PROTOTIPE SECURITY OPERATIONS CENTER BERBASIS INTEGRASI SOFTWARE OPEN SOURCE UNTUK OPTIMASI DETEKSI DAN RESPONS ANCAMAN SIBER

SKRIPSI
Diajukan untuk memenuhi sebagian persyaratan mendapatkan gelar Strata Satu Program Studi Informatika

DIMAS GALIH LAKSONO
22EO10019

PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER
UNIVERSITAS NAHDLATUL ULAMA AL GHAZALI
CILACAP
2026

PROTOTIPE SECURITY OPERATIONS CENTER BERBASIS INTEGRASI SOFTWARE OPEN SOURCE UNTUK OPTIMASI DETEKSI DAN RESPONS ANCAMAN SIBER

SKRIPSI
Diajukan untuk memenuhi sebagian persyaratan mendapatkan gelar Strata Satu Program Studi Informatika

DIMAS GALIH LAKSONO
22EO0019

PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER
UNIVERSITAS NAHDLATUL ULAMA AL GHAZALI
CILACAP
2026

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya menyatakan bahwa penulisan skripsi dengan judul "Prototipe Security Operations Center Berbasis Integrasi Software Open Source Untuk Optimasi Deteksi Dan Respons Ancaman Siber" adalah hasil karya saya dengan arahan dari pembimbing dan belum diajukan kepada pihak manapun. Sumber informasi yang dikutip dalam skripsi ini telah dicantumkan dalam Daftar Pustaka.

Demikian pernyataan ini dibuat dengan sebenarnya. Apabila di kemudian hari terdapat ketidaksesuaian dalam pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Cilacap, Bulan Tahun

Materai
Rp. 10.000

Dimas Galih Laksono
22EO10019

LEMBAR PENGESAHAN

Skripsi Saudara

Nama	:	Dimas Galih Laksono
NIM	:	22EO10019
Fakultas/Prodi	:	Fakultas MIKOM /Informatika
Judul	:	Prototipe Security Operations Center Berbasis Integrasi Software Open Source Untuk Optimasi Deteksi Dan Respons Ancaman Siber

Telah disidangkan oleh Dewan Penguji Fakultas Matematika dan Ilmu Komputer Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap pada hari / tanggal :

Hari, Tanggal Bulan Tahun

Dan dapat diterima sebagai pemenuhan tugas akhir mahasiswa Program Strata 1 (S.1) Nama Prodi (Mat/Kom) Fakultas Matematika dan Ilmu Komputer (FMIKOM) pada Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap.

Tempat, Tanggal Bulan Tahun

Dewan Sidang

Ketua	Sekretaris
Nama NIDN	Nama NIDN
Penguji 1	Penguji 2
Nama NIDN	Nama NIDN
Pembimbing I	Pembimbing II
Nama NIDN	Nama NIDN

Mengetahui,
Dekan Fakultas Matematika dan Ilmu Komputer

M.T Abdul Aziz Zein, M.Kom.

NIDN. 0613065801

HALAMAN NOTA KONSULTAN

Nama Dosen

Dosen Fakultas Matematika dan Ilmu Komputer
Universitas Nahdlatul Ulama Al Ghazali Cilacap

NOTA KONSULTAN
Hal : Skripsi Saudara/i Nama Mahasiswa
Lampiran : -

Kepada
Yth. Dekan Fakultas Matematika dan Ilmu Komputer
Universitas Nahdlatul Ulama Al Ghazali Cilacap
di Cilacap

Assalamu'alaikum Wr. Wb.
Setelah membaca, memeriksa dan melakukan perbaikan seperlunya maka skripsi saudara/i:

Nama	: Dimas Galih Laksono
NIM	: 22EO10019
Prodi	: Informatika
Judul	: Prototipe Security Operations Center Berbasis Integrasi Software Open Source Untuk Optimasi Deteksi Dan Respons Ancaman Siber

Dapat diajukan ke Fakultas Matematika dan Ilmu Komputer, Universitas Nahdlatul Ulama Al Ghazali Cilacap untuk memenuhi syarat memperoleh gelar Strata Satu (S1).
Wassalamu'alaikum Wr. Wb.

Cilacap, Hari Bulan Tahun
Konsultan

Nama Dosen
NIDN

NOTA PEMBIMBING

Cilacap, 23 juni 2026

Kepada Yth :
Fakultas Matematika dan Komputer (FMIKOM)
UNUGHA Cilacap
di Cilacap

Assalamu'alaikum Wr. Wb.
Setelah melakukan bimbingan, telaah, arahan dan koreksi tahap penulisan skripsi saudari:

Nama	: Dimas Galih Laksono
NIM	: 22EO10019
Fakultas	: Matematika dan Ilmu Komputer
Prodi	: Informatika
Judul	: Prototipe Security Operations Center Berbasis Integrasi Software Open Source Untuk Optimasi Deteksi Dan Respons Ancaman Siber

Kami berpendapat bahwa skripsi tersebut sudah dapat diajukan ke sidang skripsi. Bersamaan ini kami kirimkan skripsi tersebut, semoga dapat segera disidangkan.

Atas perhatiannya kami ucapkan terima kasih.
Wassalamu'alaikum Wr. Wb.

Mengetahui,

Pembimbing I	Pembimbing II
Abdul Haq, S.E., M.Cs	Sani Nurmalasari, M.Kom
NIDN.0606067701	NIDN.4341767668231113

HALAMAN MOTO

“Dan jika kamu membalas, maka balaslah dengan (balasan) yang sama dengan siksaan yang ditimpakan kepadamu. Tetapi jika kamu bersabar, sesungguhnya itulah yang lebih baik bagi orang yang sabar”

(QS. An-Nahl : 126)

“angan takut untuk melangkah lambat, takutlah jika kamu hanya berdiri diam tanpa pernah bergerak sama sekali”

(Confucius)

“Jika kamu merasa duniamu kejam dan sulit, jangan menyerah. Teruslah berjuang, karena kamu tidak akan pernah tahu hasil akhir dari sebuah perjuangan sebelum kamu benar-benar menyelesaikannya”

(Eren Yeager)

HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT, Rabb semesta alam yang senantiasa memberikan karunia sehingga penulis mampu menyelesaikan penulisan skripsi ini. Karya ini saya persembahkan kepada :

1. Orang tua yang selalu mendidik saya, memberikan do'a, dukungan, nasihat dan semangat yang tiada henti.
2. Keluarga FMIKOM Angkatan 2022 yang selalu memberikan keceriaan, kebersamaan dan motivasi.
3. Keluarga program studi Teknik Informatika, Sistem Informasi, Matematika yang saya banggakan
4. Seluruh teman UNUGHA yang telah memberikan do'a, dukungan, dan semangat.

KATA PENGANTAR

Puji syukur kepada Allah SWT atas segala limpahan nikmat dan karuniaNya, sehingga penulis dapat menyelesaikan penulisan skripsi. Sholawat dan salam senantiasa penulis haturkan kepada Nabi Muhammad SAW sebagai pembimbing seluruh umat manusia.

Skripsi ini tidak akan selesai tanpa adanya bantuan dari banyak pihak, karena itu penulis menyampaikan terima kasih kepada :

1. Bapak M.T Abdul Aziz Zein, M.Kom, Dekan FMIKOM UNUGHA
2. Bapak Abdul Haq, S.E., M.Cs, pembimbing I yang telah dengan sabar memberikan bimbingan dan petunjuk dalam menyelesaikan skripsi ini.
3. Ibu Sani Nurmalasari, M.Kom, pembimbing II yang telah dengan sabar memberikan bimbingan dan petunjuk dalam menyelesaikan skripsi ini.
4. Ibu Riski Aspriyani, M.Pd, sebagai Pembimbing Akademis
5. Bapak-Ibu dosen Program Studi Informatika FMIKOM UNUGHA
6. Semua pihak yang tidak dapat kami sebutkan satu persatu

Semoga Allah SWT membalas jerih payah dan pengorbanan yang telah diberikan dengan balasan yang lebih baik. Amiin. Penulis berharap semoga karya kecil ini bermanfaat bagi pembaca.

Cilacap, 23 Juni 2026

Dimas Galih Laksono

ABSTRAK

DIMAS GALIH LAKSONO. Prototipe Security Operations Center Berbasis Integrasi Software Open Source Untuk Optimasi Deteksi Dan Respons Ancaman Siber. Dibimbing oleh Abdul Haq, S.E., M.Cs. dan Sani Nurmalasari, M.Kom.

Serangan siber tidak lagi menjadi ancaman yang bersifat potensial, melainkan telah menjadi realitas yang dihadapi hampir setiap institusi. Ancaman siber tidak hanya berpotensi menyebabkan gangguan operasional dan kerusakan sistem, tetapi juga dapat mengakibatkan kebocoran data serta kerugian *finansial*. *Security Operations Center* merupakan salah satu solusi yang dapat digunakan untuk melakukan pemantauan, deteksi, analisis, dan respons terhadap ancaman siber secara terpusat. Namun, implementasi SOC konvensional umumnya membutuhkan biaya investasi yang tinggi sehingga sulit diterapkan oleh organisasi dengan keterbatasan sumber daya.

Dalam penelitian ini dilakukan beberapa tahapan, yaitu studi literatur, perancangan arsitektur dan alur kerja SOC, implementasi sistem, pengujian, analisis hasil pengujian, serta penarikan kesimpulan. Prototipe SOC dibangun dengan mengintegrasikan berbagai *software open source*, yaitu Wazuh sebagai *Intrusion Detection System (IDS), Endpoint Detection and Response (EDR)* dan *Intrusion Prevention System (IPS)*, Graylog sebagai sistem manajemen *log (centralized log management),* MISP sebagai platform *threat intelligence*, serta VirusTotal sebagai *Extended Detection and Response (XDR)* untuk mendukung analisis dan korelasi ancaman keamanan. Selain itu, Telegram Bot dan *Email Notification* digunakan sebagai media notifikasi keamanan secara *real-time*. Pengujian sistem dilakukan terhadap berbagai mekanisme keamanan, meliputi *File Integrity Monitoring (FIM),* Docker *Monitoring*, *Active Response*, Anti *Malware*, *Threat Intelligence*, serta simulasi serangan yang mengacu pada OWASP Top 10 dan MITRE ATT&CK guna mengevaluasi kemampuan deteksi dan respons terhadap ancaman siber.

Hasil penelitian menunjukkan bahwa prototipe *Security Operations Center* berbasis integrasi *software open source* berhasil diimplementasikan dan seluruh komponen dapat terintegrasi dengan baik. Sistem mampu melakukan *monitoring log* secara terpusat, mendeteksi berbagai aktivitas dan ancaman keamanan, memanfaatkan *threat intelligence*, serta menjalankan respons otomatis terhadap insiden keamanan. Berdasarkan hasil pengujian yang dilakukan, sistem menunjukkan tingkat akurasi deteksi yang tinggi dan mampu menghasilkan *alert* serta notifikasi keamanan secara *near real-time*. Dengan demikian, prototipe SOC yang dibangun dapat menjadi alternatif solusi yang efektif, terintegrasi, dan lebih terjangkau dalam mendukung proses deteksi dan respons ancaman siber.

Kata kunci : *Cyber Security, Cyber Defense, Security Operations Center, Blue Team.*

ABSTRACT

DIMAS GALIH LAKSONO. *Prototype of a Security Operations Center Based on Integrated Open-Source Software for Optimizing Cyber Threat Detection and Response*. *Supervised by 1st* Abdul Haq, S.E., M.Cs. and *2nd* Sani Nurmalasari, M.Kom.

Cyberattacks are no longer merely potential threats but have become a reality faced by nearly every institution. Cyber threats not only have the potential to cause operational disruptions and system damage, but can also lead to data breaches and financial losses. A Security Operations Center (SOC) is one of the solutions that can be used to centrally monitor, detect, analyze, and respond to cyber threats. However, conventional SOC implementations generally require high investment costs, making them difficult to adopt for organizations with limited resources.

This study was conducted through several stages, namely literature review, SOC architecture and workflow design, system implementation, testing, analysis of test results, and conclusion drawing. The SOC prototype was developed by integrating various open-source software tools, namely Wazuh as an Intrusion Detection System (IDS), Endpoint Detection and Response (EDR) and Intrusion Prevention System (IPS), Graylog as a centralized log management system, MISP as a threat intelligence platform, and VirusTotal as an Extended Detection and Response (XDR) component to support threat analysis and correlation. In addition, Telegram Bot and Email Notification were used as real-time security alerting mechanisms. System testing was carried out across various security mechanisms, including File Integrity Monitoring (FIM), Docker Monitoring, Active Response, Anti-Malware, and Threat Intelligence, as well as attack simulations based on the OWASP Top 10 and MITRE ATT&CK frameworks to evaluate detection and response capabilities against cyber threats.

The results of the study indicate that the prototype of the Security Operations Center based on integrated open-source software was successfully implemented and all components were effectively integrated. The system is capable of centralized log monitoring, detecting various security activities and threats, utilizing threat intelligence, and executing automated responses to security incidents. Based on the testing results, the system demonstrated a high level of detection accuracy and was able to generate alerts and security notifications in near real-time. Therefore, the developed SOC prototype can serve as an effective, integrated, and more affordable alternative solution to support cyber threat detection and response processes.

Keywords*: Cybersecurity, Cyber Defense, Security Operations Center, Blue Team.*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I PENDAHULUAN

A. Latar Belakang

Serangan siber tidak lagi menjadi ancaman yang bersifat potensial, melainkan telah menjadi realitas yang dihadapi hampir setiap institusi. Berbagai organisasi, mulai dari instansi pemerintahan, lembaga pendidikan, hingga pelaku usaha kecil dan menengah (UKM), kini bergantung pada sistem digital dalam menjalankan operasionalnya. Kondisi ini secara tidak langsung memperluas permukaan serangan *(attack surface)* yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab untuk mengeksploitasi kerentanan sistem[1].

Dalam perspektif keilmuan teknologi informasi, khususnya pada bidang keamanan siber, peningkatan intensitas dan kompleksitas serangan ini menuntut adanya mekanisme pertahanan yang tidak hanya bersifat reaktif, tetapi juga proaktif. Serangan siber modern cenderung dilakukan secara terorganisasi, menggunakan teknik yang semakin canggih, serta memiliki tujuan yang beragam, mulai dari pencurian data hingga gangguan layanan. Oleh karena itu, diperlukan suatu pendekatan yang mampu mengintegrasikan proses deteksi, analisis, dan respons ancaman secara efektif dan berkelanjutan[2].

Kerugian akibat ancaman siber tidak hanya terbatas pada kerusakan infrastruktur teknologi, tetapi juga dapat melumpuhkan operasional, merusak reputasi organisasi, bahkan bisa menimbulkan kerugian ekonomi. Hal ini diperkuat dengan laporan *IBM Cost of a Data Breach Report* tahun 2024, yang menunjukkan rata-rata kerugian yang dialami oleh organisasi akibat satu insiden kebocoran data mencapai 4,9 juta USD, jumlah ini meningkat sebesar 10% dari tahun sebelumnya dan menjadikan angka tertinggi yang tercatat[3]. Data ini menegaskan bahwa ancaman siber bukan hanya isu teknis, melainkan juga risiko strategis yang dapat memengaruhi keberlangsungan organisasi.

Dalam menghadapi ancaman siber yang semakin kompleks dan sulit diprediksi, keberadaan *Security Operations Center* (SOC) menjadi kebutuhan yang tidak dapat diabaikan, karena berfungsi sebagai pusat terintegrasi yang mampu melakukan pemantauan, deteksi, analisis, hingga respons terhadap insiden keamanan secara berkelanjutan, sehingga organisasi dapat mengidentifikasi ancaman lebih cepat sebelum berkembang menjadi serangan yang merugikan. Namun di sisi lain, penerapan SOC secara konvensional masih menjadi tantangan besar, terutama karena tingginya biaya investasi yang harus dikeluarkan, mulai dari pengadaan infrastruktur perangkat keras, lisensi perangkat lunak, hingga kebutuhan sumber daya manusia yang memiliki keahlian khusus di bidang keamanan siber[4].

Kondisi ini menimbulkan kesenjangan yang cukup signifikan, di mana institusi dengan keterbatasan anggaran seperti sekolah, perguruan tinggi, dan organisasi publik berskala kecil tetap berada dalam posisi rentan terhadap serangan siber, tetapi belum memiliki kemampuan yang memadai untuk membangun sistem pertahanan yang optimal.

Tingginya biaya investasi justru membuka ruang bagi lahirnya solusi yang lebih inovatif. Kemajuan teknologi serta dukungan komunitas *open source* memungkinkan pengembangan sistem SOC yang andal melalui pendekatan integrasi *software* berbasis *open source*. Saat ini, tersedia beragam *software open source* yang terbukti mampu menjalankan fungsi utama SOC secara efektif dan fleksibel [5], [6]. Pemanfaatan ekosistem teknologi *open source* menjadi faktor penting dalam mewujudkan keamanan siber yang lebih terjangkau, sehingga organisasi yang sebelumnya terkendala biaya dapat menerapkan sistem keamanan yang memadai.

Oleh karena itu, penulis mengambil topik penelitian "PROTOTIPE *SECURITY OPERATIONS CENTER* BERBASIS INTEGRASI *SOFTWARE OPEN SOURCE* UNTUK OPTIMASI DETEKSI DAN RESPONS ANCAMAN SIBER" sebagai tugas akhir. Penelitian ini difokuskan pada evaluasi prototipe *Security Operations Center (SOC)* berbasis integrasi perangkat lunak *open source* untuk deteksi dan respons ancaman siber secara terpusat. Pendekatan ini bertujuan untuk membangun SOC yang tidak hanya mampu melakukan deteksi dan notifikasi ancaman secara *real-time*, tetapi juga memberikan arsitektur yang modular, skalabel, dan lebih hemat biaya dibandingkan SOC berbasis lisensi komersial.

B. Rumusan Masalah

Berdasarkan latar belakang, maka didapatkan perumusan masalah sebagai berikut:

1. Bagaimana melakukan perancangan arsitektur dan alur kerja prototipe SOC berbasis integrasi *software open source* supaya memenuhi fungsi inti keamanan siber?
2. Bagaimana melakukan implementasi dan interpretasi hasil pengujian rancangan arsitektur SOC untuk optimasi deteksi dan respons ancaman siber?

C. Batasan Masalah

Adapun batasan masalah dari penelitian yang akan dilakukan adalah sebagai berikut:

1. Software *open source* yang digunakan dalam pengembangan prototipe *Security Operations Center (SOC)* terbatas pada Wazuh versi 4.14 sebagai

Security Information and Event Management (SIEM), Graylog versi 7.0.3 sebagai *log management,* serta MISP versi 2.5 sebagai platform *threat intelligence*, yang seluruhnya berjalan pada sistem operasi Linux Ubuntu Server versi 24.04 LTS.

2. Penelitian pada tiga fungsi utama SOC: pemantauan, deteksi ancaman, dan respons insiden.
3. Penelitian ini tidak membahas aspek keamanan fisik, kebijakan dan tata kelola organisasi, sertifikasi keamanan, maupun audit dan kepatuhan terhadap standar atau regulasi tertentu.
4. Pengujian prototipe SOC dilaksanakan melalui penerapan skenario serangan yang mengacu pada kerangka kerja MITRE ATT&CK dan OWASP Top 10 sebagai dasar evaluasi fungsionalitas sistem, tanpa mencakup seluruh kemungkinan teknik serangan yang ada.

D. Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan sebagai berikut:

1. Merancang arsitektur dan alur kerja prototipe *Security Operations Center (SOC)* berbasis integrasi perangkat lunak sumber terbuka yang mampu memenuhi fungsi inti keamanan siber, meliputi pemantauan, deteksi ancaman, dan respons insiden.
2. Mengimplementasikan serta mengevaluasi prototipe SOC yang telah dirancang melalui pengujian untuk menganalisis efektivitasnya dalam mengoptimalkan deteksi dan respons terhadap ancaman siber.

E. Manfaat Penelitian

Terdapat berbagai manfaat yang dapat diperoleh melalui penelitian ini. Berikut beberapa manfaat penelitian ini:

1. Menyediakan prototipe *Security Operations Center (SOC)* berbasis integrasi perangkat lunak sumber terbuka yang mampu mendukung proses deteksi dan respons ancaman siber secara *real-time.*
2. Memberikan alternatif solusi yang efisien dan terjangkau bagi organisasi dalam meningkatkan kemampuan deteksi dini dan respons terhadap ancaman siber tanpa bergantung pada solusi komersial berbiaya tinggi.
3. Mendukung upaya peningkatan ketahanan siber nasional, khususnya pada sektor-sektor yang rentan terhadap serangan siber namun memiliki keterbatasan sumber daya.

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Dalam penyusunan penelitian ini, berbagai penelitian terdahulu dijadikan sebagai landasan referensi. Referensi tersebut mencakup jurnal dan karya ilmiah yang memiliki relevansi dengan topik yang dikaji serta kesamaan tujuan dalam pengembangan *Security Operations Center*.

Tabel 1 Penelitian Terkait

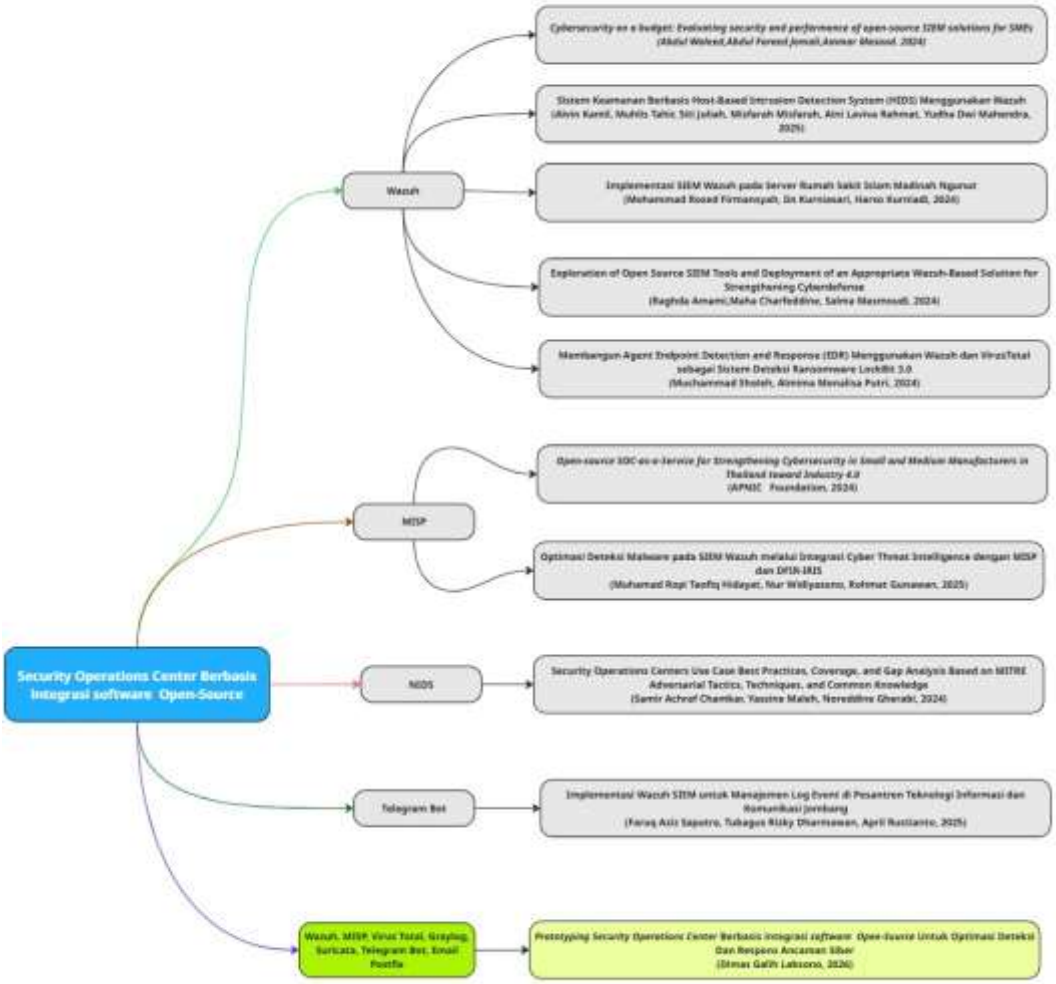
No	Penulis (Tahun)	Keterangan
1	Saputra, F. *et al.* (2023)	**Judul:** Implementasi Wazuh SIEM untuk Manajemen *Log Event* di Pesantren Teknologi Informasi dan Komunikasi Jombang **Hasil:** Sistem berhasil mengirimkan notifikasi *real-time* untuk berbagai insiden keamanan seperti serangan *brute force, Denial of Service (DoS),* dan *SQL Injection*, sehingga meningkatkan respons dan kesadaran keamanan di lingkungan pesantren..
2	Sholeh, M. et al. (2024)	**Judul:** Membangun *Agent Endpoint Detection and Response (EDR)* Menggunakan Wazuh dan VirusTotal sebagai Sistem Deteksi Ransomware LockBit 3.0 **Hasil:** Integrasi Wazuh dengan API VirusTotal berhasil mendeteksi dan menghapus ransomware dalam waktu rata-rata 0,7 detik, menunjukkan efektivitas tinggi dalam deteksi dan respons ancaman berbasis EDR.
3	Manzoor, J. *et al.* (2024)	**Judul:** *Cybersecurity on a budget: Evaluating security and performance of open source SIEM solutions for SMEs* **Hasil:** Evaluasi menunjukkan bahwa Wazuh dan SIEMonster *(Community Edition)* memiliki kapabilitas terbaik dari sisi keamanan dan performa, sehingga dinilai paling sesuai untuk implementasi SIEM dengan anggaran terbatas.
4	APNIC Foundation (2024)	**Judul:** *Open source SOC-as-a-Service for Strengthening Cybersecurity in Small and Medium Manufacturers in Thailand toward Industry 4.0*

No	Penulis (Tahun)	Keterangan
		Hasil: Integrasi Wazuh dengan MISP memungkinkan ekstraksi *Indicators of Compromise (IoC)* secara otomatis. Pengujian menunjukkan peningkatan kesadaran keamanan serta penurunan risiko kerentanan pada lingkungan UKM manufaktur.
5	Chamkar, S. *et al.* (2024)	**Judul:** *Security Operations Centers Use Case Best Practices, Coverage, and Gap Analysis Based on MITRE Adversarial Tactics, Techniques, and Common Knowledge* **Hasil:** *Playbook* respons insiden berbasis MITRE ATT&CK dapat diimplementasikan secara efektif pada SOC yang menggunakan SIEM/SOAR *open source*, sehingga meningkatkan kesiapan dan konsistensi respons terhadap serangan siber.
6	Amami, R. et al. (2024)	**Judul:** *Exploration of Open Source SIEM Tools and Deployment of an Appropriate Wazuh-Based Solution for Strengthening Cyberdefense* **Hasil:** Penelitian menunjukkan bahwa penggunaan Wazuh sebagai solusi SIEM berbasis *open source* efektif dalam meningkatkan kemampuan pertahanan siber melalui pemantauan *log*, deteksi ancaman, serta analisis keamanan secara terpusat.
7	Owolafe, O. & Alabi, A. J. (2024)	**Judul:** *Comparative Analysis of Wazuh, Graylog, and ELK Stack for Cyber Threat Detection and Response* **Hasil:** Hasil perbandingan menunjukkan bahwa Wazuh, Graylog, dan ELK Stack masing-masing memiliki keunggulan dalam deteksi dan respons ancaman, dengan Wazuh unggul dalam integrasi keamanan, sedangkan ELK Stack dan Graylog lebih fleksibel dalam pengolahan dan visualisasi *log*.
8	Firmansyah, M. R. et al. (2024)	**Judul:** Implementasi SIEM Wazuh pada Server Rumah Sakit Islam Madinah Ngunut **Hasil:** Integrasi Wazuh dengan sistem notifikasi (Telegram Bot) mampu meningkatkan monitoring keamanan secara real-time serta mempercepat respons terhadap insiden keamanan
9	Hidayat, M. R. T. *et al.* (2025)	**Judul:** Optimasi Deteksi *Malware* Pada SIEM Wazuh Melalui Integrasi *Cyber Threat Intelligence* Dengan MISP Dan DFIR-IRIS

No	Penulis (Tahun)	Keterangan
		Hasil: Integrasi Wazuh dengan MISP dan DFIR-IRIS terbukti meningkatkan efektivitas deteksi *malware*, mempercepat respons insiden, serta memperbaiki manajemen informasi ancaman secara keseluruhan.
10	Ang, S. et al. (2025)	**Judul:** *Utilizing IDS and IPS to Improve Cybersecurity Monitoring Process* **Hasil:** Implementasi IDS dan IPS mampu meningkatkan efektivitas proses pemantauan keamanan dengan mendeteksi aktivitas mencurigakan secara lebih cepat serta memberikan perlindungan berlapis terhadap ancaman siber.
11	Kamil, A. T. *et al.* (2025)	**Judul:** Sistem Keamanan Berbasis *Host-Based Intrusion Detection System (HIDS)* Menggunakan Wazuh **Hasil:** Wazuh berhasil digunakan untuk memantau perubahan file dan aktivitas mencurigakan pada *host*, sehingga meningkatkan keamanan server dan mendukung konsep SOC berbasis *open source* tanpa biaya lisensi.

Penelitian ini memiliki perbedaan yang signifikan dibandingkan dengan studi-studi sebelumnya karena berfokus pada pengembangan model *Security Operations Center (SOC)* berbasis integrasi *software open source* yang bersifat komprehensif dan terukur. Model yang diusulkan mengintegrasikan komponen Wazuh, Graylog, dan MISP, serta memanfaatkan mekanisme *enrichment* intelijen ancaman dan notifikasi *real-time* untuk mendukung deteksi, *triase*, dan respons insiden secara efisien.

Selain itu, penelitian ini juga menekankan pada aspek evaluasi kinerja dan efektivitas operasional SOC secara sistematis. Pengujian dilakukan dengan menggunakan skenario serangan simulatif yang mencakup berbagai vektor ancaman, termasuk *malware*, *intrusion attempts*, dan serangan berbasis jaringan. Hasil pengujian menunjukkan bahwa integrasi antara Wazuh, Graylog, dan MISP tidak hanya meningkatkan kemampuan deteksi dini, tetapi juga mempercepat proses *triase* insiden melalui otomatisasi korelasi *log* dan *enrichment* intelijen ancaman.



Gambar 1 *Mind Map* Penelitian Sebelumnya

B. Landasan Teori

1. *Security Operations Center (SOC)*

Security Operations Center adalah lokasi tempat *security professional* bertanggung jawab untuk memantau, mendeteksi, menganalisis, dan merespons ancaman keamanan siber secara berkelanjutan. SOC berfungsi sebagai benteng utama dalam mempertahankan keamanan informasi melalui pengawasan lalu lintas jaringan dan aktivitas sistem secara *real-time*, serta pengelolaan insiden keamanan dari tahap awal hingga mitigasi, dengan tujuan menjaga integritas, ketersediaan, dan kerahasiaan aset digital organisasi[7], [8].

Gambar 2 CIA triad

Dalam kerangka operasionalnya, SOC menggabungkan tiga komponen utama, yaitu *people*, *process*, dan *technology* yang didukung oleh tata kelola dan kepatuhan terhadap standar *governance* dan *compliance* untuk meningkatkan postur keamanan organisasi secara keseluruhan[7], [9], [10].

Gambar 3 Three Pillars of Cyber Security

2. Prototipe

Prototipe adalah pendekatan dalam pengembangan sistem atau perangkat lunak dengan membuat model awal dari sistem yang diusulkan untuk diuji dan dievaluasi sebelum pengembangan penuh dilakukan. Prototipe berfungsi sebagai representasi sementara dari produk akhir yang membantu pengembang dan pemangku kepentingan memahami kebutuhan, fungsi, dan interaksi sistem secara lebih jelas serta memperoleh umpan balik untuk penyempurnaan[11]. Proses ini bersifat iteratif, di mana prototipe dibuat, diuji, dan direvisi berulang kali hingga sesuai dengan kebutuhan

pengguna, sehingga dapat mengurangi risiko kesalahan desain dan mempercepat validasi sebelum implementasi akhir.

3. *Software Open Source*

   Software open source adalah perangkat lunak yang kode sumbernya tersedia secara bebas untuk umum sehingga dapat dilihat, dipelajari, dimodifikasi, dan didistribusikan kembali oleh siapapun sesuai dengan lisensi *open source* yang berlaku[12]. Pendekatan ini mendorong transparansi, kolaborasi, dan inovasi dari berbagai latar belakang pengembang untuk berkontribusi dalam pengembangan dan perbaikan perangkat lunak[13].

4. Optimasi

   Optimasi didefinisikan sebagai suatu pendekatan sistematis yang bertujuan untuk mencapai hasil yang paling optimal pada suatu sistem, melalui pemanfaatan sumber daya yang tersedia secara tepat. Dalam ranah ilmu komputer, matematika, maupun manajemen, optimasi diarahkan untuk memaksimalkan kinerja, kualitas, atau keuntungan, sekaligus meminimalkan biaya, waktu, dan penggunaan sumber daya, dengan memperhatikan berbagai kendala yang berlaku[14].

5. Deteksi ancaman siber

   Deteksi ancaman siber adalah proses mengidentifikasi, memantau, dan menganalisis aktivitas mencurigakan atau berbahaya pada sistem, jaringan, maupun aplikasi yang berpotensi mengancam keamanan informasi. Proses ini bertujuan untuk menemukan indikasi serangan seperti *malware*, *intrusion*, kebocoran data, atau penyalahgunaan akses sedini mungkin dengan menggunakan teknologi dan metode seperti IDS/IPS, SIEM, EDR, analisis *log*, serta *threat intelligence* sehingga organisasi dapat mengambil tindakan pencegahan dan respons yang tepat untuk meminimalkan dampak serangan[15], [16].

6. Respons ancaman siber

   Respons ancaman siber adalah serangkaian tindakan terstruktur yang dilakukan untuk menangani, mengendalikan, dan memulihkan sistem setelah adanya indikasi ancaman atau insiden keamanan siber. Proses ini mencakup identifikasi dan analisis insiden, penahanan *(containment)* untuk mencegah penyebaran dampak, penghapusan ancaman *(eradication)*, serta pemulihan sistem agar dapat beroperasi normal kembali[16], [17]. Respons ancaman siber juga melibatkan dokumentasi, evaluasi, dan perbaikan berkelanjutan guna meningkatkan kesiapan keamanan di masa depan serta meminimalkan dampak kerugian terhadap organisasi.

7. Efektivitas

   Efektivitas merupakan ukuran kemampuan suatu sistem atau proses dalam mencapai tujuan yang telah ditetapkan. Dalam keamanan siber, efektivitas merujuk pada kemampuan solusi keamanan dalam mendeteksi, mencegah, dan menanggapi ancaman secara tepat sesuai kebutuhan organisasi. Tingkat efektivitas ini tercermin dari kecepatan dan ketepatan dalam mendeteksi serta menangani insiden keamanan, yang umumnya diukur melalui indikator seperti *Mean Time to Detect (MTTD), Mean Time to Respond (MTTR),* dan tingkat akurasi deteksi[18].

8. Wazuh

   Wazuh adalah platform *open source Security Information and Event Management (SIEM)* yang berfungsi untuk memantau, mendeteksi, dan menganalisis ancaman siber melalui pengelolaan *log* secara terpusat serta pemantauan aktivitas sistem secara *real-time,* sehingga meningkatkan visibilitas dan kemampuan respons terhadap insiden keamanan pada lingkungan TI[19].

9. Malware Information Sharing Platform (MISP)

   MISP adalah platform intelijen ancaman *open source* yang dirancang untuk mengumpulkan, menyimpan, berbagi, dan mengaitkan *Indicator of Compromise (IoC)* serta informasi ancaman siber secara terstruktur. Platform ini membantu organisasi meningkatkan kemampuan deteksi, respons, dan kolaborasi terhadap ancaman[15]. MISP juga mendukung pertukaran intelijen ancaman secara otomatis dan terstandarisasi, serta dapat diintegrasikan dengan sistem lain seperti SIEM atau IDS untuk memperkuat keamanan siber secara menyeluruh[20].

10. Graylog

    Graylog adalah platform *open source* untuk menerima, mengelola, menganalisis, dan memvisualisasikan *log* dari berbagai sumber dalam infrastruktur TI secara *real-time*[21]. Platform ini membantu tim keamanan mendeteksi ancaman, menyelidiki insiden, dan memantau aktivitas sistem dengan lebih efektif[22]. Graylog menawarkan fitur pencarian *real-time*, pembuatan *dashboard*, pemberian peringatan *(alerting),* serta kemampuan integrasi dengan berbagai sumber *log* untuk mendukung operasi keamanan dan kepatuhan.

11. Indicator of Compromise (IoC)

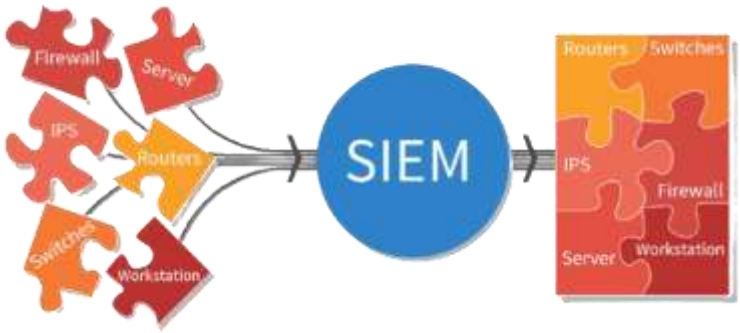
    Indicator of Compromise (IoC) merupakan bukti teknis yang mengindikasikan adanya potensi atau indikasi pelanggaran keamanan pada sistem, jaringan, maupun *endpoint*. IoC dimanfaatkan oleh tim SOC sebagai acuan dalam mendeteksi aktivitas berbahaya, melakukan analisis insiden, serta mengenali pola serangan yang telah atau sedang terjadi. IoC berperan

penting dalam proses deteksi dan penanganan insiden karena memungkinkan SOC untuk mengintegrasikan dan mengkorelasikan data dari berbagai sumber keamanan, seperti SIEM, EDR, IDS/NIDS, serta *Threat Intelligence Platform*. Dengan mengkorelasikan IoC terhadap *log* dan *event* yang dikumpulkan, analis SOC dapat mempercepat proses *triage*, menentukan tingkat keparahan insiden, serta mengambil tindakan mitigasi yang tepat. Selain itu, pemanfaatan IoC berkontribusi dalam menurunkan nilai *Mean Time to Detect (MTTD)* dan *Mean Time to Respond (MTTR)* dengan menyediakan indikator yang telah tervalidasi untuk mendukung deteksi ancaman yang lebih cepat dan presisi[7], [23], [24].

Gambar 4 Indicator of Compromise Pyramid

12. Security Information and Event Management (SIEM)

Security Information and Event Management (SIEM) adalah solusi keamanan siber yang berfungsi mengumpulkan, menganalisis, dan mengkorelasikan *log* serta *event* keamanan dari berbagai sumber dalam infrastruktur TI, seperti server, perangkat jaringan, aplikasi, dan sistem keamanan. SIEM membantu organisasi mendeteksi ancaman dan insiden keamanan secara *real-time*, menyediakan visibilitas terpusat, mendukung investigasi insiden, serta membantu pemenuhan kepatuhan dan pelaporan keamanan [5], [25], [26], [27].



Gambar 5 Security Information and Event Management

13. Endpoint Detection and Response (EDR)

Endpoint Detection and Response adalah solusi keamanan siber yang memantau, mendeteksi, menganalisis, dan merespons aktivitas berbahaya pada *endpoint* seperti komputer dan server secara *real-time*[28]. Pada konteks *Security Operations Center*, EDR berperan penting dalam mengumpulkan data aktivitas *endpoint* untuk mengidentifikasi perilaku mencurigakan, memungkinkan investigasi insiden secara mendalam, serta menyediakan kemampuan respons seperti isolasi perangkat, penghentian proses berbahaya, dan remediasi guna mencegah penyebaran ancaman di dalam jaringan[7], [28].

14. Intrusion Detection System (IDS)

Intrusion Detection System (IDS) adalah sistem keamanan yang berfungsi untuk memantau lalu lintas jaringan atau aktivitas sistem guna mendeteksi adanya upaya penyusupan, serangan, atau aktivitas mencurigakan secara *real-time*. IDS bekerja dengan menganalisis pola lalu lintas dan perilaku sistem, lalu memberikan peringatan *(alert)* kepada administrator ketika terdeteksi ancaman, sehingga memungkinkan tindakan lanjutan sebelum serangan menyebabkan dampak yang lebih besar[29], [30], [31], [32].

15. Intrusion Prevention System (IPS)

Intrusion Prevention System (IPS) adalah sistem keamanan yang berfungsi untuk memantau lalu lintas jaringan secara *real-time* serta mendeteksi dan mencegah upaya penyusupan atau serangan siber, berbeda dengan IDS yang hanya memberikan peringatan, IPS dapat secara otomatis mengambil tindakan seperti memblokir *trafik* berbahaya, memutus koneksi, atau menghentikan serangan, sehingga membantu melindungi jaringan dari ancaman sebelum menimbulkan dampak yang lebih serius [32], [33].

16. Network-based Intrusion Detection System (NIDS)

Network-based Intrusion Detection System (NIDS) adalah sistem keamanan yang berfungsi untuk memantau dan menganalisis lalu lintas jaringan guna mendeteksi aktivitas mencurigakan, serangan, atau upaya penyusupan. NIDS ditempatkan pada titik strategis dalam jaringan untuk menginspeksi paket data yang melintas dan akan memberikan peringatan kepada administrator ketika terdeteksi pola serangan, sehingga membantu organisasi mengidentifikasi ancaman terhadap keamanan jaringan secara dini[30], [34], [35], [36].

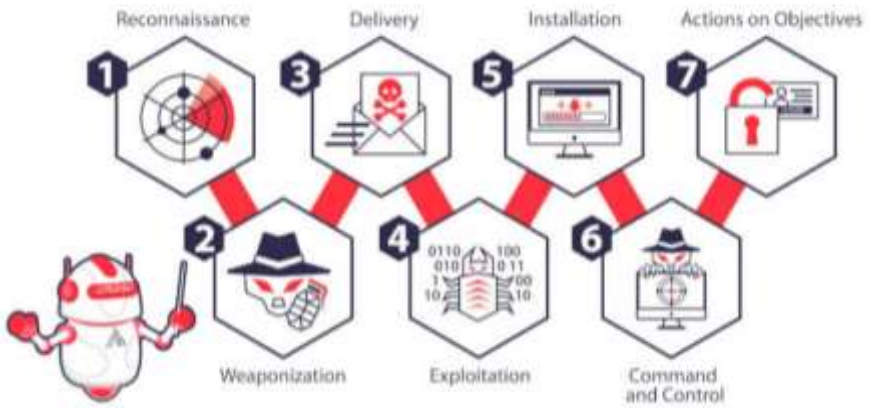
17. Extended Detection and Response (XDR)

Extended Detection and Response adalah pendekatan keamanan siber terpadu yang menggabungkan deteksi dan respons ancaman dari berbagai *domain* ke dalam satu platform terpusat. Tidak seperti EDR yang hanya

fokus pada *endpoint*, XDR mengumpulkan dan mengkorelasikan data dari berbagai sumber seperti *endpoint*, jaringan, *email*, *cloud*, dan aplikasi untuk memberikan visibilitas ancaman yang lebih luas. Pendekatan ini membantu tim SOC dalam mengidentifikasi serta menangani serangan yang kompleks yang melibatkan banyak vektor serangan sekaligus, sehingga mempercepat proses deteksi dan respons[37], [38], [39].

18. MITRE ATT&CK

MITRE ATT&CK *(Adversarial Tactics, Techniques, and Common Knowledge)* adalah kerangka kerja berbasis pengetahuan yang memetakan perilaku pelaku ancaman siber dalam bentuk taktik, teknik, dan prosedur (TTP) yang disusun berdasarkan observasi terhadap serangan dunia nyata. Kerangka ini menggambarkan tahapan lengkap siklus serangan, mulai dari akses awal hingga tujuan akhir penyerang, sehingga menyediakan taksonomi serangan yang terstruktur untuk memahami pola ancaman secara komprehensif[4], [40].



Gambar 6 Cyber Kill Chain

BAB III METODOLOGI

A. Waktu dan Tempat Penelitian

Waktu penelitian dilaksanakan pada bulan Januari sampai dengan Maret 2026. Tempat penelitian dilaksanakan di Laboratorium Jaringan Universitas Nahdlatul Ulama Al Ghazali Cilacap.

B. Alat dan Bahan (opsional)

Dalam penelitian ini diperlukan sejumlah alat dan bahan untuk merancang prototipe *Security Operations Center*. Daftar alat dan bahan yang digunakan dalam proses perancangan dan pengembangan prototipe adalah sebagai berikut:

Kebutuhan Perangkat Keras:

Tabel 2 Kebutuhan Perangkat Keras

No	Perangkat Keras	Keterangan
1	*Laptop (Prosesor Intel Core i5-8350U, RAM 16 GB, SSD 256 GB)*	Digunakan untuk mencari referensi dan membuat proyek penelitian S*ecurity Operations Center (SOC).*

Kebutuhan Virtual Private Server (VPS):

Tabel 3 Kebutuhan VPS

No	Spesifikasi	Keterangan
1	8 vCPU, 8 GB RAM, 200 GB SSD	Server utama untuk menjalankan SIEM Wazuh untuk memantau *endpoint* dan analisis *log* keamanan.
2	4 vCPU, 4 GB RAM, 100 GB SSD	1. Server Graylog untuk menerima *log* dari Wazuh untuk ditampilkan sebagai *dashboard*. 2. Server MISP untuk menyimpan dan membagikan IoC.
3	4 vCPU, 8 GB RAM, 50 GB SSD	Sebagai server *endpoint* dan web server dalam skenario pengujian, guna mensimulasikan aktivitas sistem yang akan dianalisis oleh sistem keamanan.

Kebutuhan Perangkat Lunak:

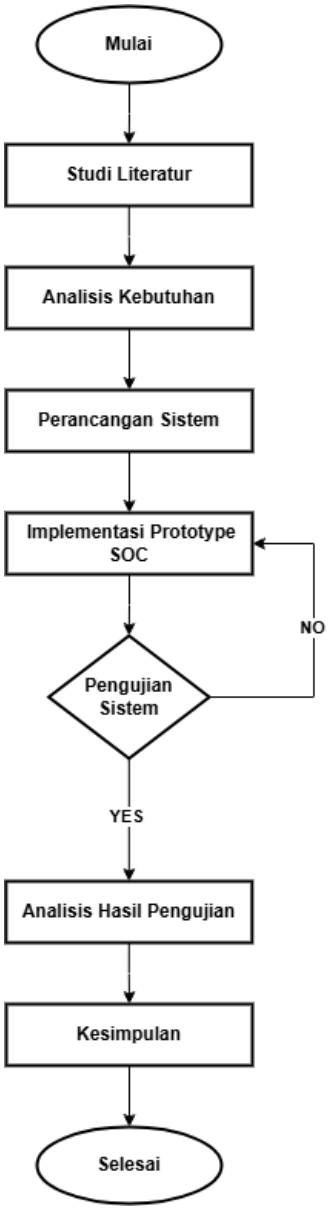
Tabel 4 Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Ubuntu Server versi 22.04.3 LTS	Sistem Operasi yang digunakan server.
2	Google Chrome versi 144.0.7559.133 (Official Build) (64-bit)	Browser untuk menjalankan infrastruktur keamanan.
3	Microsoft Edge versi 144.0.3719.115 (Official build) (64-bit)	Browser untuk menjalankan website sistem *endpoint*.
4	Wazuh	Sebagai *Security Information and Event Management (SIEM)* dengan fungsi memantau, menganalisis *log*, mendeteksi anomali, dan merespons ancaman keamanan secara *real-time.*
5	Suricata	Digunakan sebagai NIDS/IPS untuk menganalisis lalu lintas jaringan dan mendeteksi pola serangan berdasarkan *signature* dan *behavior*.
6	Graylog	Graylog digunakan untuk pengelolaan dan visualisasi *log* dari berbagai sumber (Wazuh, Suricata, sistem Operasi), untuk membantu dalam investigasi insiden dan audit keamanan.
7	MISP	Digunakan untuk berbagi dan mengelola *threat intelligence* dan *Indicator of Compromise (IoC),* sehingga sistem keamanan dapat mengenali ancaman lebih cepat.

No	Perangkat Lunak	Keterangan
8	Docker	Docker berfungsi sebagai platform untuk mendukung *deployment* web server dan aplikasi berbasis *container*.
9	VirusTotal	Digunakan sebagai bagian dari sistem XDR untuk mendukung proses analisis, deteksi, dan respons terhadap ancaman siber seperti *hash file* yang teridentifikasi sebagai *malware*, alamat IP berbahaya, serta *domain* dan URL mencurigakan.
10	Sysmon	Digunakan untuk memantau aktivitas penting di sistem Linux, seperti eksekusi perintah, koneksi jaringan, dan manipulasi file, untuk mendukung analisis forensik dan deteksi ancaman dalam arsitektur XDR.
11	Fluent Bit	Digunakan untuk mengumpulkan, memproses, dan meneruskan data berupa *log* dan *metrik* ke dalam Graylog, serta berfungsi sebagai lapisan pengamanan data melalui penerapan enkripsi.
12	Postfix	Berperan sebagai *mail server* internal yang mendistribusikan notifikasi *alert* dari sistem kepada administrator maupun tim SOC.
13	Telegram Bot	Digunakan sebagai media notifikasi *alert* utama.
14	Auditd	Digunakan untuk mencatat aktivitas sistem untuk dikirim ke SIEM.

C. Prosedur Penelitian

Gambar 7 menunjukkan prosedur penelitian yang dilakukan oleh peneliti. Prosedur tersebut di antaranya adalah pendahuluan dan studi literatur, analisis kebutuhan, perancangan sistem, implementasi prototipe SOC, pengujian sistem, analisis hasil pengujian, dan penarikan kesimpulan.



Gambar 7 Diagram Alir Penelitian

Diagram penelitian menggambarkan alur dan tahapan pelaksanaan penelitian yang dijelaskan sebagai berikut:

1. Studi Literatur

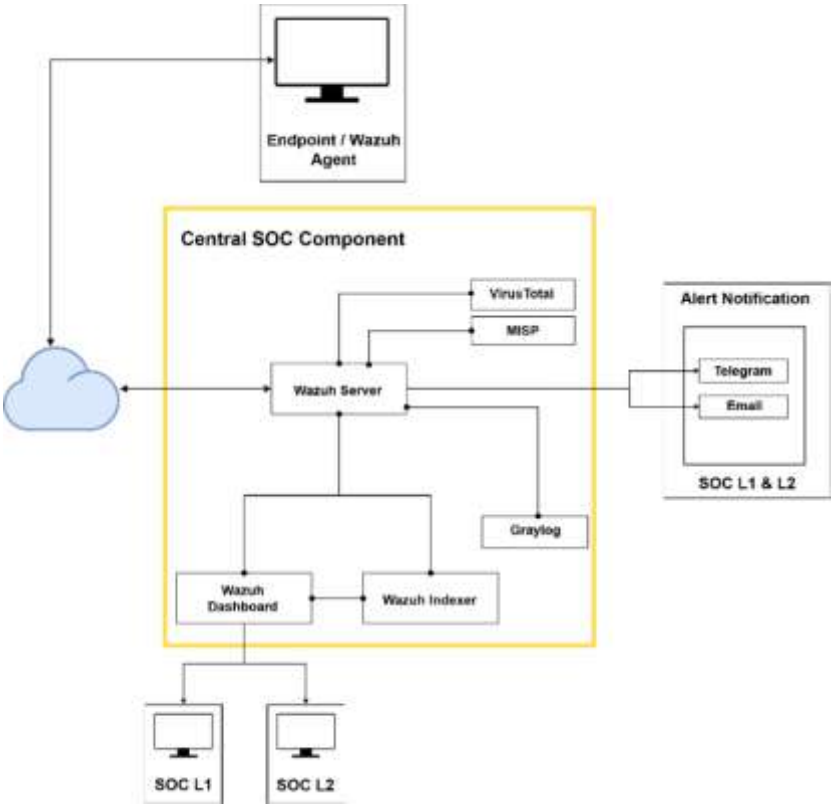
   Studi literatur merupakan tahapan awal dalam prosedur penelitian yang bertujuan untuk memperoleh informasi yang valid sebagai landasan pelaksanaan penelitian. Informasi tersebut dikumpulkan melalui penelusuran berbagai sumber, seperti buku, jurnal ilmiah*, e-book*, serta situs internet. Pada tahap ini, peneliti mengidentifikasi kebutuhan sistem yang akan dikembangkan dengan menelaah berbagai referensi terkait *Security Operations Center* berbasis *open source*. Data yang diperoleh dari studi literatur digunakan untuk memahami *tools*, metode, dan pendekatan yang telah diterapkan pada penelitian sebelumnya, kemudian dianalisis untuk menentukan arah penelitian serta kebutuhan teknis yang diperlukan.

2. Analisis Kebutuhan

   Setelah tahap studi literatur selesai dilakukan, langkah berikutnya adalah analisis kebutuhan untuk merancang Prototipe *Security Operations Center*. Tahap analisis ini meliputi identifikasi kebutuhan perangkat lunak, perangkat keras, arsitektur sistem, serta komponen lain yang mendukung perancangan sistem SOC.

3. Perancangan Sistem

   Tahapan selanjutnya setelah melakukan identifikasi kebutuhan sumber daya pada Prototipe *Security Operations Center* adalah menyusun rancangan sistem pemantauan berdasarkan hasil analisis yang telah dilakukan. Perancangan sistem ini bertujuan untuk memberikan gambaran komprehensif mengenai sistem yang akan dikembangkan serta memudahkan proses implementasi, khususnya dalam menentukan komponen-komponen utama yang diperlukan.



Gambar 8 Diagram Arsitektur *Security Operations Center*

Pada arsitektur SOC ini, terdapat beberapa komponen utama yang saling terintegrasi untuk menunjang proses pemantauan, deteksi, serta respons terhadap insiden keamanan siber, yang meliputi:

a) *Endpoint*

Endpoint adalah perangkat akhir *(end device)* yang terhubung ke jaringan dan menjadi titik interaksi langsung antara pengguna dan sistem, seperti server, komputer, laptop, atau mesin virtual. Dalam konteks keamanan siber, *endpoint* merupakan objek utama yang dipantau untuk mengumpulkan *log*, memantau aktivitas sistem, mendeteksi perilaku mencurigakan, serta mengirimkan data keamanan ke server pusat untuk dianalisis dan ditindaklanjuti sebagai bagian dari proses respons insiden.

b) Wazuh

Wazuh digunakan untuk mengumpulkan, menganalisis, dan mengorelasikan *log* serta aktivitas keamanan dari berbagai *endpoint* secara terpusat. Platform ini mampu mendeteksi ancaman seperti *system intrusion*, *malware*, *brute force*, dan perubahan tidak sah pada sistem, serta mendukung proses respons insiden melalui pembuatan layanan *threat intelligence* maupun sistem notifikasi.

c) VirusTotal

VirusTotal digunakan sebagai sumber pengayaan intelijen ancaman *(threat intelligence enrichment).* Wazuh mengirimkan *hash file*, alamat IP, atau URL yang mencurigakan ke VirusTotal untuk dilakukan pemeriksaan menggunakan berbagai *antivirus* dan basis data ancaman, sehingga membantu meningkatkan akurasi dalam mengidentifikasi *malware* atau aktivitas berbahaya.

d) Graylog

Graylog berfungsi sebagai pusat *log management* yang mengumpulkan, menyimpan, dan memvisualisasikan *log* dari berbagai sumber, termasuk Wazuh. Melalui fitur pencarian dan *dashboard*, Graylog membantu analis SOC dalam melakukan pemantauan, analisis, serta investigasi insiden keamanan secara lebih efektif.

e) Telegram

Telegram dimanfaatkan sebagai media notifikasi *real-time* untuk pengiriman *alert* keamanan. Ketika terjadi insiden atau aktivitas mencurigakan, sistem secara otomatis mengirimkan pesan peringatan ke kanal atau bot Telegram, sehingga memungkinkan tim keamanan untuk segera mengetahui dan merespons insiden.

f) *Email*

Email yang dikelola melalui Postfix digunakan sebagai sarana notifikasi formal dan dokumentasi insiden keamanan. *Alert* yang dihasilkan oleh sistem SOC dikirimkan melalui *email* kepada administrator atau tim terkait sebagai laporan insiden, pelengkap notifikasi Telegram, serta arsip komunikasi keamanan dengan tingkat keparahan tinggi *(high severity).*

4. Implementasi Prototipe SOC

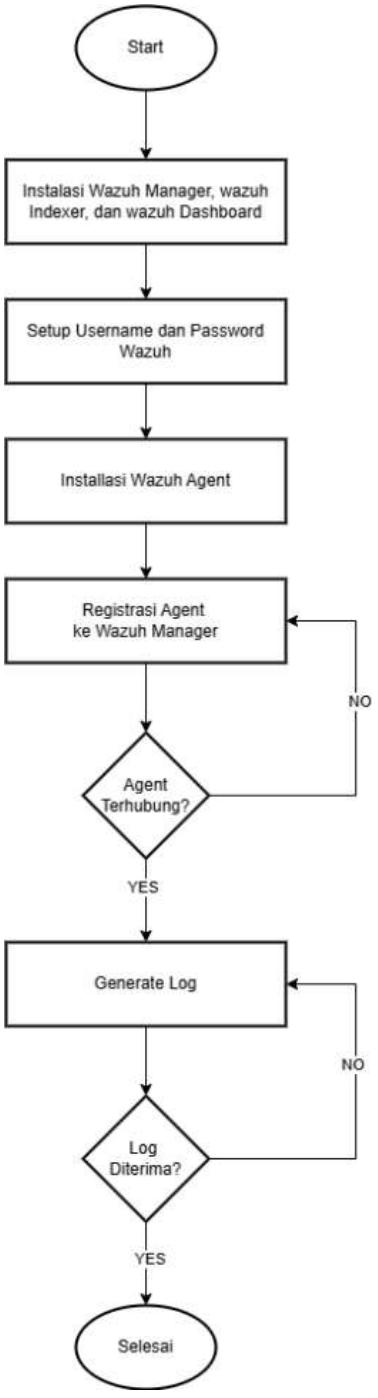
Tahapan selanjutnya setelah proses perancangan sistem adalah melakukan implementasi prototipe *Security Operations Center (SOC)* berdasarkan rancangan arsitektur dan spesifikasi teknis yang telah ditetapkan sebelumnya. Tahap ini bertujuan untuk merealisasikan desain sistem ke dalam bentuk prototipe yang dapat diuji fungsionalitasnya, sehingga seluruh komponen yang direncanakan dapat diintegrasikan dan beroperasi secara terpadu sesuai dengan kebutuhan sistem keamanan.

Berikut beberapa tahapan implementasi prototipe yang dilakukan dalam penelitian ini sebagai bagian dari perancangan dan pengujian sistem:

a) Instalasi Wazuh

Tahap awal dalam implementasi prototipe SOC diawali dengan pemasangan Wazuh Manager, Wazuh Indexer, dan Wazuh Dashboard, kemudian dilanjutkan dengan pengaturan akun pengguna. Setelah itu

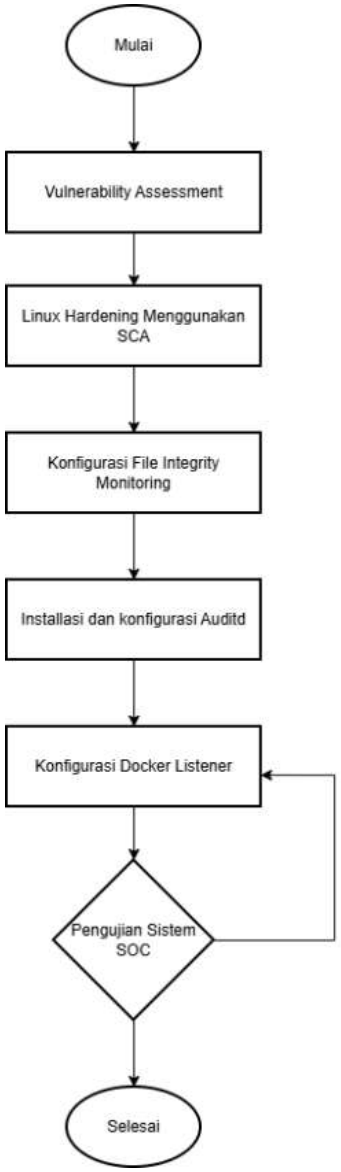
dilakukan instalasi serta registrasi Wazuh Agent ke Wazuh Manager. Apabila agent telah berhasil terhubung, proses dilanjutkan dengan pengujian melalui pembuatan *log*. Implementasi sistem dinyatakan berhasil apabila *log* dapat diterima oleh sistem.



Gambar 9 Diagram Alir Instalasi Wazuh

b) Konfigurasi Fungsi Dasar Wazuh

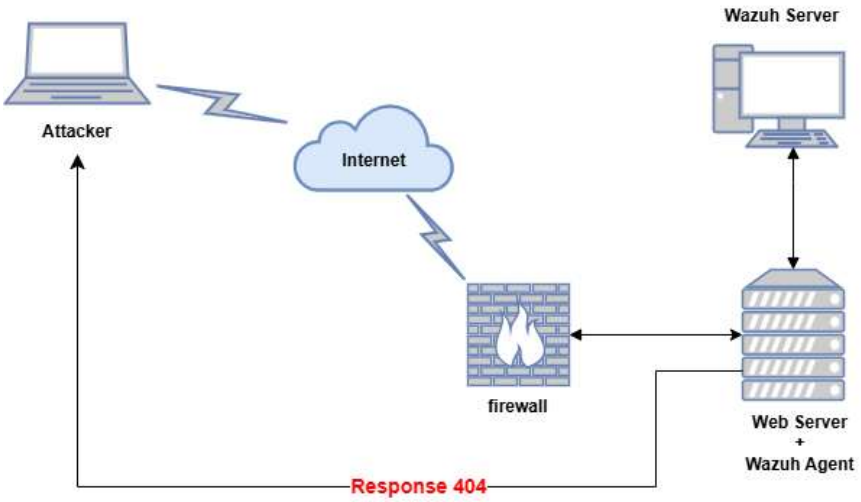
Tahap kedua bertujuan untuk mengimplementasikan serta meningkatkan mekanisme keamanan pada sistem SOC. Proses diawali dengan pelaksanaan *vulnerability assessment* guna mengidentifikasi potensi celah keamanan. Selanjutnya dilakukan penguatan sistem Linux melalui penerapan *Security Configuration Assessment (SCA),* kemudian dilanjutkan dengan konfigurasi *File Integrity Monitoring* serta pemasangan dan pengaturan Auditd. Setelah itu, dilakukan konfigurasi Docker Listener untuk memantau aktivitas pada lingkungan *container*. Tahapan ini diakhiri dengan pengujian sistem SOC guna memastikan seluruh fungsi keamanan berjalan secara optimal.



Gambar 10 Diagram Alir Konfigurasi Dasar Wazuh

c) Konfigurasi *Intrusion Prevention System (IPS)*

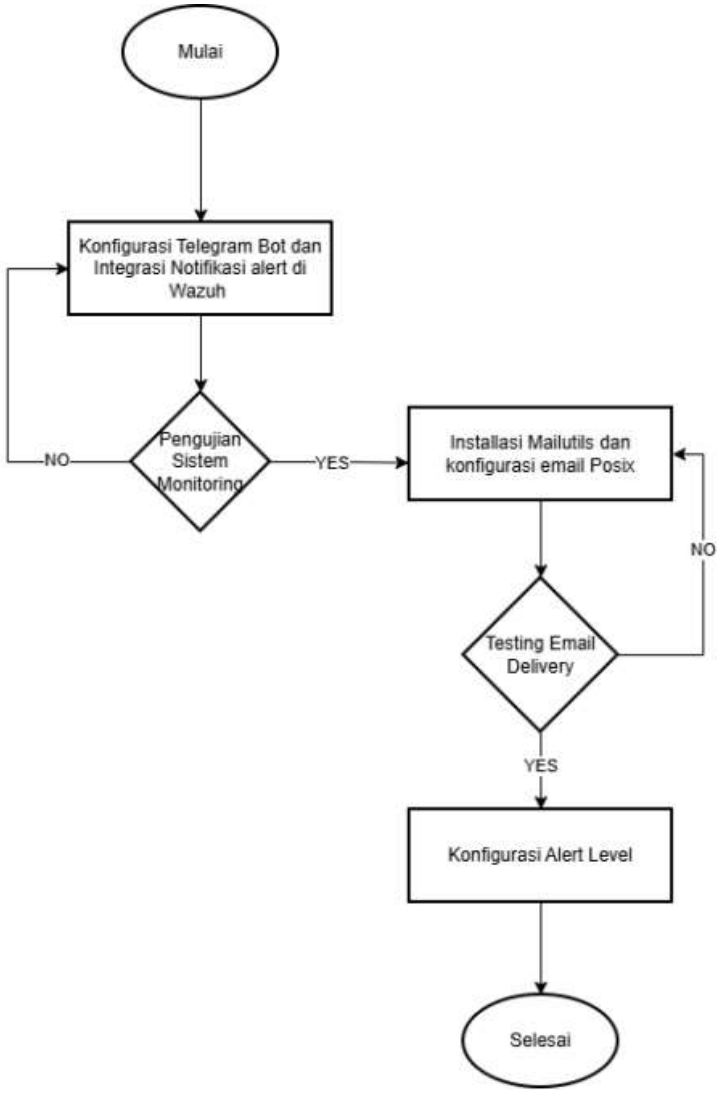
Tahap selanjutnya adalah konfigurasi *Intrusion Prevention System (IPS)* dengan menerapkan aturan *firewall drop* sebagai mekanisme *active response.* Pada tahap ini, Wazuh dikonfigurasikan untuk secara otomatis memblokir alamat IP atau aktivitas mencurigakan yang terdeteksi, sehingga serangan dapat dicegah secara langsung dan risiko terhadap sistem dapat diminimalkan.



Gambar 11 Intrusion Prevention System (IPS)

d) *Alert Notification*

Tahap *alert notification* berfokus pada konfigurasi mekanisme pemberitahuan ketika sistem mendeteksi aktivitas keamanan. Proses dimulai dengan konfigurasi Telegram Bot dan integrasi notifikasi *alert* pada Wazuh. Selanjutnya dilakukan pengujian *system monitoring* untuk memastikan *alert* dapat dihasilkan dengan benar. Selain itu, dilakukan instalasi Mailutils dan konfigurasi *email* berbasis Postfix sebagai media notifikasi tambahan. Pengujian pengiriman *email* dilakukan untuk memastikan notifikasi dapat diterima dengan baik. Setelah pengujian berhasil, dilakukan pengaturan tingkat keparahan *(alert level)* sebelum sistem dinyatakan selesai dan siap digunakan.



Gambar 12 Diagram Alir Implementasi Notifikasi

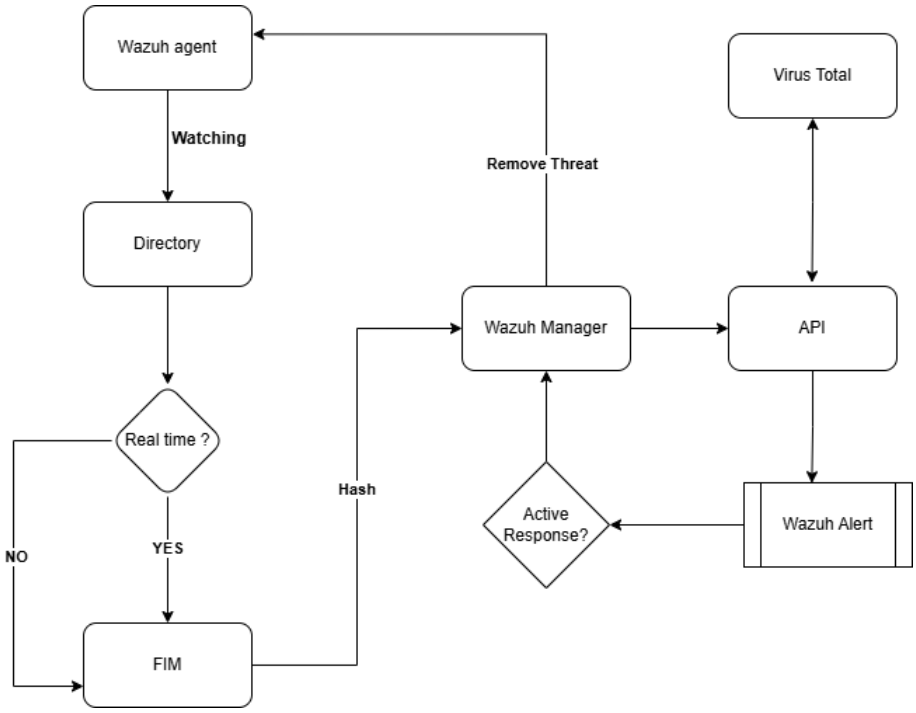
e) NIDS Integration

Tahap ini dilakukan dengan mengintegrasikan *Network Intrusion Detection System (NIDS)* menggunakan Suricata ke dalam sistem SOC. Suricata berfungsi untuk memantau dan menganalisis lalu lintas jaringan guna mendeteksi aktivitas mencurigakan atau serangan jaringan. Hasil deteksi kemudian dikirimkan ke Wazuh untuk dianalisis dan ditampilkan sebagai *alert* pada *dashboard* SOC.

f) Antimalware Integration

Tahap ini dilakukan dengan mengintegrasikan fitur anti *malware* pada Wazuh dengan layanan VirusTotal. Integrasi ini bertujuan untuk melakukan analisis *hash file* secara otomatis guna mendeteksi indikasi *malware*. Hasil pemindaian dari VirusTotal kemudian digunakan

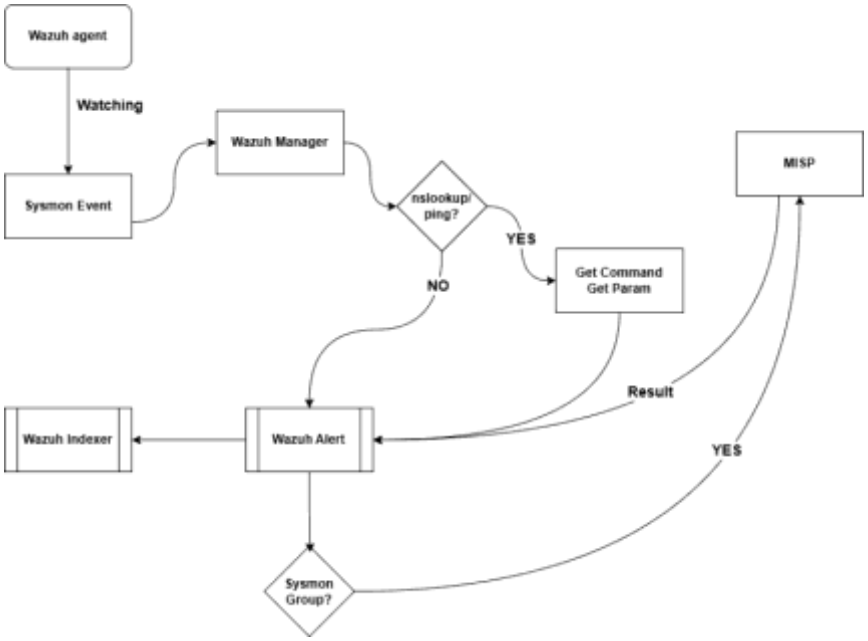
sebagai dasar pembuatan *alert* sehingga membantu meningkatkan akurasi deteksi ancaman *malware* pada sistem.



Gambar 13 Diagram Alir Integrasi VirusTotal

g) Threat Intelligence Integration

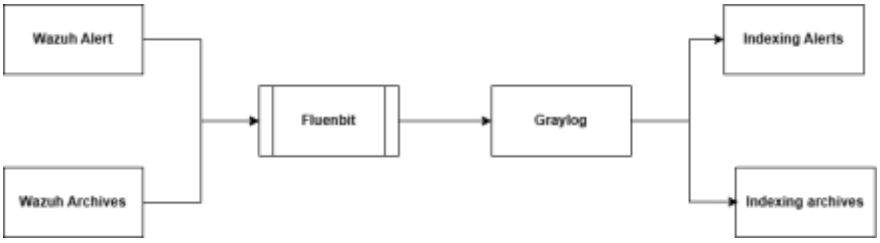
Tahap ini dilakukan dengan mengintegrasikan sistem SOC dengan MISP sebagai sumber *threat intelligence.* Integrasi ini memungkinkan sistem untuk memanfaatkan IoC seperti alamat IP*, domain,* dan *hash file*. Informasi ancaman dari MISP digunakan oleh Wazuh untuk meningkatkan akurasi deteksi, mempercepat respons insiden, serta memperkuat mekanisme pencegahan terhadap ancaman keamanan siber.



Gambar 14 Diagram Alir Integrasi MISP

h) Centralized Log Management

Tahap ini bertujuan untuk menerapkan *log management* terpusat menggunakan Graylog. Seluruh *log* yang dihasilkan oleh sistem, perangkat, dan aplikasi dikirimkan ke Graylog untuk dikumpulkan, disimpan, dan dianalisis secara terpusat. Dengan penerapan ini, proses pemantauan, pencarian, dan analisis *log* menjadi lebih efisien, sehingga membantu dalam deteksi insiden keamanan serta mendukung aktivitas forensik.



Gambar 15 Diagram Alir Integrasi Graylog

5. Pengujian Sistem

Tahapan pengujian ini bertujuan untuk memastikan bahwa sistem SOC yang dikembangkan mampu berfungsi sesuai dengan rancangan serta memenuhi kebutuhan keamanan yang telah ditetapkan. Proses pengujian dilakukan dengan mensimulasikan beberapa skenario serangan keamanan, seperti aktivitas

pemindaian jaringan dan percobaan akses tidak sah, guna mengevaluasi kemampuan sistem dalam mengumpulkan *log*, mendeteksi anomali, serta menghasilkan peringatan keamanan secara tepat waktu.

Selain itu, pengujian sistem juga mencakup verifikasi integrasi antar komponen SOC, mulai dari sumber *log*, sistem deteksi, hingga platform SIEM. Hasil pengujian ini menjadi dasar dalam menentukan apakah sistem telah berjalan dengan baik atau masih memerlukan penyesuaian konfigurasi dan aturan deteksi sebelum memasuki tahap evaluasi lebih lanjut.

Tabel 5 Pengujian Fungsional Sistem SOC

NO	Jenis Pengujian	Tujuan Pengujian	Skenario Pengujian	Metrik Evaluasi
1	*File Integrity Monitoring (FIM)*	Mendeteksi perubahan tidak sah pada file dan direktori	Melakukan modifikasi, penghapusan, dan penambahan file pada direktori yang dimonitor, kemudian memverifikasi apakah sistem menghasilkan *log* dan *alert* atas perubahan tersebut.	MTTD, Akurasi Deteksi
2	*Directory Monitoring & User Root*	Memverifikasi aktivitas dan perubahan file oleh *user root* pada direktori sensitif	Melakukan akses dan perubahan file menggunakan *user root* pada direktori sensitif seperti */etc* dan */var*, lalu memastikan aktivitas tersebut tercatat pada sistem *monitoring.*	MTTD, *False Positive*
3	*Docker Activity Detection*	Mengevaluasi efektivitas sistem SOC dalam melakukan	Menjalankan perintah *start*, *stop, restart*, dan	MTTD, Kelengkapan *Log*

NO	Jenis Pengujian	Tujuan Pengujian	Skenario Pengujian	Metrik Evaluasi
		pencatatan dan pemantauan terhadap aktivitas operasional Docker, seperti *start, stop, dan exec.*	*exec* pada Docker, kemudian memverifikasi apakah aktivitas tersebut tercatat dan muncul pada *dashboard* SOC.	
4	*Docker Container Monitoring*	Mengevaluasi kemampuan sistem SOC dalam melakukan deteksi terhadap aktivitas berbahaya pada aplikasi yang dijalankan di lingkungan *container*.	Mensimulasikan permintaan berbahaya pada aplikasi berbasis *container*, seperti *SQL Injection* dan *Command Injection,* kemudian memverifikasi apakah aktivitas tersebut terdeteksi, dikorelasikan, dan menghasilkan peringatan pada *dashboard* SOC.	MTTD, Korelasi *Alert*
5	*Active Response*	Mengevaluasi efektivitas sistem dalam mengimplementasikan mekanisme respons otomatis terhadap insiden keamanan.	Mensimulasikan insiden keamanan tertentu dan memverifikasi apakah sistem secara otomatis menjalankan respons seperti pemblokiran atau isolasi.	MTTD, MTTR
6	*Block Malicious Actor*	Menguji pemblokiran otomatis terhadap IP atau aktor berbahaya	Mengirimkan *trafik* dari alamat IP yang dikategorikan berbahaya dan memastikan	MTTD, MTTR

NO	Jenis Pengujian	Tujuan Pengujian	Skenario Pengujian	Metrik Evaluasi
			sistem memblokir koneksi tersebut secara otomatis.	
7	*Monitoring Resource Usage*	Mengevaluasi kemampuan sistem dalam melakukan pemantauan penggunaan sumber daya (CPU, *memory*, dan *disk*) untuk mengidentifikasi indikasi terjadinya serangan.	Melakukan peningkatan beban sistem secara abnormal pada CPU, *memory*, dan *disk* serta memverifikasi apakah sistem mencatat serta memberikan peringatan atas anomali tersebut.	MTTD, *Anomaly Rate*
8	*Anomaly Detection*	Menguji kemampuan sistem mendeteksi pola aktivitas yang menyimpang dari kondisi normal	Mensimulasikan pola aktivitas tidak normal seperti lonjakan *login* atau *trafik* yang tidak biasa, kemudian memverifikasi apakah sistem mendeteksi dan menandainya sebagai anomali.	MTTD, *False Negative*
9	Pengujian NIDS	Mengevaluasi kemampuan sistem dalam melakukan deteksi terhadap *trafik* jaringan berbahaya sebagai bagian dari mekanisme keamanan jaringan.	Mensimulasikan serangan pada jaringan seperti *port scanning* dan serangan *Denial of Service (DoS)* untuk memverifikasi kemampuan sistem dalam mendeteksi serta menghasilkan	MTTD, Akurasi Deteksi

NO	Jenis Pengujian	Tujuan Pengujian	Skenario Pengujian	Metrik Evaluasi
			peringatan keamanan.	
10	Pengujian Anti *Malware*	Menguji kemampuan sistem mendeteksi, mencatat, dan merespons aktivitas *malware*	Menjalankan file uji *malware* dan memverifikasi apakah sistem mendeteksi, mencatat, dan merespons aktivitas tersebut.	MTTD, MTTR
11	Pengujian *Threat Intelligence* (MISP)	Memverifikasi integrasi intelijen ancaman untuk mendeteksi indikator serangan yang dikenal	Menguji IoC (Alamat IP, *hash file*, *domain*) yang berasal dari MISP dan memastikan sistem mampu mengkorelasi serta menghasilkan peringatan keamanan.	MTTD, Korelasi IoC
12	Pengujian skenario serangan OWASP Top 10 dan MITRE ATT&CK	Mengevaluasi kemampuan sistem SOC dalam mendukung deteksi, korelasi, serta penyajian peringatan terhadap berbagai teknik dan vektor serangan yang umum digunakan.	Mensimulasikan beberapa skenario serangan berdasarkan OWASP Top 10 dan teknik MITRE ATT&CK, lalu mengevaluasi kemampuan sistem dalam mendeteksi, mengkorelasi *log*, dan menghasilkan *alert*.	MTTD, MTTR

D. Analisis Data

Tahapan Analisis dilakukan untuk menilai efektivitas dan kinerja sistem SOC berdasarkan data kuantitatif yang diperoleh selama proses pengujian. Pada tahap ini dilakukan evaluasi terhadap akurasi deteksi serangan yang dinyatakan dalam persentase, serta jumlah dan persentase *false positive* dan *false negative* yang dihasilkan oleh sistem. Analisis ini bertujuan untuk menilai kemampuan sistem dalam mengidentifikasi serta menangani insiden keamanan secara optimal.

E. Jadwal Penelitian

Tabel 6 Jadwal penelitian

No	Nama Kegiatan	Tahun 2026 Bulan ke-					
		1	2	3	4	5	6
1	Tahap Persiapan						
	a. Studi Literatur	■					
	b. Rumusan masalah	■					
	c. Penetapan Metode	■					
	d. Seminar Proposal		■				
2	Tahap Penelitian						
	a. Implementasi *Prototype Security Operations Center*		■				
	b. Pengujian dan pengambilan data		■				
	c. Analisis Data		■				
	d. Evaluasi		■				
	e. Penulisan Skripsi		■				
3	Tahap Akhir						
	a. Sidang Skripsi			■			
	b. Publikasi ilmiah			■			

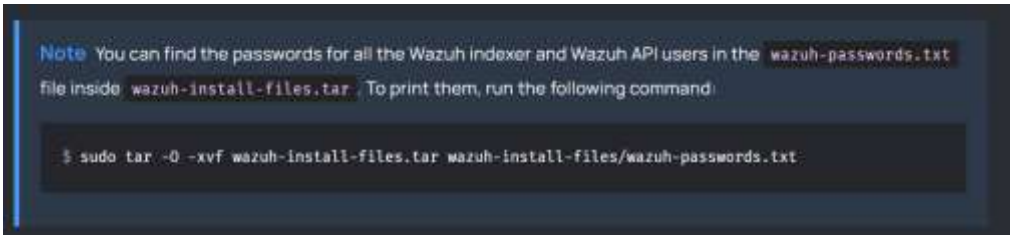
BAB IV HASIL DAN PEMBAHASAN

A. Hasil Implementasi Prototipe SOC

Dalam implementasi prototipe *Security Operations Center (SOC)*, peneliti mengintegrasikan beberapa perangkat lunak *open source* untuk membangun sistem deteksi dan respons ancaman siber yang terpusat. Komponen utama yang digunakan terdiri dari Wazuh sebagai *Security Information and Event Management (SIEM),* Suricata sebagai *Network Intrusion Detection System (NIDS),* VirusTotal sebagai layanan analisis *malware*, MISP (Malware Information Sharing Platform) sebagai sumber *threat intelligence*, serta Graylog sebagai *centralized log management*. Rangkaian tahapan implementasi yang dilakukan dijelaskan secara rinci sebagai berikut:

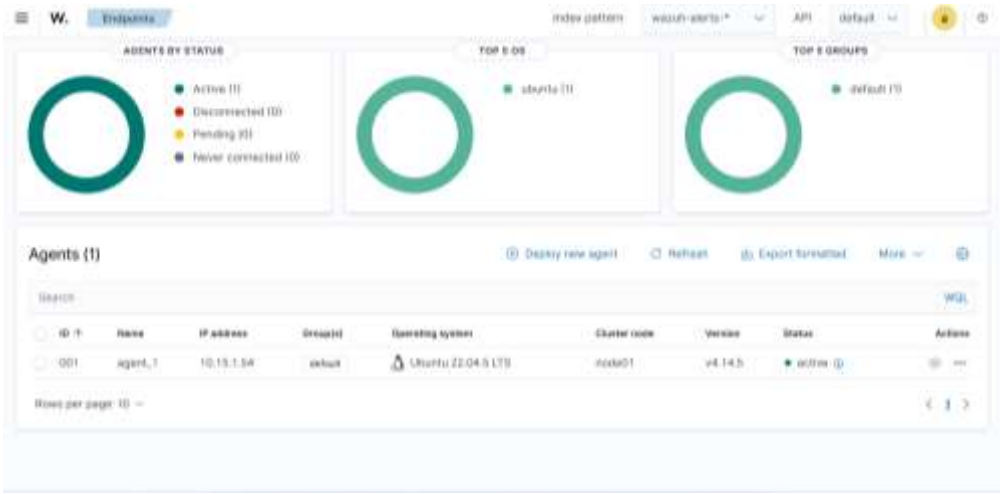
1. Hasil Implementasi Wazuh

   Wazuh digunakan sebagai komponen utama dalam prototipe *Security Operations Center (SOC)* yang berfungsi *sebagai Security Information and Event Management (SIEM).* Implementasi diawali dengan pemasangan Wazuh Manager, Wazuh Indexer, dan Wazuh Dashboard pada server yang telah disiapkan. Wazuh Manager berfungsi untuk menerima dan menganalisis *log* dari *endpoint*, Wazuh Indexer digunakan untuk menyimpan data hasil pengumpulan *log*, sedangkan Wazuh Dashboard digunakan sebagai antarmuka untuk melakukan pemantauan dan analisis keamanan secara terpusat.



Gambar 16 Instalasi wazuh

Setelah proses instalasi selesai, dilakukan konfigurasi akun pengguna dan pengaturan layanan agar seluruh komponen dapat saling terhubung dengan baik. Selanjutnya dilakukan instalasi Wazuh Agent pada *endpoint* yang akan dipantau. *Endpoint* tersebut kemudian diregistrasikan ke Wazuh Manager sehingga aktivitas yang terjadi pada sistem dapat dikumpulkan dan dianalisis secara terpusat. Hasil implementasi menunjukkan bahwa Wazuh Agent berhasil terhubung dengan Wazuh Manager dan berada dalam status *active* pada *dashboard monitoring*.



Gambar 17 Dashboard Wazuh

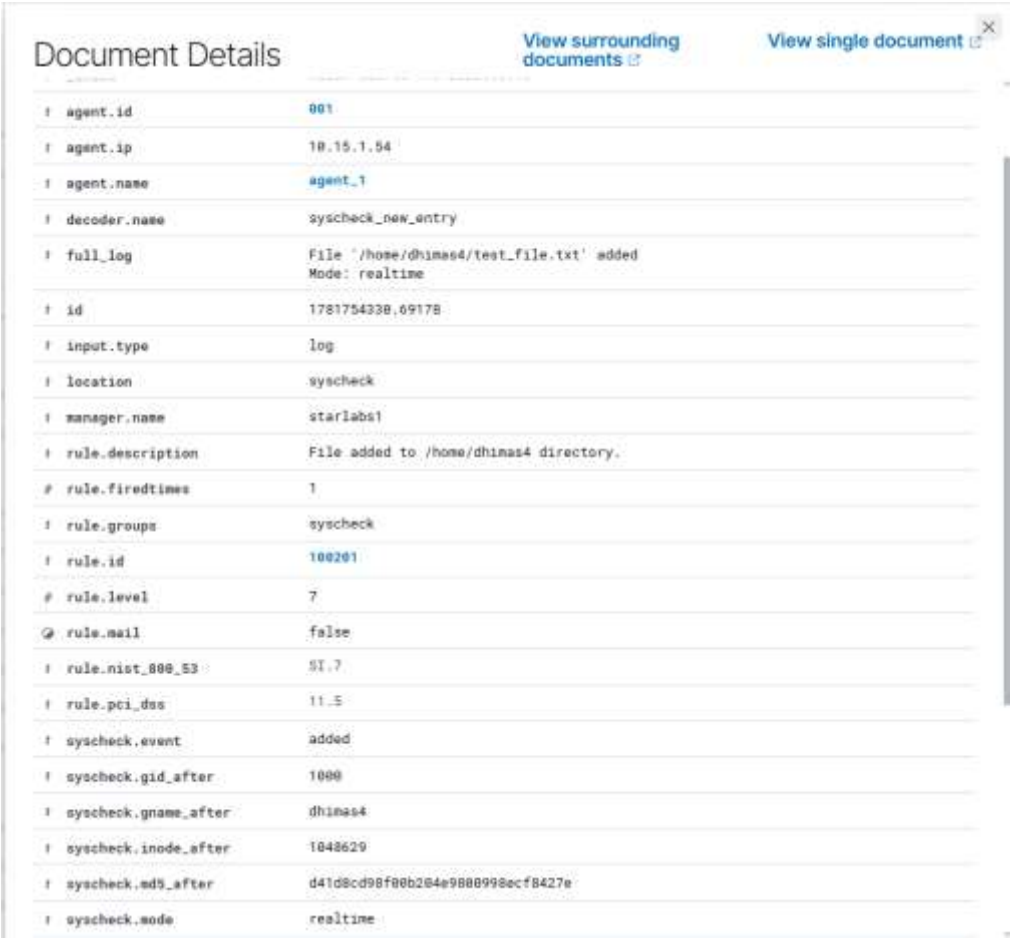
2. Hasil Implementasi *File Integrity Monitoring*

File Integrity Monitoring (FIM) dilakukan menggunakan fitur *Syscheck* pada Wazuh untuk memantau perubahan file dan direktori yang dianggap penting dalam sistem. Fitur ini bertujuan untuk mendeteksi aktivitas yang berpotensi mengindikasikan adanya akses tidak sah atau modifikasi terhadap file sistem yang dapat memengaruhi keamanan dan integritas server.

```
<!-- Files/directories Monitoring -->
<directories check_all="yes" report_changes="yes" realtime="yes">/home/dhimas4</directories>
<directories check_all="yes" report_changes="yes" realtime="yes">/etc/passwd</directories>
```

Gambar 18 Implementasi *File Integrity Monitoring*

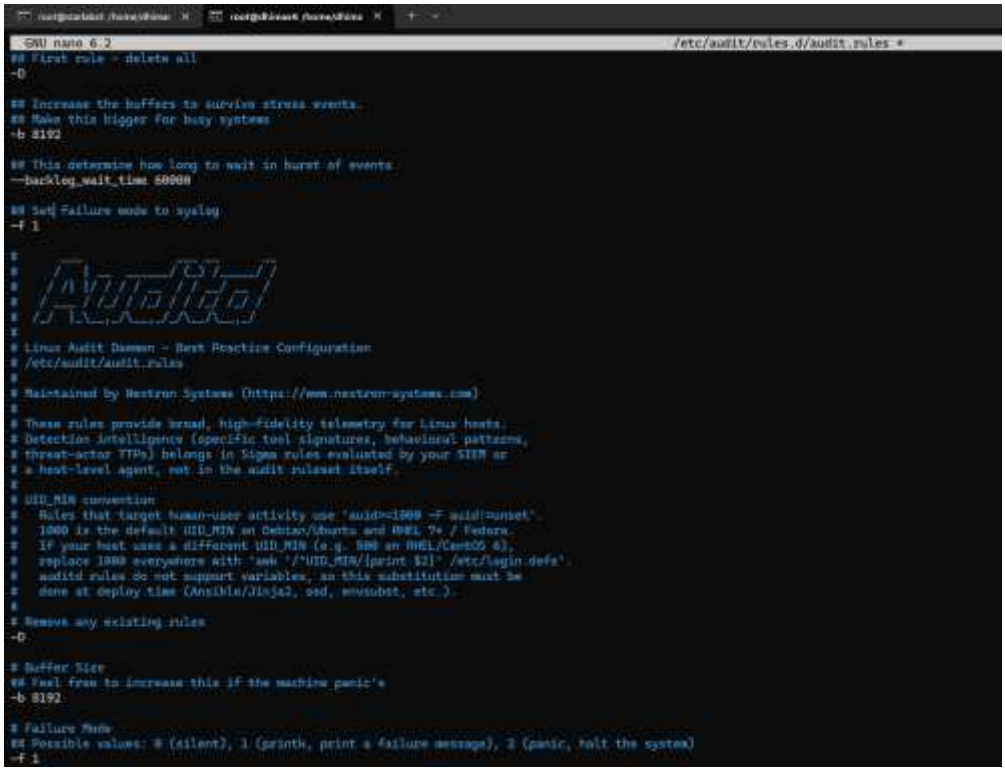
Hasil implementasi menunjukkan bahwa mekanisme FIM berhasil berjalan dengan baik. Setiap perubahan yang dilakukan pada file atau direktori yang dipantau dapat terdeteksi secara otomatis dan menghasilkan *log* keamanan pada dashboard Wazuh. Informasi yang ditampilkan meliputi nama file, lokasi file, jenis perubahan yang terjadi, waktu kejadian, serta agen yang mendeteksi aktivitas tersebut.



Gambar 19 Hasil File Integrity Monitoring (FIM)

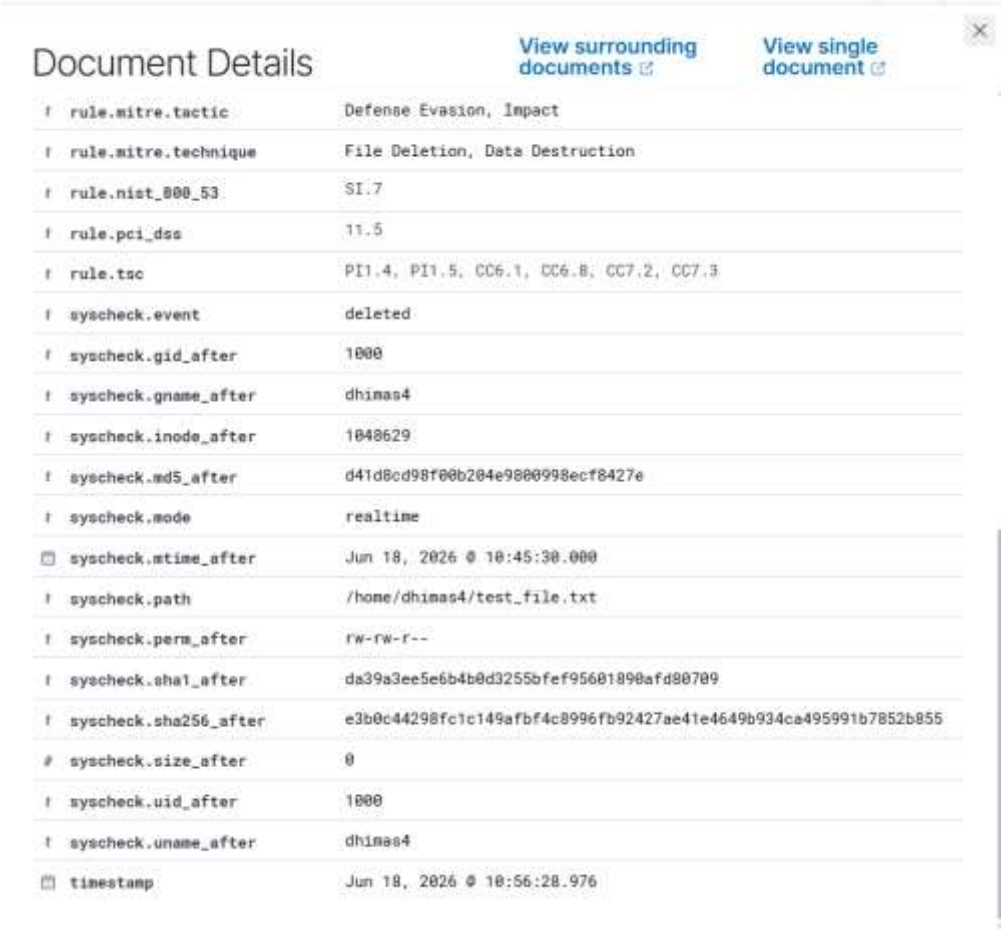
3. Implementasi Auditd

Auditd dipasang dan dikonfigurasi pada *endpoint* yang terhubung dengan Wazuh Agent. Selanjutnya, dibuat beberapa aturan *(audit rules)* untuk memantau aktivitas pada direktori dan file penting, aktivitas pengguna dengan hak akses tinggi *(privileged user)*, serta eksekusi perintah tertentu yang berpotensi memengaruhi keamanan sistem. *Log* yang dihasilkan oleh Auditd kemudian dikirimkan ke Wazuh Manager untuk dianalisis dan dikorelasikan dengan sumber *log* lainnya.



Gambar 20 Best practice rules

Hasil implementasi menunjukkan bahwa Auditd berhasil mencatat aktivitas sistem sesuai dengan aturan yang telah ditetapkan. Setiap aktivitas yang dimonitor menghasilkan *log* yang berisi informasi seperti identitas pengguna, perintah yang dijalankan, waktu kejadian, serta objek yang diakses. *Log* tersebut selanjutnya ditampilkan pada *dashboard* Wazuh sehingga memudahkan administrator dalam melakukan pemantauan dan investigasi terhadap aktivitas yang terjadi pada sistem.



Gambar 21 Hasil Implementasi Auditd

4. Implementasi Docker *Monitoring*

Implementasi Docker *Monitoring* dilakukan untuk memantau aktivitas yang terjadi pada lingkungan *container* yang digunakan dalam prototipe SOC. *Monitoring* ini bertujuan untuk memberikan visibilitas terhadap aktivitas operasional Docker sehingga administrator dapat mengetahui perubahan maupun aktivitas yang terjadi pada *container* secara terpusat melalui *dashboard* Wazuh.

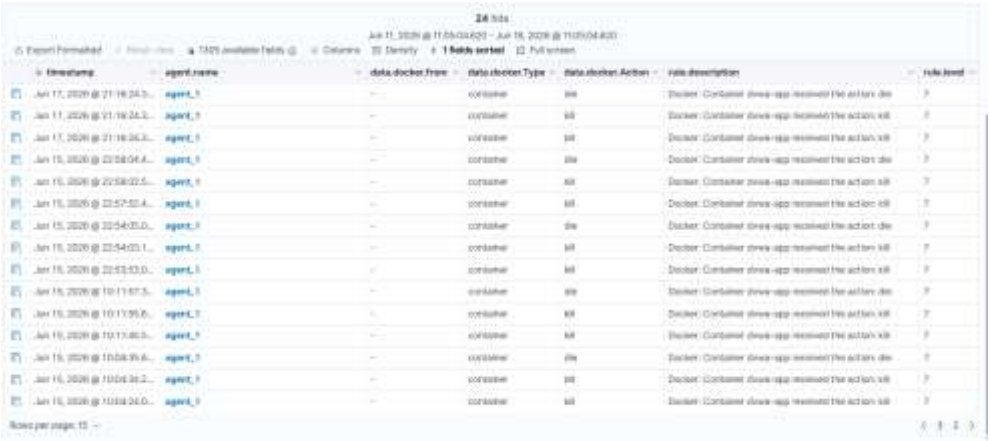
Pada tahap implementasi, Wazuh dikonfigurasi untuk mengumpulkan *log* yang dihasilkan oleh layanan Docker melalui fitur Docker Listener. Konfigurasi ini memungkinkan Wazuh menerima dan memproses berbagai aktivitas Docker seperti pembuatan *(create)*, menjalankan *(start),* menghentikan *(stop),* menghapus *(remove),* serta aktivitas eksekusi perintah *exec* pada container. Seluruh *log* yang diterima kemudian dianalisis dan ditampilkan pada *dashboard monitoring*.

```
<wodle name="cis-cat">
  <disabled>yes</disabled>
  <timeout>1800</timeout>
  <interval>1d</interval>
  <scan-on-start>yes</scan-on-start>

  <java_path>wodles/java</java_path>
  <ciscat_path>wodles/ciscat</ciscat_path>
</wodle>

<wodle name="docker-listener">
  <interval>10m</interval>
  <attempts>5</attempts>
  <run_on_start>yes</run_on_start>
  <disabled>no</disabled>
</wodle>
```

Gambar 22 Implementasi Docker Listener



Gambar 23 *Log* Aktivitas Docker

5. Implementasi *Active Response*

Implementasi *Active Response* dilakukan untuk memberikan kemampuan respons otomatis terhadap aktivitas yang teridentifikasi sebagai ancaman keamanan. Pada penelitian ini, mekanisme *Active Response* diterapkan menggunakan aturan *firewall-drop* yang disediakan oleh Wazuh. Mekanisme ini memungkinkan sistem untuk secara otomatis memblokir alamat IP yang terdeteksi melakukan aktivitas mencurigakan atau melanggar aturan keamanan yang telah ditetapkan.

Pada tahap implementasi, Wazuh dikonfigurasi untuk menjalankan aksi *firewall-drop* ketika *alert* dengan tingkat keparahan tertentu terdeteksi.

Konfigurasi dilakukan dengan menghubungkan aturan deteksi *(rule)* pada Wazuh dengan modul *Active Response* sehingga sistem dapat mengeksekusi perintah pemblokiran tanpa memerlukan intervensi administrator. Ketika sebuah ancaman terdeteksi, alamat IP sumber akan ditambahkan ke aturan *firewall* dan seluruh koneksi dari alamat IP tersebut akan ditolak secara otomatis.

```
<!-- Active response firewall-drop scanning -->
<active-response>
  <command>firewall-drop</command>
  <location>all</location>
  <rules_id>5760</rules_id>
  <timeout>3600</timeout>
</active-response>
```

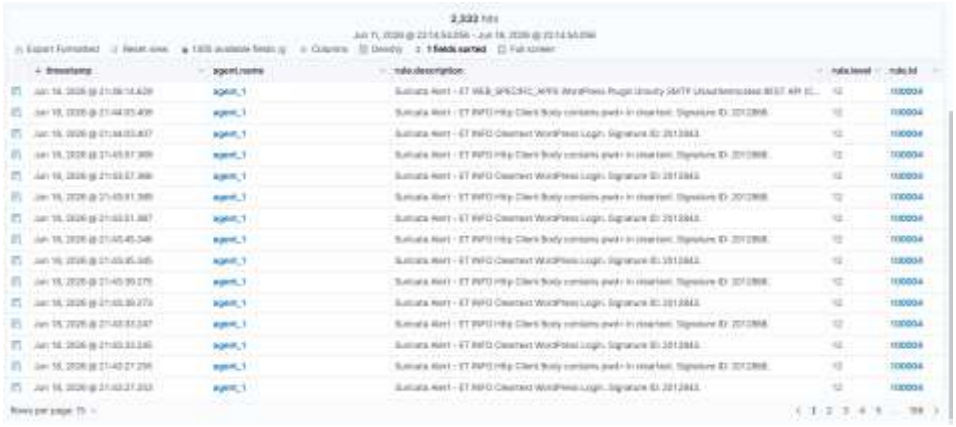
Gambar 24 Active Response firewall-drop

Penerapan *Active Response* memberikan kemampuan mitigasi ancaman secara cepat dengan mengurangi waktu respons terhadap insiden keamanan. Selain berfungsi sebagai mekanisme deteksi, sistem SOC yang dibangun juga mampu melakukan tindakan pencegahan awal terhadap aktivitas yang berpotensi membahayakan sistem.

Gambar 25 Hasil Implementasi *Active Response*

6. Implementasi Suricata

Pada prototipe *Security Operations Center (SOC).* Suricata berfungsi untuk memantau lalu lintas jaringan secara *real-time* dan mendeteksi berbagai aktivitas yang berpotensi mengancam keamanan sistem. Pada tahap implementasi, Suricata dipasang dan dikonfigurasi pada server yang bertugas memantau lalu lintas jaringan. Suricata diintegrasikan dengan Wazuh sehingga *log* dan *alert* yang dihasilkan dapat dikirim dan dianalisis secara terpusat. Integrasi ini memungkinkan informasi ancaman yang terdeteksi oleh Suricata ditampilkan pada Wazuh Dashboard untuk memudahkan proses *monitoring* dan investigasi keamanan.



Gambar 26 *Log* aktifitas jaringan

Hasil implementasi menunjukkan bahwa Suricata berhasil melakukan inspeksi terhadap lalu lintas jaringan dan menghasilkan *log* keamanan berdasarkan aktivitas yang terdeteksi. *Log* yang dihasilkan meliputi informasi seperti alamat IP sumber dan tujuan, jenis ancaman yang terdeteksi, protokol yang digunakan, serta tingkat keparahan *(severity)* dari aktivitas tersebut. Informasi tersebut kemudian berhasil dikirim ke Wazuh dan ditampilkan dalam bentuk *alert* pada *dashboard* SOC.

7. Hasil Implementasi VirusTotal

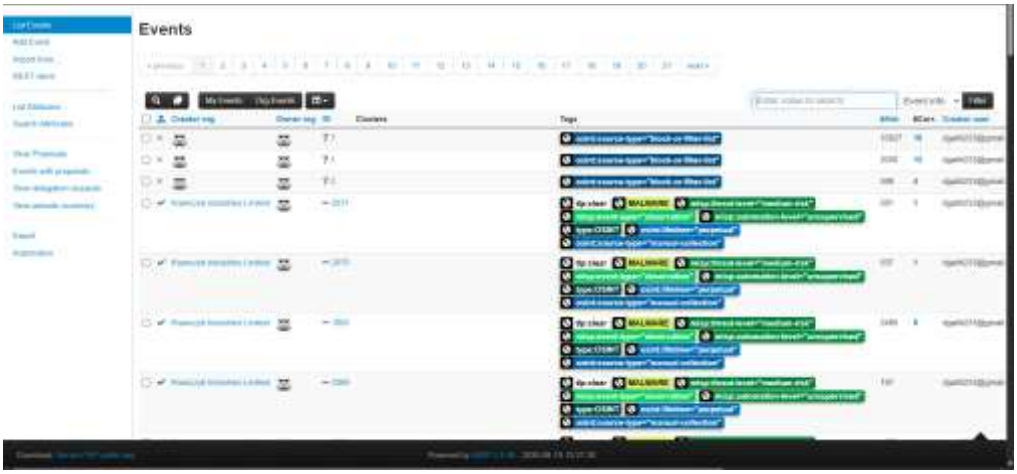
Implementasi VirusTotal dilakukan untuk menambahkan kemampuan deteksi *malware* pada prototipe *Security Operations Center (SOC)*. Integrasi ini memungkinkan sistem melakukan analisis terhadap file yang terdeteksi pada *endpoint* dengan memanfaatkan basis data dan mesin deteksi yang tersedia pada layanan VirusTotal. Pada tahap implementasi, Wazuh dikonfigurasi untuk terhubung dengan layanan VirusTotal melalui *Application Programming Interface (API).* Integrasi dilakukan dengan memanfaatkan fitur *File Integrity Monitoring (FIM)* sehingga setiap file baru atau file yang mengalami perubahan dapat diperiksa nilai *hash*-nya. Selanjutnya, nilai *hash* tersebut dikirim ke VirusTotal untuk dilakukan pencocokan terhadap basis data *malware* yang tersedia.

Gambar 27 *Alert* VirusTotal

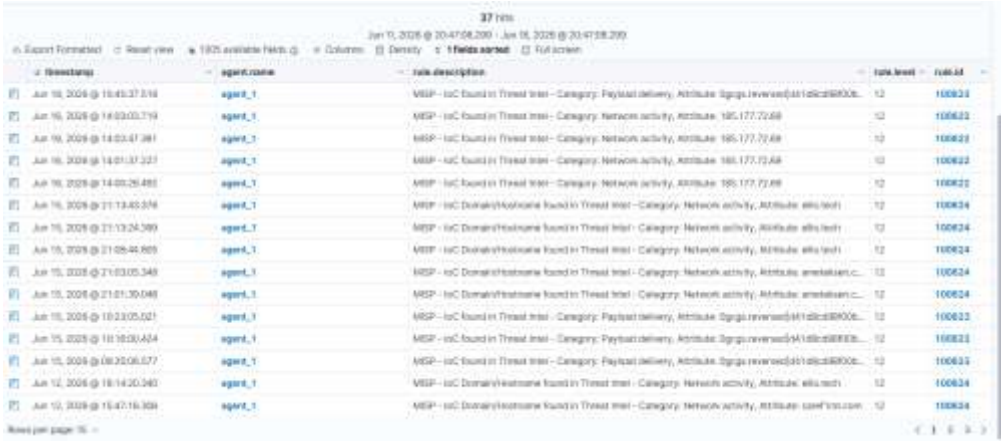
8. Implementasi MISP

Implementasi Malware Information Sharing Platform (MISP) dilakukan sebagai sumber *threat intelligence* pada prototipe *Security Operations Center*. MISP digunakan untuk menyediakan informasi ancaman berupa *Indicators of Compromise (IoC),* seperti alamat IP berbahaya, domain mencurigakan, URL berbahaya, dan *hash file malware* yang dapat dimanfaatkan untuk meningkatkan kemampuan deteksi ancaman pada sistem.

Pada implementasi ini, MISP diintegrasikan dengan Wazuh yang memanfaatkan *log* aktivitas yang dihasilkan oleh Sysmon. Sysmon berfungsi untuk mencatat berbagai aktivitas sistem, seperti eksekusi proses, koneksi jaringan, dan aktivitas file. *Log* yang dihasilkan kemudian dianalisis oleh Wazuh dan dikorelasikan dengan data IoC yang diperoleh dari MISP. Dengan diterapkannya integrasi MISP dan Sysmon, prototipe SOC mampu meningkatkan kemampuan deteksi terhadap ancaman yang telah diketahui sebelumnya *(known threats)*.



Gambar 28 Implementasi MISP

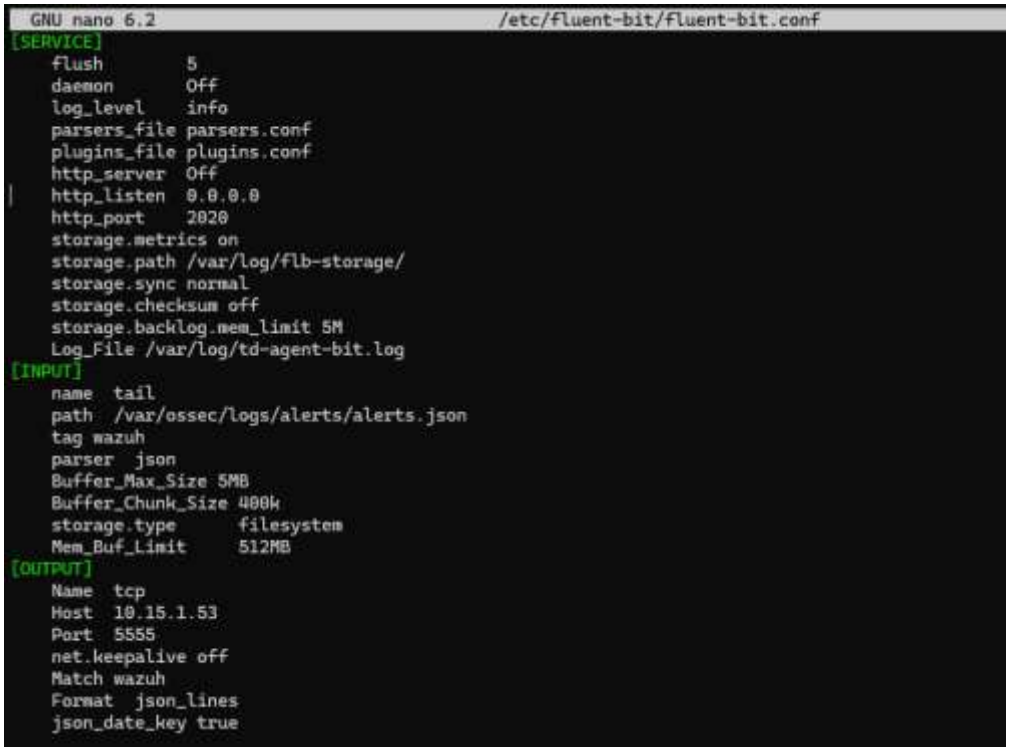


Gambar 29 Hasil Implementasi MISP

Hasil implementasi menunjukkan bahwa data *threat intelligence* dari MISP berhasil digunakan dalam proses korelasi *log* keamanan. Ketika Sysmon mencatat aktivitas yang memiliki kesesuaian dengan indikator ancaman yang tersedia pada MISP, sistem secara otomatis menghasilkan *alert* pada Wazuh. *Alert* tersebut menampilkan informasi mengenai indikator ancaman yang terdeteksi, sumber aktivitas, serta tingkat keparahan ancaman yang terkait.

9. Implementasi Graylog

Graylog digunakan untuk mengumpulkan, menyimpan, mengelola, dan menganalisis *log* yang berasal dari berbagai sumber secara terpusat. Graylog diintegrasikan dengan Fluentbit sebagai agen pengirim *log* yang bertugas mengumpulkan *log* dari berbagai sumber dan meneruskannya ke Graylog secara *real-time*. Dengan diterapkannya Graylog yang terintegrasi dengan Fluentbit, prototipe SOC yang dibangun memiliki kemampuan *centralized logging* yang lebih efisien dan terstruktur, sehingga meningkatkan visibilitas terhadap aktivitas sistem dan mendukung proses deteksi serta analisis ancaman siber secara lebih optimal.



Gambar 30 Implementasi Fluentbit

Seluruh *log* yang diterima kemudian ditampilkan dalam bentuk stream, pencarian *(search)*, dan *dashboard* visualisasi sehingga memudahkan proses pemantauan dan analisis keamanan. Dengan adanya integrasi ini, administrator dapat melakukan pencarian *log*, korelasi, serta investigasi insiden secara lebih

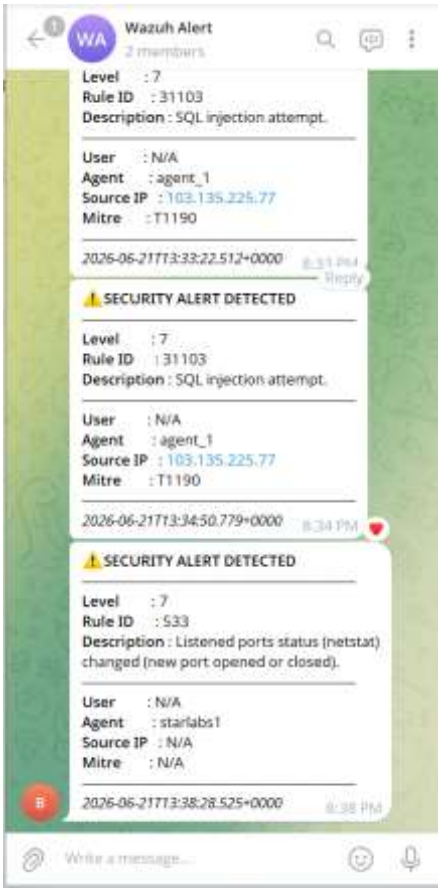
efektif dibandingkan dengan pengelolaan *log* yang tersebar pada masing-masing sistem.

Gambar 31 Hasil implementasi Graylog

10. Implementasi *Alert Notification*

Pada penelitian ini, mekanisme notifikasi diimplementasikan menggunakan Telegram Bot dan *email* berbasis Postfix sebagai media penyampaian informasi keamanan secara *real-time.* Penerapan notifikasi bertujuan untuk mempercepat proses identifikasi dan respons terhadap insiden keamanan tanpa harus melakukan pemantauan *dashboard* secara terus-menerus. Pada tahap implementasi, Telegram Bot dikonfigurasi dan diintegrasikan dengan Wazuh sehingga setiap *alert* yang memenuhi kriteria tertentu dapat dikirimkan secara otomatis ke grup Telegram yang telah dibuat. Selain itu, dilakukan konfigurasi layanan *email* menggunakan Postfix untuk mengirimkan notifikasi melalui surat elektronik. Pengaturan tingkat keparahan *(alert level)* juga dilakukan untuk menentukan jenis *alert* yang akan diteruskan kepada administrator guna mengurangi jumlah notifikasi yang tidak diperlukan.

Gambar 32 Hasil implementasi *email* berbasis Postfix



Gambar 33 Hasil Implementasi Bot Telegram

B. Hasil Pengujian Prototipe SOC

Pada tahap pengujian, dilakukan beberapa skenario pengujian untuk mengevaluasi kinerja prototipe *Security Operations Center (SOC)* yang telah dibangun. Pengujian meliputi kemampuan sistem dalam melakukan *monitoring*, deteksi ancaman, korelasi *log*, serta respons terhadap insiden keamanan. Hasil pengujian disajikan berdasarkan masing-masing fitur yang diimplementasikan dan dianalisis menggunakan metrik *Mean Time To Detect (MTTD), Mean Time To Respond (MTTR),* dan tingkat akurasi deteksi guna mengukur efektivitas sistem dalam mengoptimalkan deteksi dan respons ancaman siber. Berikut merupakan hasil pengujian yang telah dilakukan terhadap prototipe *Security Operations Center*:

1. Hasil Pengujian *File Integrity Monitoring*

   Pengujian *File Integrity Monitoring* dilakukan untuk menguji sejauh mana sensitivitas dan responsivitas Wazuh dalam mendeteksi perubahan status pada file maupun perubahan konfigurasi sistem secara *real-time*. Skenario yang diterapkan mencakup siklus hidup file (pembuatan, modifikasi, penamaan ulang, dan penghapusan) serta perubahan atribut sistem berupa penambahan pengguna baru.

Tabel 7 Hasil pengujian *file integrity monitoring*

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	Akurasi
1	*Create File*	3	0,013	100%
2	*Modify File*	3	0,020	100%
3	*Rename File*	3	0,010	100%
4	*Delete File*	2	0,0085	100%
5	*Add User*	1	0,999	100%

Berdasarkan hasil pengujian pada Tabel 7, seluruh skenario berhasil dideteksi dengan tingkat akurasi 100%. Waktu deteksi tercepat diperoleh pada skenario *Delete File* dengan nilai MTTD sebesar 0,0085 detik, sedangkan skenario *Add User* membutuhkan waktu deteksi paling lama yaitu 0,999 detik. Perbedaan ini menunjukkan bahwa deteksi perubahan file dilakukan lebih cepat dibandingkan perubahan konfigurasi sistem yang memerlukan proses pencatatan tambahan. Secara keseluruhan, hasil tersebut menunjukkan bahwa mekanisme FIM mampu memberikan visibilitas terhadap perubahan integritas file maupun sistem.

2. Hasil Pengujian *Directory Monitoring & Root User*

Pengujian *Directory Monitoring* dan *Root User* dilakukan untuk mengevaluasi kemampuan Wazuh dalam mendeteksi aktivitas yang dilakukan menggunakan hak akses istimewa *(sudo)* pada direktori yang dipantau.

Tabel 8 Hasil Pengujian *Directory Monitoring & Root User*

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	Akurasi
1	*Add File with Sudo*	3	1,481	100%

Berdasarkan hasil pengujian pada Tabel 8, aktivitas pembuatan file menggunakan hak akses *sudo* berhasil menghasilkan *alert* dengan nilai MTTD sebesar 1,481 detik. Waktu deteksi tersebut menunjukkan bahwa sistem mampu mengidentifikasi aktivitas yang dilakukan oleh pengguna dengan hak akses istimewa dalam waktu yang relatif singkat dengan tingakt akurasi deteksi sebesar 100%. Kemampuan ini penting karena aktivitas yang dilakukan oleh akun dengan *privilese* tinggi memiliki risiko yang lebih besar terhadap keamanan dan integritas sistem.

3. Hasil Pengujian Docker *Activity Detection*

Pengujian Docker *Activity Detection* dilakukan untuk mengevaluasi kemampuan Wazuh dalam mendeteksi berbagai aktivitas yang terjadi pada lingkungan Docker secara *real-time*. Skenario pengujian meliputi pengunduhan *image*, pembuatan *container*, menjalankan *container*, eksekusi perintah di dalam *container*, penghentian *container*, dan penghapusan *container*.

Tabel 9 Hasil Pengujian *Docker Activity Detection*

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	Akurasi
1	Docker *Pull Image*	3	0,0067	100%
2	Docker *Create container*	3	1,068	100%
3	Docker *Start Container*	3	0,723	100%
4	Docker *Exec Container*	3	0,088	100%
5	Docker *Stop*	3	1,402	100%
6	Docker *Delete Container*	3	0,250	100%

Berdasarkan hasil pengujian pada Tabel 9, seluruh aktivitas Docker berhasil terdeteksi dengan tingkat akurasi 100%. Waktu deteksi tercepat diperoleh pada aktivitas Docker *Pull Image* dengan nilai MTTD sebesar 0,0067 detik, sedangkan aktivitas Docker *Stop Container* memiliki waktu deteksi tertinggi yaitu 1,402 detik. Perbedaan waktu deteksi tersebut dipengaruhi oleh jenis aktivitas yang dilakukan serta proses pencatatan *event* yang dihasilkan oleh Docker. Meskipun demikian, seluruh aktivitas mulai dari pengunduhan *image*, pembuatan dan pengelolaan *container*, hingga eksekusi perintah di dalam *container* berhasil direkam dan ditampilkan pada *dashboard monitoring*. Hasil ini menunjukkan bahwa integrasi Docker *Monitoring* dengan Wazuh mampu memberikan visibilitas yang baik terhadap aktivitas operasional *container* sehingga dapat mendukung proses pemantauan dan investigasi keamanan pada lingkungan berbasis *container*.

4. Hasil Pengujian *Container Monitoring*

Pengujian *Container Monitoring* dilakukan untuk mengevaluasi kemampuan sistem SOC dalam mendeteksi aktivitas berbahaya yang terjadi pada aplikasi yang berjalan di lingkungan *container*. Skenario pengujian meliputi simulasi serangan *SQL Injection*, *Cross-Site Scripting (XSS),* dan *Command Injection* terhadap aplikasi yang dijalankan di dalam *container* Docker.

Tabel 10 Hasil Pengujian *Container Monitoring*

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	Akurasi
1	*SQL Injection Attempt*	3	1,318	100%
2	*Cross-Site Scripting (XSS)*	3	2,044	100%
3	*Command Injection*	3	1,307	100%

Berdasarkan hasil pengujian pada Tabel 10, seluruh skenario serangan berhasil dideteksi oleh sistem dengan tingkat akurasi 100%. Nilai MTTD tercepat diperoleh pada skenario *Command Injection* sebesar 1,307 detik, sedangkan nilai MTTD tertinggi terdapat pada skenario *Cross-Site Scripting*

sebesar 2,044 detik. Hasil tersebut menunjukkan bahwa sistem mampu melakukan pemantauan dan deteksi terhadap berbagai serangan pada aplikasi berbasis *container* secara efektif, sehingga dapat mendukung proses monitoring dan analisis keamanan pada lingkungan *container*.

5. Hasil Pengujian *Active Response*

Pengujian *Active Response* dilakukan untuk mengevaluasi kemampuan sistem SOC dalam melakukan respons otomatis terhadap aktivitas serangan yang terdeteksi, khususnya pada skenario serangan *SSH brute force*. Pengujian ini bertujuan untuk mengukur kecepatan sistem dalam mendeteksi ancaman serta mengeksekusi tindakan mitigasi berupa pemblokiran alamat IP secara otomatis menggunakan mekanisme *firewall drop.*

Tabel 11 Hasil Pengujian *Active Response*

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	MTTR (detik)	Akurasi
1	*SSH Bruteforce (Low Rate Attempt)*	3	4,912	1,803	100%
2	*SSH Bruteforce (Medium Rate Attempt)*	3	3,487	1,502	100%
3	*SSH Bruteforce (High Rate Attempt)*	3	2,118	1,126	100%

Berdasarkan hasil pengujian, sistem berhasil melakukan deteksi dan respons otomatis terhadap seluruh skenario serangan *SSH brute force* dengan tingkat akurasi 100%. Hasil pengujian menunjukkan bahwa sistem mampu tidak hanya mendeteksi aktivitas mencurigakan, tetapi juga melakukan tindakan mitigasi secara otomatis dalam waktu yang relatif singkat. Hal ini menunjukkan bahwa mekanisme *Active Response* yang diterapkan pada Wazuh berjalan dengan baik dan mampu mendukung proses respons terhadap ancaman siber secara efektif.

6. Hasil Pengujian *Block Malicious Actor*

Pengujian *Block Malicious Actor* dilakukan untuk mengevaluasi kemampuan sistem SOC dalam mendeteksi dan memblokir aktivitas yang berasal dari alamat IP yang terindikasi berbahaya. Mekanisme pengujian ini memanfaatkan respons otomatis *Web application firewall (WAF)* untuk melakukan pemblokiran terhadap *request* yang berasal dari IP bereputasi buruk secara *real-time*.

Tabel 12 Hasil Pengujian Block Malicious Actor

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	MTTR (detik)	Akurasi
1	HTTP Request menggunakan ip bereputasi buruk	3	1.084 detik	0 detik (realtime)	100%

Berdasarkan hasil pengujian, sistem berhasil mendeteksi dan memblokir seluruh request yang berasal dari IP bereputasi buruk dengan tingkat akurasi 100% pada *application layer* melalui pemberian respons HTTP 403. Proses pemblokiran dilakukan secara *real-time* melalui mekanisme *Web Application Firewall,* sehingga respons terhadap ancaman terjadi secara langsung pada saat request diterima. *Nilai Mean Time To Detect (MTTD)* tercatat sebesar 1,084 detik, sedangkan *Mean Time To Respond (MTTR)* bernilai 0 detik karena tindakan mitigasi dilakukan secara otomatis tanpa jeda waktu tambahan. Hasil ini menunjukkan bahwa mekanisme *Active Response* pada sistem SOC mampu memberikan perlindungan secara cepat terhadap aktivitas berbahaya dari sumber eksternal.

7. Hasil Pengujian *Monitoring Resource Usage*

Pengujian *Monitoring Resource Usage* dilakukan untuk mengevaluasi kemampuan sistem SOC dalam mendeteksi anomali pada penggunaan sumber daya sistem, khususnya CPU dan memori. Pengujian ini bertujuan untuk memastikan bahwa sistem mampu mengidentifikasi kondisi abnormal yang dapat mengindikasikan adanya aktivitas serangan atau beban berlebih pada sistem secara *near real-time*.

Tabel 13 Hasil Pengujian Monitoring Resource Usage

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	Anomaly Detection Rate
1	Stress Test CPU/Memory *(Anomaly Detection)*	3	2,474	100%

Berdasarkan hasil pengujian, sistem berhasil mendeteksi seluruh skenario anomali penggunaan CPU dan memori dengan tingkat deteksi 100%. Nilai *Mean Time To Detect (MTTD)* sebesar 2,474 detik menunjukkan bahwa sistem mampu mengidentifikasi kondisi abnormal dalam waktu yang relatif cepat setelah anomali terjadi. Hal ini membuktikan bahwa mekanisme *monitoring resource* pada sistem SOC yang dibangun mampu memberikan visibilitas terhadap kondisi beban sistem secara efektif dan mendukung deteksi dini terhadap potensi gangguan atau aktivitas mencurigakan.

8. Hasil Pengujian *Anomaly Detection*

Pengujian *Anomaly Detection* dilakukan untuk mengevaluasi kemampuan sistem SOC dalam mendeteksi perilaku tidak normal yang terjadi pada penggunaan sumber daya jaringan, khususnya lonjakan trafik web. Pengujian ini bertujuan untuk memastikan bahwa sistem mampu mengidentifikasi kondisi anomali dalam berbagai tingkat beban *(low, medium, dan high load)* secara *near real-time* dengan tetap mempertahankan akurasi deteksi dan tanpa kehilangan *event* selama nilai *log* masih berada di bawah batas *buffer* yang telah ditetapkan.

Tabel 14 Hasil Pengujian Anomaly Detection

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	False Negative	Akurasi
1	*Web Traffic Spike Detection (Low Load)*	3	1,214	0%	100%
2	*Web Traffic Spike Detection (Medium Load)*	3	1,587	0%	100%
3	*Web Traffic Spike Detection (High Load)*	3	2,036	0%	100%

Berdasarkan hasil pengujian, sistem berhasil mendeteksi seluruh skenario anomali lonjakan *trafik* dengan tingkat akurasi 100% dan *false negative rate* sebesar 0%. Nilai MTTD menunjukkan adanya variasi berdasarkan tingkat beban trafik, di mana semakin tinggi intensitas trafik maka waktu deteksi cenderung sedikit lebih lama akibat peningkatan proses pemrosesan *log*. Meskipun demikian, seluruh skenario masih dapat dideteksi secara konsisten selama jumlah event tidak melebihi batas *buffer* yang telah ditetapkan yaitu 5000 *event*. Hal ini menunjukkan bahwa sistem SOC yang dibangun mampu menjaga stabilitas deteksi anomali dalam kondisi beban rendah hingga tinggi, serta tetap memberikan visibilitas terhadap aktivitas tidak normal secara efektif.

9. Hasil Pengujian Pengujian NIDS

Pengujian *Network Intrusion Detection System (NIDS)* dilakukan untuk mengevaluasi kemampuan sistem SOC dalam mendeteksi aktivitas pemindaian jaringan yang bersifat mencurigakan. Pengujian ini menggunakan skenario serangan *port scanning* menggunakan Nmap (-sV) untuk mengidentifikasi layanan yang berjalan pada *host* target. Tujuan pengujian ini adalah untuk mengetahui kemampuan sistem dalam melakukan deteksi aktivitas jaringan yang berbahaya.

Tabel 15 Hasil Pengujian Pengujian NIDS

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	Akurasi
1	Scanning Host Menggunakan Nmap (-sV)	3	44,374 detik	100%

Berdasarkan hasil pengujian, sistem berhasil mendeteksi aktivitas *port scanning* menggunakan Nmap (-sV) dengan tingkat akurasi 100%. Nilai MTTD yang diperoleh sebesar 44,374 detik menunjukkan bahwa proses deteksi membutuhkan waktu yang relatif lebih lama dibandingkan skenario pengujian lainnya. Hal ini disebabkan oleh mekanisme deteksi berbasis korelasi *log*, di

mana sistem memerlukan akumulasi beberapa *event scanning* sebelum *rule* NIDS terpenuhi dan menghasilkan *alert*. Meskipun demikian, seluruh aktivitas pemindaian berhasil terdeteksi tanpa kehilangan *event*, sehingga menunjukkan bahwa integrasi NIDS dengan sistem SOC tetap mampu memberikan visibilitas terhadap aktivitas *scanning* jaringan secara efektif.

10. Hasil Pengujian Pengujian Anti *Malware*

Pengujian Anti *Malware* dilakukan untuk mengevaluasi kemampuan sistem SOC dalam mendeteksi keberadaan file yang terindikasi berbahaya serta melakukan tindakan respons otomatis terhadap ancaman *malware*. Pengujian ini bertujuan untuk memastikan bahwa sistem tidak hanya mampu melakukan deteksi, tetapi juga dapat menjalankan mekanisme mitigasi secara cepat melalui fitur *active response* pada Wazuh.

Tabel 16 Hasil Pengujian Pengujian Anti Malware

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	MTTR (detik)	Akurasi
1	Pengujian Anti Malware	3	4.976 detik	0.351 detik	100%

Berdasarkan hasil pengujian, sistem berhasil mendeteksi seluruh file *malware* yang diuji dengan tingkat akurasi 100%. Nilai MTTD sebesar 4,976 detik menunjukkan bahwa proses deteksi *malware* memerlukan waktu pemrosesan sebelum sistem menghasilkan *alert* keamanan. Selain itu, nilai *Mean Time To Respond (MTTR)* sebesar 0,351 detik menunjukkan bahwa mekanisme *active response* pada Wazuh mampu melakukan tindakan mitigasi secara sangat cepat setelah *alert* dihasilkan. Respons yang dilakukan berupa penghapusan file *malware* secara otomatis sehingga ancaman dapat segera dinetralisir sebelum memberikan dampak lebih lanjut pada sistem.

11. Hasil Pengujian Pengujian *Threat Intelligence*

Pengujian *Threat Intelligence* dilakukan untuk mengevaluasi kemampuan sistem SOC dalam memanfaatkan indikator ancaman *(Indicators of Compromise / IoC)* yang berasal dari MISP dalam proses deteksi aktivitas mencurigakan. Pengujian ini berfokus pada aktivitas resolusi DNS terhadap *domain* yang telah teridentifikasi sebagai IoC untuk mengetahui kemampuan sistem dalam melakukan korelasi antara *log* aktivitas dan data intelijen ancaman.

Tabel 17 Hasil Pengujian Pengujian Threat Intelligence (MISP)

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	Akurasi
1	nslookup *Domain IoC*	3	5,445 detik	100%

Berdasarkan hasil pengujian, sistem berhasil mendeteksi seluruh aktivitas resolusi DNS terhadap *domain* yang terdaftar sebagai IoC pada MISP dengan tingkat akurasi 100%. Nilai MTTD sebesar 5,445 detik menunjukkan bahwa sistem memerlukan waktu pemrosesan untuk melakukan korelasi antara *log* yang dihasilkan oleh sistem dan data *threat intelligence* yang berasal dari MISP.

Hasil ini menunjukkan bahwa integrasi MISP dengan sistem SOC mampu meningkatkan kemampuan deteksi terhadap ancaman yang telah diketahui sebelumnya *(known threats)* melalui mekanisme korelasi IoC. Dengan demikian, sistem dapat memberikan peringatan dini terhadap aktivitas yang berpotensi berbahaya berdasarkan data intelijen ancaman yang tersedia.

12. Hasil Pengujian Pengujian skenario serangan OWASP Top 10 dan MITRE ATT&CK

Pengujian skenario serangan OWASP Top 10 dan MITRE ATT&CK dilakukan untuk mengevaluasi kemampuan sistem SOC dalam mendeteksi berbagai jenis serangan aplikasi web yang umum digunakan oleh penyerang. Pengujian ini mencakup beberapa teknik serangan seperti *file upload* berbahaya, *SQL Injection, Cross-Site Scripting (XSS), File Inclusion*, dan *Command Injection* untuk menguji ketahanan sistem terhadap ancaman pada lapisan aplikasi serta efektivitas mekanisme deteksi dan pencegahan secara *real-time.*

Tabel 18 Hasil Pengujian Pengujian skenario serangan

No.	Sekenario Pengujian	Total Uji	MTTD (detik)	MTTR (detik)	Akurasi
1	*File Upload*	3	1,917	0,000	100%
2	*SQL Injection*	3	1,700	0,000	100%
3	*XSS*	3	2,437	0,000	100%
4	*File Inclusion*	3	1,399	0,000	100%
5	*Command Injection*	3	1,629	0,000	100%

Berdasarkan hasil pengujian, sistem berhasil mendeteksi seluruh skenario serangan yang mewakili OWASP Top 10 dan teknik MITRE ATT&CK dengan tingkat akurasi 100%. *Nilai Mean Time To Detect (MTTD)* menunjukkan variasi waktu deteksi pada setiap jenis serangan. Selain itu, nilai *Mean Time To Respond (MTTR)* sebesar 0 detik menunjukkan bahwa sistem mampu melakukan tindakan pencegahan secara otomatis melalui mekanisme *active response* tanpa adanya keterlambatan setelah serangan terdeteksi. Hal ini menegaskan bahwa integrasi SOC yang dibangun mampu memberikan deteksi dan respons secara *real-time* terhadap berbagai jenis serangan aplikasi web yang umum digunakan dalam skenario pengujian keamanan.

C. Pembahasan Hasil Implementasi dan Pengujian

1. Pembahasan *Monitoring* dan *Log Management*

   Hasil pengujian menunjukkan bahwa prototipe SOC yang dibangun mampu melakukan *monitoring* terhadap berbagai aktivitas sistem secara efektif. Implementasi *File Integrity Monitoring* berhasil mendeteksi perubahan file dan konfigurasi sistem dengan tingkat akurasi yang tinggi, sehingga dapat meningkatkan visibilitas terhadap aktivitas yang terjadi pada *endpoint*. Selain itu, implementasi Auditd dan Docker *Monitoring* memungkinkan sistem untuk mencatat aktivitas pengguna maupun aktivitas *container* secara terpusat. Integrasi Graylog juga mendukung pengelolaan *log* dari berbagai sumber sehingga proses *monitoring* dan analisis keamanan dapat dilakukan dengan lebih mudah.

2. Pembahasan Deteksi Ancaman Siber

   Kemampuan deteksi ancaman pada prototipe SOC ditunjukkan melalui integrasi Wazuh, Suricata, VirusTotal, dan MISP. Suricata berhasil mendeteksi aktivitas pemindaian jaringan dan menghasilkan alert berdasarkan lalu lintas jaringan yang mencurigakan. Integrasi VirusTotal memungkinkan sistem mengidentifikasi file yang terindikasi *malware*, sedangkan integrasi MISP meningkatkan kemampuan deteksi melalui pemanfaatan indikator ancaman (IoC). Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi ancaman dari berbagai sumber dengan tingkat akurasi yang tinggi.

3. Pembahasan Respons Insiden Keamanan

   Pada aspek respons insiden, implementasi *Active Response* dan *Block Malicious Actor* menunjukkan bahwa sistem mampu melakukan mitigasi ancaman secara otomatis melalui mekanisme *firewall drop* dan *Web application firewall (WAF)*. Respons otomatis tersebut memungkinkan aktivitas berbahaya diblokir segera setelah terdeteksi sehingga dapat mengurangi potensi dampak terhadap sistem. Rendahnya nilai MTTR pada beberapa skenario pengujian menunjukkan bahwa mekanisme respons yang diterapkan berjalan dengan efektif.

4. Pembahasan Pengujian Skenario Serangan

   Pengujian berdasarkan skenario OWASP Top 10 dan MITRE ATT&CK menunjukkan bahwa sistem mampu mendeteksi berbagai jenis serangan aplikasi web seperti *SQL Injection, Cross-Site Scripting (XSS), File Inclusion, Command Injection,* dan upload file berbahaya. Selain menghasilkan *alert*, sebagian besar serangan juga berhasil dicegah melalui mekanisme respons otomatis yang telah diterapkan. Hasil tersebut menunjukkan bahwa prototipe SOC mampu mendukung proses deteksi dan respons terhadap ancaman yang umum ditemukan pada lingkungan aplikasi web.

D. Perbandingan Hasil Sebelum dan Sesudah Implementasi

1. Pengujian *SQL Injection*

a) Sebelum Implementasi SOC

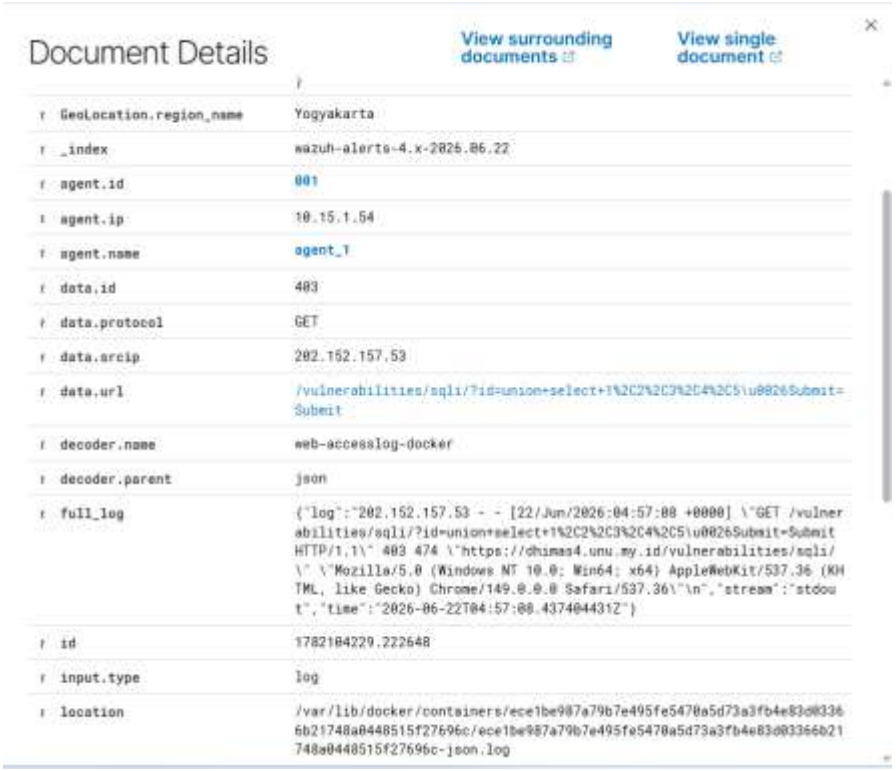
Pengujian *SQL Injection* berhasil dilakukan pada aplikasi yang diuji. Penyerang dapat memanipulasi input pengguna untuk mengakses data yang seharusnya tidak dapat diakses. Aktivitas serangan tidak menghasilkan notifikasi maupun *log* yang dapat digunakan untuk mendeteksi adanya percobaan eksploitasi.

Gambar 34 Hasil Pengujian SQL Injection Sebelum Implementasi SOC

b) Sesudah Implementasi SOC

Setelah implementasi SOC yang terintegrasi dengan mekanisme proteksi, percobaan *SQL Injection* tidak lagi dapat diproses oleh aplikasi. Sistem memberikan respons HTTP 403 *Forbidden* yang menandakan permintaan ditolak. Selain itu, aktivitas serangan berhasil terdeteksi dan menghasilkan *alert* pada Wazuh, sehingga administrator dapat segera melakukan investigasi lebih lanjut.

Gambar 35 Respons HTTP 403 *Forbidden* pada Pengujian *SQL Injection*



Gambar 36 *Alert SQL Injection* Wazuh

2. Pengujian *File Upload Vulnerability*
   a) Sebelum Implementasi SOC

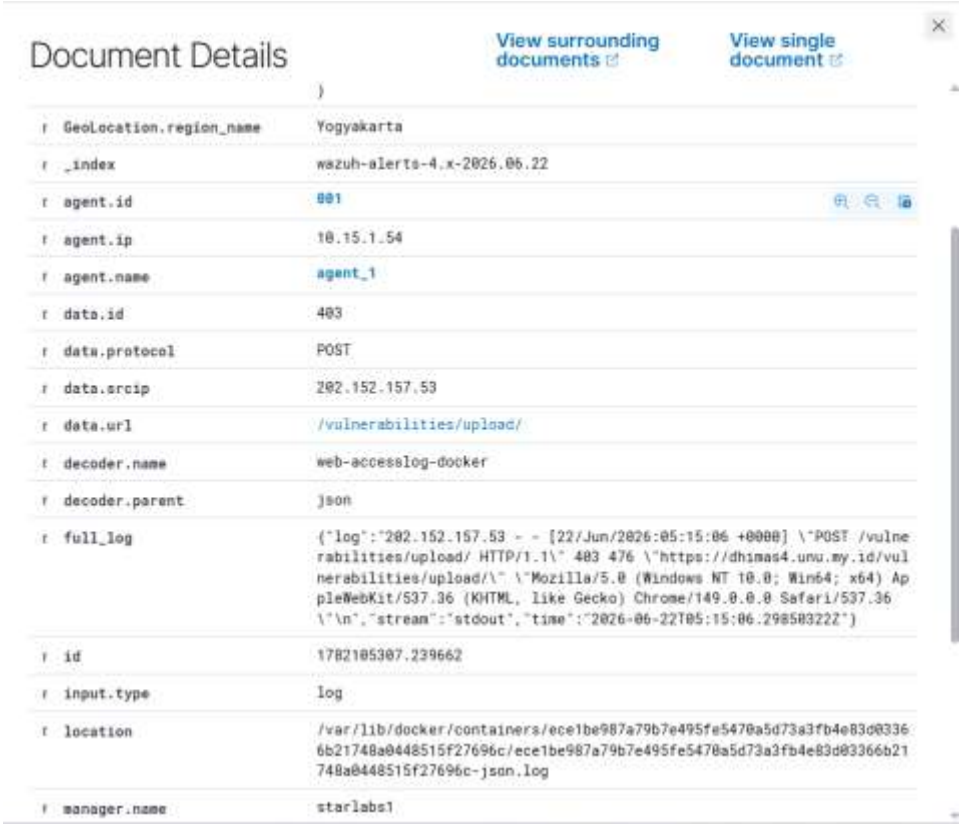
      Pengujian *file upload* menunjukkan bahwa sistem masih mengizinkan unggahan file berbahaya yang berpotensi digunakan untuk memperoleh akses tidak sah ke server. Aktivitas tersebut tidak terdeteksi oleh sistem keamanan.

Gambar 37Hasil Pengujian *File Upload* Sebelum Implementasi SOC

b) Sesudah Implementasi SOC

Setelah implementasi SOC dan mekanisme proteksi, percobaan unggah file berbahaya ditolak oleh sistem dengan respons *403 Forbidden*. Aktivitas upload mencurigakan juga menghasilkan alert pada Wazuh sehingga dapat dimonitor secara *real-time*.

Gambar 38 Respons HTTP 403 *Forbidden* pada Pengujian *File Upload*



Gambar 39 *Alert File Upload Vulnerability* pada Wazuh

3. Pengujian *Command Injection*

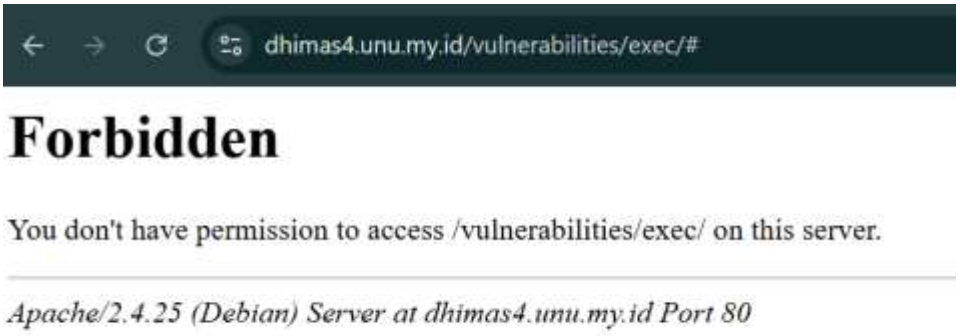
a) Sebelum Implementasi SOC

Pengujian *Command Injection* berhasil dilakukan dengan menyisipkan perintah sistem operasi pada parameter aplikasi. Perintah tersebut dapat dieksekusi tanpa adanya deteksi keamanan.

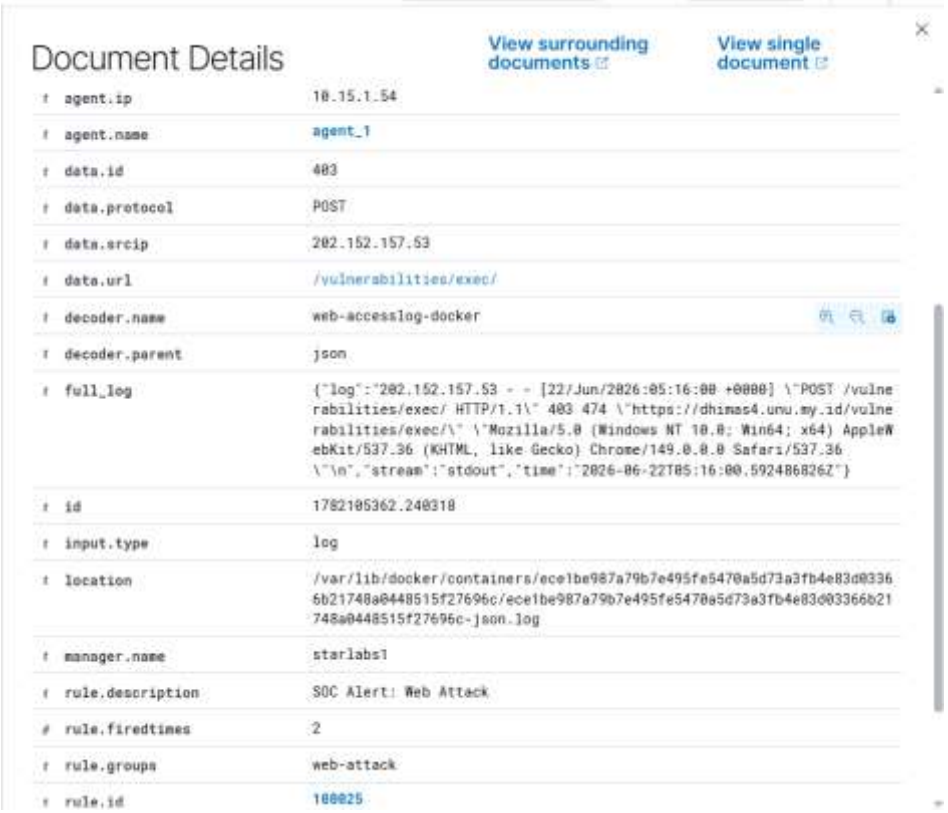
Gambar 40 Hasil Pengujian Command Injection Sebelum Implementasi SOC

b) Sesudah Implementasi SOC

Setelah implementasi SOC dan mekanisme proteksi, percobaan *Command Injection* ditolak oleh sistem dengan respons 403 *Forbidden*. Selain itu, aktivitas serangan berhasil terdeteksi dan menghasilkan *alert* pada Wazuh.



Gambar 41 Respons HTTP 403 *Forbidden* pada Pengujian *Command Injection*



Gambar 42 *Alert Command Injection* pada Wazuh

BAB V KESIMPULAN

A. Kesimpulan

Berdasarkan hasil implementasi, pengujian, dan pembahasan yang telah dilakukan pada bab sebelumnya, dapat disimpulkan sebagai berikut:

1. Prototipe *Security Operations Center* dirancang dan dibangun melalui integrasi berbagai perangkat lunak *open source*, yang mencakup Wazuh sebagai *Intrusion Detection System (IDS), Endpoint Detection and Response (EDR),* serta *Intrusion Prevention System (IPS),* Suricata sebagai *Network Intrusion Detection System (NIDS),* VirusTotal sebagai *Extended Detection and Response (XDR),* MISP sebagai sumber *threat intelligence*, Graylog sebagai *centralized log management*, serta Telegram dan *email* sebagai media notifikasi. Seluruh komponen tersebut diintegrasikan untuk mendukung fungsi utama SOC, yaitu *monitoring*, pengumpulan *log*, deteksi ancaman, *threat intelligence*, dan respons insiden secara terpusat, sehingga prototipe tersebut dapat dikatakan berhasil dikembangkan.
2. Implementasi dan pengujian prototipe SOC dilakukan melalui berbagai skenario, meliputi *File Integrity Monitoring (FIM)*, *Directory Monitoring*, Docker *Monitoring*, *Active Response, Block Malicious Actor, Monitoring Resource Usage, Anomaly Detection*, Anti *Malware*, *Threat Intelligence*, serta skenario serangan berdasarkan OWASP Top 10 dan MITRE ATT&CK. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi berbagai aktivitas dan ancaman keamanan dengan tingkat akurasi yang tinggi serta menjalankan respons otomatis melalui mekanisme *active response,* sehingga prototipe tersebut dapat dinyatakan berhasil dalam mendukung proses deteksi dan respons ancaman siber secara efektif.

B. Saran/Rekomendasi

Mengingat proses penanganan insiden keamanan pada penelitian ini masih memiliki keterbatasan dalam hal otomatisasi yang belum berjalan secara *end-to-end,* sehingga pada beberapa tahapan kritis masih diperlukan intervensi manual, maka disarankan pengembangan penelitian lebih lanjut sebagai berikut:

1. Mengimplementasikan platform *Security Orchestration, Automation, and Response (SOAR)* dan DFIR-IRIS untuk meningkatkan kemampuan otomatisasi dalam proses investigasi serta respons insiden keamanan.

DAFTAR PUSTAKA

[1] F. Pangandaheng, J. B. Maramis, D. P. E. Saerang, L. O. H. Dotulong, And D. Soepeno, "Transformasi Digital: Sebuah Tinjauan Literatur Pada Sektor Bisnis Dan Pemerintah," *Jurnal Emba : Jurnal Riset Ekonomi, Manajemen, Bisnis Dan Akuntansi*, Vol. 10, No. 2, Jun. 2022, Doi: 10.35794/Emba.V10i2.41388.

[2] Y. Ginanjar, "Strategi Indonesia Membentuk Cyber Security Dalam Menghadapi Ancaman Cyber Crime Melalui Badan Siber Dan Sandi Negara," *Jurnal Dinamika Global*, Vol. 7, No. 02, Pp. 291–312, Dec. 2022, Doi: 10.36859/Jdg.V7i02.1187.

[3] John Zorabedian, "Surging Data Breach Disruption Drives Costs To Record Highs," Https://Www.Ibm.Com/Think/Insights/Whats-New-2024-Cost-Of-A-Data-Breach-Report.

[4] S. A. Chamkar, Y. Maleh, And N. Gherabi, "Security Operations Centers: Use Case Best Practices, Coverage, And Gap Analysis Based On Mitre Adversarial Tactics, Techniques, And Common Knowledge," *Journal Of Cybersecurity And Privacy*, Vol. 4, No. 4, Pp. 777–793, Dec. 2024, Doi: 10.3390/Jcp4040036.

[5] R. Amami, M. Charfeddine, And S. Masmoudi, "Exploration Of Open Source Siem Tools And Deployment Of An Appropriate Wazuh-Based Solution For Strengthening Cyberdefense," In *2024 10th International Conference On Control, Decision And Information Technologies (Codit)*, Ieee, 2024, Pp. 1–7.

[6] Socradar, "Top 10 Best Free Cyber Threat Intelligence Sources And Tools In 2025," Https://Socradar.Io/. Accessed: Oct. 10, 2025. [Online]. Available: Https://Socradar.Io/Top-10-Free-Cyber-Threat-Intelligence-Sources-And-Tools-2025/

[7] M. Vielberth, F. Bohm, I. Fichtinger, And G. Pernul, "Security Operations Center: A Systematic Study And Open Challenges," *Ieee Access*, Vol. 8, Pp. 227756–227779, 2020, Doi: 10.1109/Access.2020.3045514.

[8] Kevin. Mclaughlin, *Cybersecurity Operations And Fusion Centers : A Comprehensive Guide To Soc And Tic Strategy*, 1st Edition. Boca Raton, Fl: Crc Press, 2024.

[9] E-Spincorp Documentation, "Setting Up A Comprehensive Open-Source Security Operations Center: A Practical Guide," Https://Www.E-Spincorp.Com/. Accessed: Oct. 10, 2025. [Online]. Available: Https://Www.E-Spincorp.Com/Documentation/Open-Source-Soc-Setup-Guide/

[10] J. Forsberg And T. Frantti, “Technical Performance Metrics Of A Security Operations Center,” *Comput. Secur.*, Vol. 135, Dec. 2023, Doi: 10.1016/J.Cose.2023.103529.

[11] E. Du Bois And I. Horváth, “Abstract Prototyping In Software Engineering: A Review Of Approaches,” 2011.

[12] N. A. Rakhmawati, “Software Open Source, Software Gratis?,” *Juti: Jurnal Ilmiah Teknologi Informasi*, Pp. 13–18, Jan. 2006, Doi: 10.12962/J24068535.V5i1.A201.

[13] S. Salamun And S. Sukri, “Analisa Pemanfaatan Dan Peran Software Open-Source Bagi Mahasiswa Universitas Abdurrab,” *Jurnal Buana Informatika*, Vol. 12, No. 1, Pp. 49–57, May 2021, Doi: 10.24002/Jbi.V12i1.4145.

[14] L. Shitao And E. D. Martin, “Optimization And Zspore Analysis Of Affinity Purification Coupled With Tandem Mass Spectrometry In Mammalian Cells,” *J. Proteom. Genom. Res.*, Vol. 1, No. 1, Pp. 9–20, 2012, Doi: 10.14302/Issn.2326-0793.Jpgr-12-100.

[15] M. R. T. Hidayat, N. Widiyasono, And R. Gunawan, “Optimasi Deteksi Malware Pada Siem Wazuh Melalui Integrasi Cyber Threat Intelligence Dengan Misp Dan Dfir-Iris,” *Jurnal Informatika Dan Teknik Elektro Terapan*, Vol. 13, No. 1, Jan. 2025, Doi: 10.23960/Jitet.V13i1.5686.

[16] R. Kanna R, *Cybersecurity Threats And Incident Response: Real-World Case Studies On Network Security, Data Breaches, And Risk Mitigation*, 1st Ed. Bangalore, Karnataka, India: Deep Science Publishing, 2025. Doi: 10.70593/978-93-7185-133-6.

[17] M. Y. Darus, A. Ramli, Y. Mohd Yussoff, And B. Azni, “Cybersecurity Resilience — An Integrated Framework For Detection Of Threats And Response In The Digital Age: A Review Paper,” *Malaysian Journal Of Computing*, Vol. 10, No. 1, Pp. 2099–2116, Apr. 2025, Doi: 10.24191/Mjoc.V10i1.4520.

[18] Courage Ojo, Emmanuel Ayodeji Osoko, Joy Nnenna Okolo, And Mamudat Jaji, “Incident Response: A Structured Model From Detection To Containment And Recovery,” *World Journal Of Advanced Research And Reviews*, Vol. 24, No. 1, Pp. 1401–1407, Oct. 2024, Doi: 10.30574/Wjarr.2024.24.1.3148.

[19] A. Sutanto And A. Rakhman, “Experimental Evaluation Of Wazuh-Grafana Integration For Real-Time Cyber Threat Detection In Resource-Constrained Environments,” *Journal Of Applied Informatics And Computing*, Vol. 9, No. 5, Pp. 2783–2790, Oct. 2025, Doi: 10.30871/Jaic.V9i5.10404.

[20] M. Ammi, “Cyber Threat Hunting Case Study Using Misp,” *Journal Of Internet Services And Information Security*, Vol. 13, No. 2, Pp. 1–29, May 2023, Doi: 10.58346/Jisis.2023.I2.001.

[21] G. Dahlberg, “Exploring Incident Management: A Comparative Study Of Splunk Enterprise, Graylog Open And Syslog-Ng And Their Impact On Mean Time To Resolution (Mttr) An Experimental Study Of Incident Detection And Alerting Performance Across Basic, Intermediate, And Advanced Siem Systems.”

[22] O. Owolafe And A. J. Alabi, “Comparative Analysis Of Wazuh, Graylog, And Elk Stack For Cyber Threat Detection And Response,” In *International Conference For Internet Technology And Secured Transactions (Icitst-2024)*, Infonomics Society, Dec. 2024, Pp. 182–193. Doi: 10.20533/Icitst.2024.0015.

[23] M. Husák, J. Komárková, E. Bou-Harb, And P. Čeleda, “Survey Of Attack Projection, Prediction, And Forecasting In Cyber Security,” *Ieee Communications Surveys & Tutorials*, Vol. 21, No. 1, Pp. 640–660, 2019, Doi: 10.1109/Comst.2018.2871866.

[24] E. M. Hutchins, M. J. Cloppert, And R. M. Amin, “Intelligence-Driven Computer Network Defense Informed By Analysis Of Adversary Campaigns And Intrusion Kill Chains.”

[25] M. Khayat, E. Barka, M. A. Serhani, F. Sallabi, K. Shuaib, And H. M. Khater, “Advanced Techniques For Alert Management In Security Information And Event Management Systems With Ensembled Deep Learning, Hybrid Optimization, And Multi-Feature Extraction,” *Ieee Open Journal Of The Communications Society*, Vol. 6, Pp. 7349–7368, 2025, Doi: 10.1109/Ojcoms.2025.3603000.

[26] R. Ramadhani, A. Christie, A. A. Dewani, M. Krisnawan, And R. Dhimaz Ardhana, “Optimalisasi Intelijen Ancaman Siber Di Security Operation Centers Dengan Pemanfaatan Large Language Models (Llm) Dan Siem,” *Jurnal Ilmiah Multidisipline*, Vol. 3, No. 5, 2025, Doi: 10.5281/Zenodo.15707795.

[27] K. Bezas And F. Filippidou, “Comparative Analysis Of Open Source Security Information &Amp; Event Management Systems (Siems),” *The Indonesian Journal Of Computer Science*, Vol. 12, No. 2, Pp. 443–468, Apr. 2023, Doi: 10.33022/Ijcs.V12i2.3182.

[28] H. Kaur *Et Al.*, “Evolution Of Endpoint Detection And Response (Edr) In Cyber Security: A Comprehensive Review,” *E3s Web Of Conferences*, Vol. 556, P. 01006, Aug. 2024, Doi: 10.1051/E3sconf/202455601006.

[29] A. Hozouri, A. Mirzaei, And M. Effatparvar, “A Comprehensive Survey On Intrusion Detection Systems With Advances In Machine Learning, Deep

Learning And Emerging Cybersecurity Challenges," *Discover Artificial Intelligence*, Vol. 5, No. 1, P. 314, Nov. 2025, Doi: 10.1007/S44163-025-00578-1.

[30] H. Debar, M. Dacier, And A. Wespi, "A Revised Taxonomy For Intrusion-Detection Systems," *Ann. Telecommun.*, Vol. 55, No. 7–8, Pp. 361–378, Jul. 2000, Doi: 10.1007/Bf02994844.

[31] Barlyan Kurdianto, Yohanzah Febriyanto, And Yustian Servanda, "Intrusion Detection System Analysis To Improve Computer Network Security," *Journal Of Artificial Intelligence And Engineering Applications (Jaiea)*, Vol. 4, No. 3, Pp. 1855–1862, Jun. 2025, Doi: 10.59934/Jaiea.V4i3.1036.

[32] S. Ang, S. Huy, And M. Janarthanan, "Utilizing Ids And Ips To Improve Cybersecurity Monitoring Process," *Journal Of Cyber Security And Risk Auditing*, Vol. 2025, No. 3, Pp. 77–88, Jul. 2025, Doi: 10.63180/Jcsra.Thestap.2025.3.10.

[33] Dwi Putri Amanda And Eriene Dheanda Absharina, "Implementasi Ai-Powered Intrusion Detection Systems Untuk Mendeteksi Ancaman Keamanan Pada Big Data," *Simtek : Jurnal Sistem Informasi Dan Teknik Komputer*, Vol. 10, No. 1, Pp. 29–33, Apr. 2025, Doi: 10.51876/Simtek.V10i1.1381.

[34] R. V. Rishika, B. A. Pratomo, And S. C. Hidayati, "Network Intrusion Detection System With Time-Based Sequential Cluster Models Using Lstm And Gru," *Juti: Jurnal Ilmiah Teknologi Informasi*, Pp. 1–12, Feb. 2025, Doi: 10.12962/J24068535.V23i1.A1241.

[35] M. Ahmed And Q. Abdullah, "Network Intrusion Detection Systems: Machine Learning-Based Attack And Remedy Strategies – A Review," *Al-Salam Journal For Engineering And Technology*, Vol. 4, No. 2, Pp. 11–29, Aug. 2025, Doi: 10.55145/Ajest.2025.04.02.002.

[36] M. Farhan *Et Al.*, "Network-Based Intrusion Detection Using Deep Learning Technique," *Sci. Rep.*, Vol. 15, No. 1, P. 25550, Jul. 2025, Doi: 10.1038/S41598-025-08770-0.

[37] M. K. Paul Kirvansean, "What Is Extended Detection And Response (Xdr)?," Https://Www.Techtarget.Com/Searchsecurity/Definition/Extended-Detection-And-Response-Xdr.

[38] Anne Aarness, "Extended Detection And Response (Xdr) Explained," Https://Www.Crowdstrike.Com/En-Us/Cybersecurity-101/Endpoint-Security/Extended-Detection-And-Response-Xdr/.

[39] Ibm, "What Is Extended Detection And Response (Xdr)?," Https://Www.Ibm.Com/Think/Topics/Xdr?

[40] G. Ahn, J. Jang, S. Choi, And D. Shin, “Research On Improving Cyber Resilience By Integrating The Zero Trust Security Model With The Mitre Att&Ck Matrix,” *Ieee Access*, Vol. 12, Pp. 89291–89309, 2024, Doi: 10.1109/Access.2024.3417182.

LAMPIRAN

A. Surat Permohonan Observasi

UNUGHA CILACAP
FAKULTAS MATEMATIKA DAN ILMU KOMPUTER (FMIKOM)
Keputusan Kemendikbud RI Nomor : 264/E/O/2014 Tanggal 23 Juli 2014

Nomor : B/054/Ybk.041.10/PT/2025 Cilacap, 02 Februari 2026
Lampiran : -
Hal : **Permohonan Observasi**

Yth.
Rektor Universitas Nahdlatul Ulama Al Ghazali Cilacap
di –
Tempat

Assalamu'alaikum Wr.Wb.

Salam silaturahmi dan sejahtera kami sampaikan semoga kita senantiasa mendapatkan ridlo dan pertolongan dari Allah SWT dalam menjalankan aktivitas sehari-hari. Amin.

Sehubungan dengan tuntutan kebutuhan mahasiswa untuk mendapatkan pengalaman nyata di lapangan, maka Fakultas Matematika dan Ilmu Komputer Universitas Nahdlatul Ulama Al Ghazali Cilacap menugaskan kepada mahasiswa kami:

Nama : Dimas Galih Laksono
NIM : 22EO10019
Prodi : Informatika

Untuk mengadakan riset terkait dengan skripsi yang sedang di kerjakan dengan Judul "*Prototyping Security Operations Center* Berbasis integrasi *software Open-Source* untuk Optimasi Deteksi dan Respons Ancaman Siber di UPT Sistem dan Sumber Daya Informasi [SSDI] Universitas Nahdlatul Ulama Al Ghazali Cilacap". Berkenaan dengan hal tersebut, maka kami mengajukan permohonan kepada Bapak Rektor Universitas Nahdlatul Ulama Al Ghazali Cilacap Utama untuk mengizinkan mahasiswa kami melaksanakan kegiatan tersebut.

Demikian surat ini kami sampaikan, atas bimbingan dan kerjasamanya disampaikan terimakasih.

Wassalamu'alaikum Wr.Wb.

Dekan,

Mochamad T.A. Aziz Zein, M.Kom
NIK. 41 230714 020

UNIVERSITAS NAHDLATUL 'ULAMA AL GHAZALI CILACAP
Jln. Kemerdekaan Barat No.17 Kesugihan Cilacap Jawa Tengah K.Pos 53274, http:// unugha.ac.id,
Email : fmikom@unugha.ac.id Telp. : (0282) 595415, 595407, Fax : (0282) 595407

B. Dokumentasi Hasil Pengujian

Notions: https://versed-skiff-577.notion.site/38662f99be3e80af8110ffff0c020fb6?v=38662f99be3e803dbbce000c98583175&pvs=74