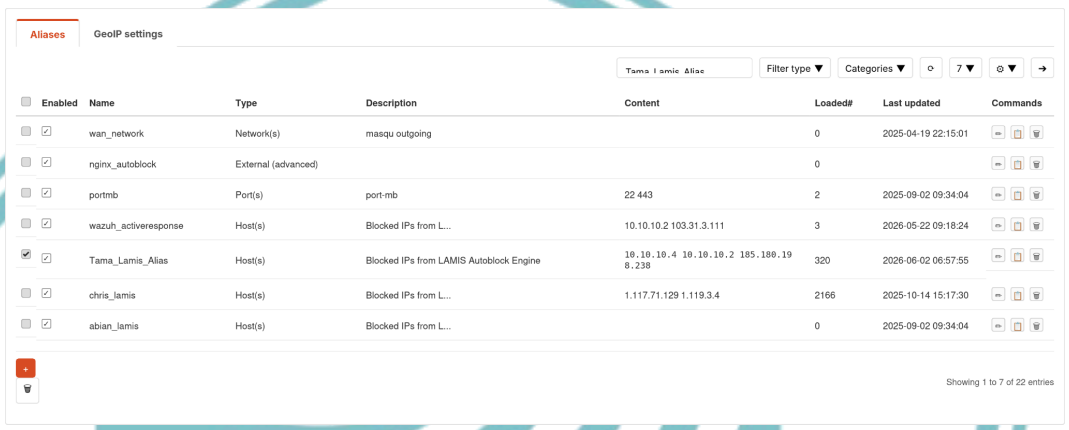
```
--- ITERATION 1 EVIDENCE LOGS ---
2026-06-05 07:44:23,011 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 1/5.
2026-06-05 07:44:23,015 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 2/5.
2026-06-05 07:44:23,017 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 3/5.
2026-06-05 07:44:23,019 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 4/5.
2026-06-05 07:44:23,021 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 5/5.
2026-06-05 07:44:23,022 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4. Cleared Redis key autoblock:ip:10.10.10.4:rule:3.
2026-06-05 07:44:23,024 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4 on rule 'ApacheBench DoS Detection'. Adding to batch.
2026-06-05 07:44:23,039 - app - INFO - Starting concurrent blocklist push to 3 firewall(s)...
2026-06-05 07:44:23,041 - app - INFO - Starting blocklist update for OPNsense instance 'Main Opnsense'
2026-06-05 07:44:23,042 - app - INFO - Starting blocklist update for OPNsense instance 'FW Copy QA 1'
2026-06-05 07:44:23,046 - app - INFO - Starting blocklist update for 'pfsense test'
2026-06-05 07:44:23,512 - app - INFO - Successfully pushed blocklist to 'pfsense test' (pfsense).
2026-06-05 07:44:25,156 - app - INFO - Successfully pushed blocklist to 'FW Copy QA 1' (opnsense).
2026-06-05 07:44:25,821 - app - INFO - Successfully pushed blocklist to 'Main Opnsense' (opnsense).

--- ITERATION 2 EVIDENCE LOGS ---
2026-06-05 07:44:58,735 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 1/5.
2026-06-05 07:44:58,737 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 2/5.
2026-06-05 07:44:58,739 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 3/5.
2026-06-05 07:44:58,741 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 4/5.
2026-06-05 07:44:58,742 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 5/5.
2026-06-05 07:44:58,744 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4. Cleared Redis key autoblock:ip:10.10.10.4:rule:3.
2026-06-05 07:44:58,745 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4 on rule 'ApacheBench DoS Detection'. Adding to batch.
2026-06-05 07:44:58,757 - app - INFO - Starting concurrent blocklist push to 3 firewall(s)...
2026-06-05 07:44:58,759 - app - INFO - Starting blocklist update for OPNsense instance 'Main Opnsense'
2026-06-05 07:44:58,763 - app - INFO - Starting blocklist update for OPNsense instance 'FW Copy QA 1'
2026-06-05 07:44:58,766 - app - INFO - Starting blocklist update for 'pfsense test'
2026-06-05 07:44:59,201 - app - INFO - Successfully pushed blocklist to 'pfsense test' (pfsense).
2026-06-05 07:45:00,191 - app - INFO - Successfully pushed blocklist to 'FW Copy QA 1' (opnsense).
2026-06-05 07:45:01,193 - app - INFO - Successfully pushed blocklist to 'Main Opnsense' (opnsense).

--- ITERATION 3 EVIDENCE LOGS ---
2026-06-05 07:46:29,293 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 1/5.
2026-06-05 07:46:29,295 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 2/5.
2026-06-05 07:46:29,296 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 3/5.
2026-06-05 07:46:29,298 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 4/5.
2026-06-05 07:46:29,300 - app - INFO - IP 10.10.10.4 matched rule 'ApacheBench DoS Detection'. Count within 1m window is now 5/5.
2026-06-05 07:46:29,301 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4. Cleared Redis key autoblock:ip:10.10.10.4:rule:3.
2026-06-05 07:46:29,302 - app - WARNING - THRESHOLD REACHED for IP 10.10.10.4 on rule 'ApacheBench DoS Detection'. Adding to batch.
2026-06-05 07:46:29,317 - app - INFO - Starting concurrent blocklist push to 3 firewall(s)...
2026-06-05 07:46:29,319 - app - INFO - Starting blocklist update for OPNsense instance 'Main Opnsense'
2026-06-05 07:46:29,321 - app - INFO - Starting blocklist update for OPNsense instance 'FW Copy QA 1'
2026-06-05 07:46:29,323 - app - INFO - Starting blocklist update for 'pfsense test'
2026-06-05 07:46:29,759 - app - INFO - Successfully pushed blocklist to 'pfsense test' (pfsense).
2026-06-05 07:46:30,476 - app - INFO - Successfully pushed blocklist to 'FW Copy QA 1' (opnsense).
2026-06-05 07:46:30,812 - app - INFO - Successfully pushed blocklist to 'Main Opnsense' (opnsense).
```

LAMPIRAN 5: HASIL PENGUJIAN AKURASI SISTEM

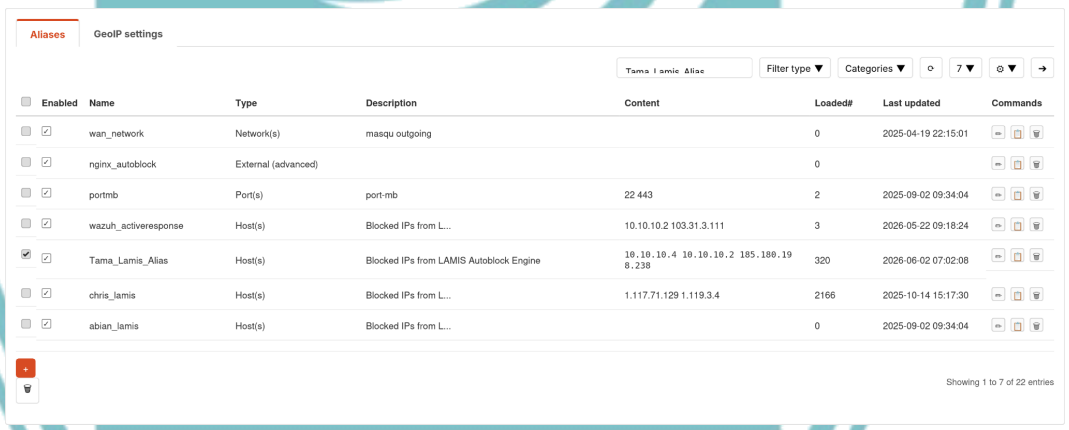
A. Bukti Screenshot Opnsense untuk skenario 1-4

a. *Full* kumpulan screenshot (40 file): https://drive.google.com/drive/folders/1VT7QCZIJ8L7egbYO1L4vwmBkbdV618Qd?usp=sharing

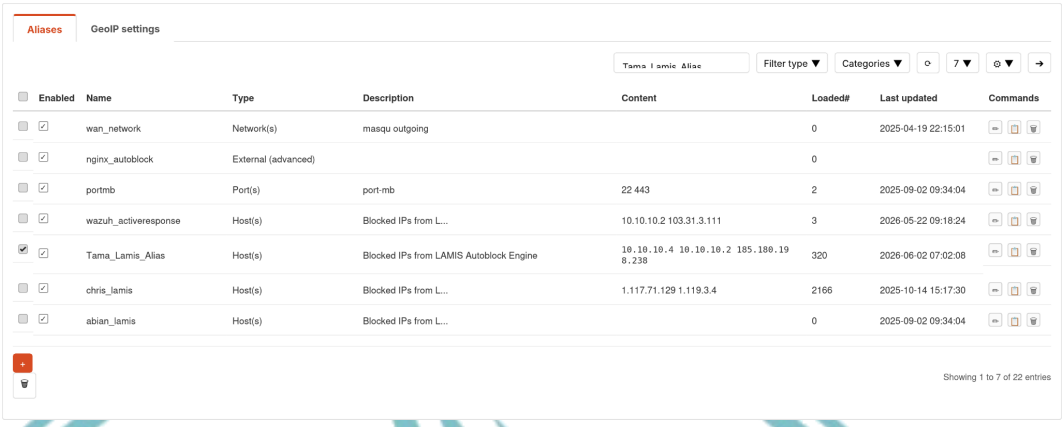
b. Contoh foto TP



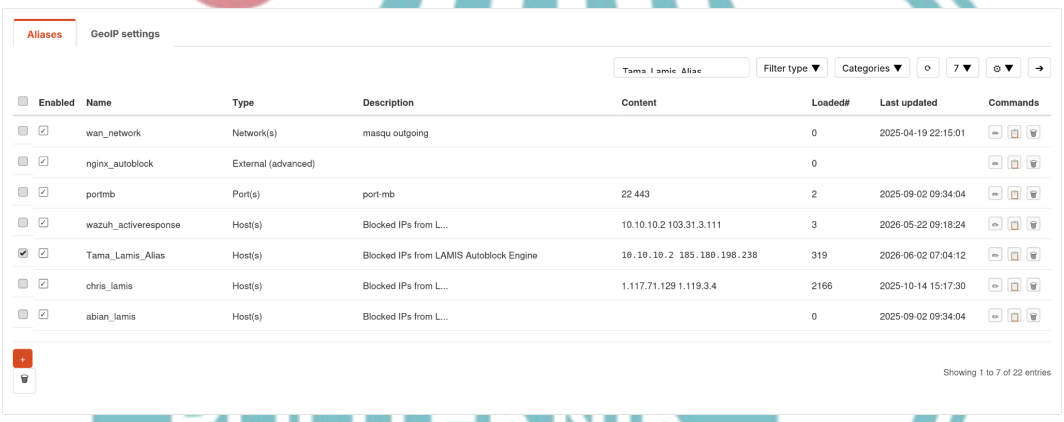
c. Contoh foto TN



d. Contoh foto FP



e. Contoh foto FN



B. *Full log* mentah APScheduler menunjukan penghitungan / sukses pemblokiran / sukses deteksi *whitelist* / : https://drive.google.com/file/d/128NjSP2ABl9K7WDr98jV08A0l8VSHx_3/view?usp=*drive_link*

LAMPIRAN 6: HASIL PENGUJIAN *PERFORMANCE TESTING*

System Performance Test Report
Generated on: 2026-05-31 10:34:49

A. Ringkasan Kinerja *Ingestion*

Scenario	*Target* EPS	*Ingestion* Cycles	Total Alerts Ingested	Average *Ingestion Time*	Peak *Ingestion Time*
Low Stress	10	20	5,919	1.0 s	1.0 s
Medium Stress	100	20	57,764	1.0 s	1.0 s
High Stress	500	20	287,443	2.35 s	3.0 s
Extreme Stress	1000	20	571,965	4.3 s	5.0 s

B. Ringkasan Pemanfaatan Sumber Daya Kontainer

Skenario: Low Stress

Container	Avg CPU (%)	Peak CPU (%)	Avg Memory	Peak Memory
flask-backend	0.1%	0.17%	574.8 MB	574.8 MB
apscheduler-worker	0.0%	0.01%	442.08 MB	442.1 MB
clickhouse-db	9.49%	28.94%	2.03 GB	2.09 GB
redis-cache	0.79%	1.07%	7.41 MB	7.62 MB
postgres-db	1.24%	10.56%	56.92 MB	56.94 MB

Skenario: Medium Stress

Container	Avg CPU (%)	Peak CPU (%)	Avg Memory	Peak Memory

flask-backend	0.1%	0.14%	574.8 MB	574.8 MB
apscheduler-worker	0.02%	1.03%	448.18 MB	449.5 MB
clickhouse-db	10.67%	83.4%	2.02 GB	2.09 GB
redis-cache	0.79%	1.6%	7.42 MB	7.62 MB
postgres-db	1.16%	9.73%	56.95 MB	59.01 MB

Skenario: High Stress

Container	Avg CPU (%)	Peak CPU (%)	Avg Memory	Peak Memory
flask-backend	0.11%	0.14%	574.8 MB	574.8 MB
apscheduler-worker	0.0%	0.02%	517.8 MB	542.7 MB
clickhouse-db	10.95%	70.62%	2.02 GB	2.11 GB
redis-cache	0.74%	0.9%	7.42 MB	7.62 MB
postgres-db	1.29%	9.47%	56.98 MB	60.64 MB

Skenario: Extreme Stress

Container	Avg CPU (%)	Peak CPU (%)	Avg Memory	Peak Memory
flask-backend	3.53%	40.69%	571.81 MB	574.8 MB
apscheduler-worker	0.02%	0.99%	635.4 MB	683.4 MB
clickhouse-db	22.42%	145.55%	1.82 GB	2.21 GB
redis-cache	0.78%	1.06%	7.44 MB	7.65 MB
postgres-db	1.25%	10.29%	56.9 MB	57.58 MB

C. Durasi Siklus *Ingestion* Mentah

(Tabel lengkap per siklus untuk setiap skenario — Low, Medium, High, Extreme Stress — sebagaimana hasil pengujian)

Scenario	Cycle #	*Start Time*	End *Time*	Alerts Ingested	*Processing* Duration
Low Stress	1	02:40:44	02:40:45	223	1.0s
Low Stress	2	02:41:14	02:41:15	300	1.0s
Low Stress	3	02:41:44	02:41:45	306	1.0s
Low Stress	4	02:42:14	02:42:15	299	1.0s
Low Stress	5	02:42:44	02:42:45	297	1.0s
Low Stress	6	02:43:14	02:43:15	305	1.0s
Low Stress	7	02:43:44	02:43:45	300	1.0s
Low Stress	8	02:44:14	02:44:15	308	1.0s
Low Stress	9	02:44:44	02:44:45	302	1.0s
Low Stress	10	02:45:14	02:45:15	305	1.0s
Low Stress	11	02:45:44	02:45:45	308	1.0s
Low Stress	12	02:46:14	02:46:15	243	1.0s
Low Stress	13	02:46:44	02:46:45	302	1.0s
Low Stress	14	02:47:14	02:47:15	305	1.0s
Low Stress	15	02:47:44	02:47:45	304	1.0s
Low Stress	16	02:48:14	02:48:15	302	1.0s
Low Stress	17	02:48:44	02:48:45	304	1.0s
Low Stress	18	02:49:14	02:49:15	302	1.0s
Low Stress	19	02:49:44	02:49:45	302	1.0s
Low Stress	20	02:50:14	02:50:15	302	1.0s

Medium Stress	1	02:52:44	02:52:45	2104	1.0s
Medium Stress	2	02:53:14	02:53:15	2906	1.0s
Medium Stress	3	02:53:44	02:53:45	2918	1.0s
Medium Stress	4	02:54:14	02:54:15	3029	1.0s
Medium Stress	5	02:54:44	02:54:45	2921	1.0s
Medium Stress	6	02:55:14	02:55:15	3002	1.0s
Medium Stress	7	02:55:44	02:55:45	2933	1.0s
Medium Stress	8	02:56:14	02:56:15	2951	1.0s
Medium Stress	9	02:56:44	02:56:45	2978	1.0s
Medium Stress	10	02:57:14	02:57:15	2948	1.0s
Medium Stress	11	02:57:44	02:57:45	2907	1.0s
Medium Stress	12	02:58:14	02:58:15	2548	1.0s
Medium Stress	13	02:58:44	02:58:45	2906	1.0s
Medium Stress	14	02:59:14	02:59:15	2908	1.0s

Medium Stress	15	02:59:44	02:59:45	3040	1.0s
Medium Stress	16	03:00:14	03:00:15	2948	1.0s
Medium Stress	17	03:00:44	03:00:45	2934	1.0s
Medium Stress	18	03:01:14	03:01:15	2932	1.0s
Medium Stress	19	03:01:44	03:01:45	3007	1.0s
Medium Stress	20	03:02:14	03:02:15	2944	1.0s
High Stress	1	03:04:44	03:04:46	11507	2.0s
High Stress	2	03:05:14	03:05:16	14564	2.0s
High Stress	3	03:05:44	03:05:46	15075	2.0s
High Stress	4	03:06:14	03:06:16	14534	2.0s
High Stress	5	03:06:44	03:06:46	12522	2.0s
High Stress	6	03:07:14	03:07:16	14516	2.0s
High Stress	7	03:07:44	03:07:46	14510	2.0s
High Stress	8	03:08:14	03:08:16	15053	2.0s
High Stress	9	03:08:44	03:08:47	14638	3.0s
High Stress	10	03:09:14	03:09:17	15078	3.0s
High Stress	11	03:09:44	03:09:46	15011	2.0s
High Stress	12	03:10:14	03:10:16	14504	2.0s
High Stress	13	03:10:44	03:10:47	14542	3.0s

High Stress	14	03:11:14	03:11:17	15090	3.0s
High Stress	15	03:11:44	03:11:46	12066	2.0s
High Stress	16	03:12:14	03:12:17	15180	3.0s
High Stress	17	03:12:44	03:12:47	14505	3.0s
High Stress	18	03:13:14	03:13:16	14519	2.0s
High Stress	19	03:13:44	03:13:47	15013	3.0s
High Stress	20	03:14:14	03:14:16	15016	2.0s
Extreme Stress	1	03:16:44	03:16:48	23637	4.0s
Extreme Stress	2	03:17:14	03:17:18	29120	4.0s
Extreme Stress	3	03:17:44	03:17:48	29007	4.0s
Extreme Stress	4	03:18:14	03:18:17	25036	3.0s
Extreme Stress	5	03:18:44	03:18:49	29004	5.0s
Extreme Stress	6	03:19:14	03:19:18	30079	4.0s
Extreme Stress	7	03:19:44	03:19:48	29016	4.0s
Extreme Stress	8	03:20:14	03:20:19	32084	5.0s
Extreme Stress	9	03:20:44	03:20:48	29005	4.0s
Extreme Stress	10	03:21:14	03:21:18	25335	4.0s

Extreme Stress	11	03:21:44	03:21:49	29004	5.0s
Extreme Stress	12	03:22:14	03:22:18	30004	4.0s
Extreme Stress	13	03:22:44	03:22:49	30081	5.0s
Extreme Stress	14	03:23:14	03:23:18	29857	4.0s
Extreme Stress	15	03:23:44	03:23:48	29109	4.0s
Extreme Stress	16	03:24:14	03:24:19	30055	5.0s
Extreme Stress	17	03:24:44	03:24:49	29005	5.0s
Extreme Stress	18	03:25:14	03:25:17	24446	3.0s
Extreme Stress	19	03:25:44	03:25:49	30079	5.0s
Extreme Stress	20	03:26:14	03:26:19	29002	5.0s

D. Referensi Data dan Artefak Pengujian

1. Dataset *Monitoring* Resource (CSV)
   Link Google Drive:
   https://drive.google.com/file/d/1Nweol7eQEC01XUMoWXPfrl8WM6LF0GIe/view?usp=*drive_link*
2. Script Generator *Log*
   File: *log_generator*.py
   Link Repository / Drive:
   https://drive.google.com/file/d/1l4cLQCsDo6Thw7iVVnI5qDEKZHu-*zwKk*/view?usp=*drive_link*
3. Script *Monitoring Container Performance*
   File: *monitor_performance*.py
   Link Repository / Drive:

https://drive.google.com/file/d/1OdbATjldrT85yHRUJPw6JCrnpGSA3BPP/view?usp=*drive_link*

4. Script *Monitoring Ingestion Performance*
   File: *monitor_ingestion*.py
   Link Repository / Drive:
   https://drive.google.com/file/d/1vqwSEGREfutAhZYeAipZ5LX1zXP8-YMs/view?usp=sharing