

BAB 2

TINJAUAN PUSTAKA

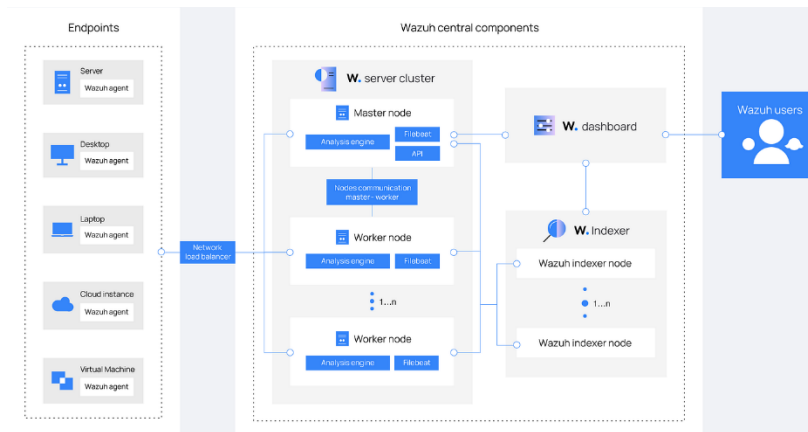
2.1 Landasan Teori

2.1.1 Security Information Event Management (SIEM)

Security Information and Event Management (SIEM) merupakan sistem yang digunakan untuk mengumpulkan, menyimpan, dan menganalisis data terkait keamanan dari berbagai sumber secara terpusat. Teknologi ini memungkinkan pendeteksian ancaman atau insiden siber secara real-time maupun historis melalui korelasi berbagai log kejadian. SIEM juga menyediakan fungsi tambahan seperti normalisasi data dari sumber yang beragam, pengayaan informasi kontekstual, pembuatan laporan, pemberian notifikasi otomatis, hingga kemampuan untuk merespons insiden secara langsung. Selain itu, SIEM mendukung integrasi dengan platform lain guna pertukaran informasi ancaman, serta menyajikan visualisasi analisis jaringan yang membantu pengguna dalam memahami kondisi keamanan sistem. Dalam jangka panjang, SIEM juga berfungsi sebagai sarana manajemen log yang efektif, dengan kemampuan menyimpan data peristiwa untuk keperluan audit dan investigasi. (Kamal, 2022)

2.1.2 Wazuh

Wazuh merupakan sebuah tools SIEM *Open Source* yang berfungsi sebagai sistem deteksi intrusi yang berbasis host (endpoint). *Wazuh* merupakan sebuah aplikasi monitoring yang berfungsi untuk mendeteksi ancaman pada server, memonitor integritas server, hingga melaporkan insiden yang ada pada server. *Wazuh* terdiri dari 2 (dua) bagian yaitu *Wazuh Server* dan *Wazuh Agent*. *Wazuh* server merupakan perangkat yang digunakan sebagai manajemen agen dan dashboard sistem monitoring baik file integrity, intrusion, maupun log. Sedangkan *Wazuh* agent merupakan perangkat yang diinstall pada perangkat endpoint untuk melakukan pembacaan sistem, pengumpulan log serta mengirimkan ke *Wazuh* server. (Azzah, 2024)



Gambar 2. 1 Arsitektur SIEM

Wazuh mempunyai beberapa komponen, berikut komponen dan penjelasan Wazuh:

1. Wazuh indexer

Mesin analitik dan pencarian teks lengkap yang sangat skalabel. Komponen ini berfungsi untuk mengindeks dan menyimpan peringatan yang dihasilkan oleh *server* Wazuh.

2. Wazuh Server

Memiliki fungsi untuk menganalisa data yang didapat dari agent, lalu melakukan decoder dan menyesuaikan dengan aturan standar keamanan, dengan menggunakan threat intelligence untuk mencari *indicators of compromise* (IOC) yang terkenal. Satu server dapat menganalisis data dari ratusan atau ribuan agen, dan memperluas secara horizontal Ketika disiapkan dalam sebuah lingkungan. Komponen sentral ini juga digunakan untuk mengelola agen, mengkonfigurasi dan melakukan upgrade dari jarak jauh bila diperlukan.

3. Wazuh dashboard

Wazuh Dashboard adalah antarmuka pengguna web untuk visualisasi dan analisis data. Ini mencakup seluruh dasbor yang siap digunakan untuk perihal keamanan, pemenuhan terhadap peraturan, aplikasi yang terdeteksi kerentanan, pemantauan integritas file, hasil penilaian konfigurasi, pemantauan infrastruktur cloud, dan lain-lain. Ini juga digunakan untuk mengelola konfigurasi Wazuh dan untuk memantau statusnya.

4. Wazuh Agents

Wazuh *Agent* adalah komponen yang diinstal di sistem yang ingin dimonitor. Tugas utama agen adalah mengumpulkan data keamanan dari sistem tersebut dan mengirimkannya ke Wazuh *Manager*. Ada beberapa jenis agen, termasuk OSSEC *Agent* (sistem operasi), filebeat *Agent*, dan *Agent Docker*

2.1.3 Security Operations Center (SOC)

Security Operations Center atau biasa dikenal dengan *Security Operations Center* (SOC) menyelidiki pesan peringatan untuk mengetahui apakah berbahaya atau tidak. Dalam hal ini, SOC sangat bergantung pada respon, analisa dan pengambilan Tindakan yang tepat untuk melindungi data dan aset perusahaan atau perusahaan (Yuswanto dkk., 2023) . Berikut adalah diagram alur kerja atau tugas dari SOC pada gambar 2.2.



Gambar 2. 2 Diagram Fungsi SOC

Security Operations Center (SOC) adalah fasilitas di mana tim menjalankan tugas-tugas keamanan perusahaan. Ini mencakup pencegahan, respons terhadap insiden, pemantauan, dan pengumpulan data keamanan. Di dalam SOC, ada juga upaya untuk mendeteksi ancaman dan menganalisis perilaku guna menjaga keamanan jaringan dan system mampu mencegah serangan siber.

2.1.4 Incident Response Lifecycle

Incident response merupakan bagian penting dalam pengelolaan keamanan siber yang bertujuan membantu organisasi mempersiapkan, mendeteksi, merespons, dan melakukan pemulihan terhadap insiden keamanan. Proses *incident response* diperlukan untuk mengurangi dampak yang ditimbulkan oleh insiden serta

mendukung penanganan insiden secara terstruktur.

Menurut National Institute of Standards and Technology (NIST), *incident response* perlu diintegrasikan ke dalam aktivitas pengelolaan risiko keamanan siber untuk meningkatkan efektivitas dalam menghadapi insiden. Proses tersebut mencakup aktivitas yang mendukung persiapan, deteksi, respons, dan pemulihan terhadap insiden keamanan (Nelson dkk, 2025) .

2.1.5 MITRE ATT&CK Framework

MITRE ATT&CK Framework merupakan basis pengetahuan yang digunakan untuk menggambarkan teknik dan taktik yang digunakan oleh *adversary* dalam melakukan serangan siber. Framework ini dapat digunakan untuk membantu mengidentifikasi dan mengklasifikasikan aktivitas serangan berdasarkan teknik yang digunakan. Pada penelitian ini, pemetaan serangan dilakukan berdasarkan karakteristik dari masing-masing skenario pengujian.

Serangan *SSH Brute Force* dapat dikaitkan dengan teknik T1110.001 (*Password Guessing*), yaitu aktivitas menebak kredensial secara berulang untuk memperoleh akses ke suatu akun. Teknik ini juga mencakup layanan SSH sebagai salah satu layanan yang umum menjadi target percobaan *password guessing* (MITRE, 2026a).

Serangan *Port Scanning* dapat dikaitkan dengan teknik T1046 (*Network Service Discovery*), yaitu aktivitas untuk memperoleh informasi mengenai layanan yang berjalan pada suatu *host* atau perangkat jaringan. Teknik tersebut mencakup penggunaan *port scanning* untuk mengidentifikasi layanan yang tersedia pada sistem target (MITRE, 2026c)

Sementara itu, serangan *SQL Injection* dapat dikaitkan dengan teknik T1190 (*Exploit Public-Facing Application*), yaitu eksploitasi kelemahan pada sistem atau aplikasi yang dapat diakses melalui jaringan. MITRE menjelaskan bahwa aplikasi yang menjadi target eksploitasi dapat berupa *web server* maupun basis data yang dapat diakses melalui jaringan (MITRE, 2026b)

2.1.6 Security Orchestration, Automation, and Response (SOAR)

Security Orchestration, Automation, and Response (SOAR) merupakan pendekatan keamanan siber yang digunakan untuk mengintegrasikan berbagai

perangkat dan proses keamanan dalam suatu alur kerja yang terkoordinasi. SOAR mencakup proses orkestrasi perangkat keamanan, otomatisasi tugas, serta pelaksanaan tindakan respons terhadap insiden berdasarkan *workflow* yang telah ditentukan. Penerapan SOAR dapat membantu meningkatkan efisiensi dalam pengelolaan insiden dengan mengurangi pekerjaan manual dan mempercepat proses respons terhadap ancaman (Gustina DM & Ananda, 2024).

SOAR dapat diintegrasikan dengan sistem *Security Information and Event Management* (SIEM) untuk menerima dan memproses *alert* keamanan yang dihasilkan dari proses deteksi. Informasi tersebut selanjutnya dapat digunakan untuk menjalankan *workflow* respons secara otomatis sesuai dengan kondisi yang telah ditentukan. Dengan mekanisme tersebut, proses penanganan insiden dapat dilakukan secara terstruktur, mulai dari penerimaan *alert*, pemrosesan informasi, hingga pelaksanaan tindakan respons (Aljahdali & Alsulami, 2025).

2.1.7 Automated Response

Automated response merupakan mekanisme pemberian respons terhadap insiden keamanan dengan menjalankan tindakan yang telah ditentukan berdasarkan hasil deteksi dan kondisi tertentu. Mekanisme ini memungkinkan tindakan terhadap insiden dilakukan secara otomatis tanpa memerlukan intervensi manual pada setiap tahapan proses respons. Konsep *automated response* berkaitan dengan *Security Orchestration, Automation, and Response* (SOAR), yang digunakan untuk mengoordinasikan proses keamanan dan mengotomatisasi tindakan respons terhadap insiden (Aljahdali & Alsulami, 2025).

Penerapan *automated response* dapat mencakup berbagai tindakan keamanan, seperti pemblokiran alamat IP, isolasi perangkat, penonaktifan akun, atau pengiriman notifikasi kepada administrator. Tindakan tersebut dijalankan berdasarkan aturan, kondisi, atau *workflow* yang telah ditentukan sebelumnya. Dengan demikian, *automated response* dapat membantu mempercepat proses penanganan insiden dan mengurangi ketergantungan terhadap tindakan manual dalam proses respons keamanan.

2.1.8 Shuffle

Shuffle SOAR adalah platform keamanan dengan konsep *Open Source* dan fokus pada *Security Orchestration, Automation and Response* (SOAR). Tujuan utama *Shuffle* SOAR adalah membuat manajemen insiden keamanan menjadi lebih sederhana dan sederhana. Dalam upaya mencapai tujuan tersebut, *Shuffle* SOAR menawarkan beberapa fitur penting yang dapat memberikan solusi terhadap berbagai tantangan yang dihadapi dalam dunia keamanan digital. Hal ini memungkinkan pengguna untuk mengelola insiden keamanan yang mungkin timbul dalam organisasi mereka secara lebih efektif (Habibie, 2024).

Workflow dan aplikasi, *Shuffle* SOAR membantu pengguna mengintegrasikan dan mengotomatiskan tindakan keamanan dengan lebih mudah dan cepat. Menggunakan model *Open Source*, *Shuffle* SOAR memungkinkan pengembangan yang cepat dan berkelanjutan. Oleh karena itu, *Shuffle* SOAR adalah platform keamanan yang menggabungkan kemudahan penggunaan, fleksibilitas, dan inovasi berkat pendekatan *Open Source*. Ini memberikan solusi terbaik untuk mengelola insiden keamanan yang semakin kompleks dan memungkinkan organisasi menjaga keamanan digital mereka dengan lebih efektif.

2.1.9 Firewall

Firewall yaitu suatu kombinasi antara Hardware (perangkat keras) dan Software (perangkat lunak) yang fungsinya menjadi pemisah diantara jaringan komputer menjadi dua atau lebih untuk menjaga keamanan data. Pengertian lain dari Firewall adalah suatu sistem keamanan pada jaringan komputer yang dipakai untuk melindungi komputer dari beberapa serangan dari komputer luar.

Firewall merupakan sebuah sistem yang didesain untuk mencegah akses yang tidak sah ke atau dari jaringan pribadi (Private Network). Implementasi Firewall pada perangkat keras (Hardware) dan perangkat lunak (Software) atau kombinasi antara keduanya. Penggunaan dari Firewall adalah membatasi atau menjadi pengontrol kepada siapa saja yang memiliki akses ke jaringan pribadi dari jaringan luar. Firewall mengacu kepada sistem pengatur komunikasi antara dua jenis jaringan yang tidak sama. (Jaya dkk, 2021).

2.1.10 Fortigate

Fortigate Firewall Security adalah solusi keamanan jaringan yang sangat komprehensif dan canggih, menawarkan berbagai fitur yang dirancang untuk melindungi jaringan dari berbagai ancaman keamanan. Dengan kemampuan analisis lalu lintas jaringan yang mendalam, autentikasi dua faktor, pencegahan intrusi, dan kontrol aplikasi, Fortigate memastikan bahwa hanya lalu lintas yang diizinkan yang dapat mengakses jaringan. Selain itu, fitur seperti antivirus, anti malware, sandboxing, dan proteksi ancaman lanjutan memberikan lapisan keamanan tambahan untuk menangani ancaman yang lebih kompleks. Integrasi dengan solusi keamanan lain melalui SIEM dan proteksi cloud semakin memperkuat kemampuan Fortigate dalam memberikan perlindungan menyeluruh terhadap serangan siber. Implementasi Fortigate Firewall Security yang tepat akan secara signifikan meningkatkan keamanan dan integritas jaringan, melindungi data, dan menjaga privasi informasi dari berbagai ancaman yang mungkin timbul (Fortigate, 2025).

2.1.11 Linux Ubuntu

Linux adalah sistem operasi yang dikembangkan bekerja sama dengan banyak kontributor dari seluruh dunia. Berdasarkan kernel Linux, sistem operasi ini dikenal dengan sifat Open Source, memungkinkan siapa saja untuk melihat, memodifikasi, dan mendistribusikan ulang kode sumber di bawah lisensi Open Source. Linux sangat fleksibel, multitasking dan stabil, sehingga sering digunakan di lingkungan server dan super komputer. Ubuntu, di sisi lain, adalah salah satu distribusi Linux paling populer. Dikembangkan oleh Canonical Ltd., Ubuntu dirancang untuk menjadi sistem operasi yang mudah digunakan dan cocok untuk berbagai pengguna, dari pemula hingga pengguna berpengalaman.

Dengan antarmuka yang ramah pengguna, pembaruan rutin, ketersediaan aplikasi yang kaya, dan dukungan komunitas yang aktif, Ubuntu adalah pilihan yang sering digunakan sebagai sistem operasi desktop. Namun, ini juga digunakan di lingkungan server dan komputasi awan. Ubuntu tersedia dalam berbagai varian, seperti Ubuntu Desktop, Ubuntu Server, dll, untuk memenuhi berbagai kebutuhan pengguna. (Khairil dkk, 2024)

2.1.12 Server

Server adalah solusi hosting di mana sebuah server fisik dibagi menjadi beberapa lingkungan virtual yang beroperasi secara terisolasi. Setiap server berperilaku seperti server fisik mandiri, meskipun sebenarnya mereka berbagi sumber daya fisik yang sama. Dengan server, pengguna memiliki kendali penuh atas konfigurasi server dan sistem operasi di dalam lingkungan virtual mereka. Keamanan dan stabilitas dijaga dengan baik karena setiap server beroperasi terpisah. Selain itu, server juga memungkinkan pengguna untuk memanfaatkan sumber daya perangkat keras secara lebih efisien, menjadikannya solusi yang lebih efisien biaya daripada memiliki server fisik sendiri. Keunggulan lainnya adalah skalabilitas, dimana pengguna dapat dengan mudah menyesuaikan sumber daya sesuai kebutuhannya. server digunakan untuk berbagai tujuan seperti hosting situs web, aplikasi, server email, dan lainnya, memberikan fleksibilitas dan kontrol yang lebih besar. (Harsono, 2022)

2.1.13 Container

Container adalah beberapa instance userspace yang terpisah dari suatu sistem operasi yang sedang berjalan. Kernel dari sistem operasi bertanggung jawab untuk membangun Container dan memberikan mekanisme manajemen sumber daya untuk memberikan operasi yang "seamless" dan "concurrent" dari beberapa Container tersebut. Teknik ini tidak memberikan fleksibilitas yang sama seperti teknik virtualisasi yang lainnya, terutama dalam hal penggunaan sistem operasi yang berbeda versi maupun distribusinya. Namun, teknik ini memberikan beban atau overhead yang jauh lebih rendah karena Container dapat menggunakan antarmuka pemanggilan sistem atau system call interface normal yang sudah ada. Selain itu, mekanisme isolasi yang diberikan oleh kernel, mampu menyediakan fitur manajemen sumber daya untuk membatasi akibat dari aktivitas salah satu Container terhadap Container yang lainnya (Risdianto dkk, 2021).

2.1.14 Docker

Docker adalah platform terbuka untuk mengembangkan, mengirimkan, dan menjalankan aplikasi. Docker memungkinkan pengguna memisahkan aplikasi dari infrastruktur sehingga Pengguna dapat mengirimkan perangkat lunak dengan cepat.

Dengan Docker, Pengguna dapat mengelola infrastruktur dengan cara yang sama seperti Pengguna mengelola aplikasi. Dengan memanfaatkan metodologi Docker untuk mengirimkan, menguji, dan menerapkan kode dengan cepat, Pengguna dapat secara signifikan mengurangi penundaan antara menulis kode dan menjalankannya dalam produksi.

Docker menyediakan kemampuan untuk mengemas dan menjalankan aplikasi di lingkungan yang terisolasi secara longgar yang disebut wadah. Isolasi dan keamanan memungkinkan Pengguna menjalankan banyak kontainer secara bersamaan di host tertentu. Kontainer ringan dan berisi semua yang diperlukan untuk menjalankan aplikasi, jadi Pengguna tidak perlu bergantung pada apa yang saat ini terinstal di host. Pengguna dapat dengan mudah berbagi wadah saat bekerja, dan pastikan semua orang yang berbagi dengan Pengguna mendapatkan wadah yang sama yang bekerja dengan cara yang sama. Docker menyediakan alat dan platform untuk mengelola siklus hidup kontainer Pengguna: Kembangkan aplikasi Pengguna dan komponen pendukungnya menggunakan wadah. Wadah menjadi unit untuk mendistribusikan dan menguji aplikasi Pengguna. Saat Pengguna siap, terapkan aplikasi Pengguna ke lingkungan produksi, sebagai wadah atau layanan yang diatur. Ini berfungsi sama apakah lingkungan produksi Pengguna adalah pusat data lokal, penyedia cloud, atau gabungan keduanya (Docker, n.d.).

2.1.15 Penetration test

Penetration Testing, atau pentesting, melibatkan simulasi serangan nyata untuk menilai risiko yang terkait dengan potensi pelanggaran keamanan. Pada pentest, penguji tidak hanya menemukan kerentanan yang dapat digunakan oleh penyerang tetapi juga mengeksploitasi kerentanan, untuk menilai apa yang mungkin didapat oleh penyerang setelah eksploitasi sukses. Proses ini mencakup tahapan mulai dari perencanaan, pengintaian, pemindaian, eksploitasi, hingga analisis, pelaporan, dan remediasi. Menggunakan kerangka kerja dan standar seperti OWASP, PTES, dan NIST, serta alat seperti Kali Linux, Metasploit, dan Nmap, penetration testing membantu organisasi meningkatkan keamanan sistem mereka dan mengurangi risiko serangan siber. (Sarker et al., 2023)

2.1.16 Secure Shell Brute Force

SSH *Brute Force* adalah algoritma yang memecahkan masalah dengan pemikiran yang sangat sederhana, langsung, dan dengan cara yang jelas/lempang. Algoritma ini mampu digunakan dalam meretas semua kemungkinan password dengan memasukkan karakter dan panjang password tertentu sehingga dapat menghasilkan banyak kemungkinan kombinasi password. Serangan Brute Force adalah sebuah teknik serangan terhadap sebuah sistem keamanan komputer yang menggunakan percobaan terhadap semua kunci yang mungkin. Secara sederhana, sistem serangan Brute Force adalah sistem menebak password dan mencoba semua kombinasi karakter yang mungkin. Serangan Brute Force digunakan untuk menjebol akses ke suatu host (server/workstation/network) atau kepada data yang ter-enkripsi. Metode ini dipakai para cracker untuk mendapatkan account secara tidak sah, dan sangat berguna untuk memecahkan enkripsi. (Fitra Pradana, 2023)

2.1.17 Port Scanning

Port scanning merupakan metode yang digunakan untuk mendeteksi port yang terbuka serta layanan yang aktif pada suatu sistem jaringan. Teknik ini termasuk dalam tahap awal serangan siber yang dikenal sebagai "reconnaissance" atau pengintaian. Pada tahap ini, pelaku serangan mengumpulkan data mengenai sistem target guna mengidentifikasi celah keamanan yang bisa dimanfaatkan. Informasi yang diperoleh dari proses port scanning membantu penyerang menentukan bagian jaringan yang rentan, yang kemudian dijadikan sasaran untuk mengeksploitasi kelemahan dan memperoleh akses ke system. (Coyle, 2024)

2.1.18 SQL Injection

SQL Injection adalah sebuah teknik hacking yang bertujuan untuk menyusup ke dalam sistem sebuah website untuk mengetahui isi database, ataupun informasi-informasi yang terdapat di situs tersebut. Teknik ini dapat terjadi dikarenakan adanya kode-kode program yang lemah dan adanya proteksi yang kurang aman dan lemah dari pengelola website tersebut atau yang disebut administrator. SQL Injection merupakan salah satu teknik hacking yang hanya memerlukan port 80 meskipun sering dilakukan patch pada server. Kode SQL tersebut diinjeksikan pada query SQL sebagai input melalui halaman website.

Hal ini memungkinkan intruder untuk memasuki halaman web administrator tanpa melakukan scan port yang terbuka, tidak terdeteksi oleh firewall dan bahkan tidak menggunakan tool apapun. SQL Injection sendiri diperkirakan muncul atau telah ada semenjak database SQL dikoneksikan pertama kali dengan aplikasi web dan mulai menjadi perhatian publik semenjak tahun 1998. Semenjak itu para peneliti mulai membangun dan mengembangkan teknik untuk mengeksploitasi SQL injection. (Elu, 2013)

Penelitian Terdahulu Penelitian "Perancangan dan Implementasi Security Information and Event Management (SIEM) pada Layanan Virtual Server" oleh (Heluka dkk., 2023). Penelitian ini membahas masalah di mana administrator kadang tidak sadar akan serangan karena banyak perangkat endpoint yang memungkinkan serangan melewati firewall dan menyerang server. Hal ini dapat menyebabkan insiden serius, dan analisis log bisa saja tidak cukup untuk mendeteksi serangan yang lebih kompleks. Tujuan penelitian ini adalah membuktikan bahwa SIEM dengan nama Wazuh dapat melakukan pemantauan server secara terpusat dengan efektif pada perangkat. Metode yang digunakan adalah penerapan SIEM. Hasil penelitian menunjukkan bahwa layanan Virtual Server berhasil menggunakan Wazuh dalam memonitor log secara efektif, dengan kolaborasi aktif tim keamanan dalam mengatur aturan SIEM yang berperan penting. Penelitian mendatang dapat fokus pada integrasi SIEM dengan perangkat jaringan seperti firewall untuk memperbaiki tindakan keamanan. Namun, pengecekan IP menggunakan abuseIPdb masih dilakukan secara manual.

Penelitian "Implementasi dan Analisis Profil Sistem pada Virtualisasi Fortigate Firewall Berdasarkan Metrik Sumber Daya Komputasi" oleh (Ibrahim dkk., 2023). Penelitian ini bertujuan untuk menganalisis kinerja firewall Fortigate dalam lingkungan virtualisasi terhadap serangan DDoS. Metode yang digunakan adalah pengujian beban dengan dua skenario, yaitu service HTTP allow dan service HTTP block, untuk mengukur konsumsi sumber daya CPU, memori, dan sesi pada firewall. Hasil penelitian menunjukkan bahwa konsumsi CPU dapat mencapai 97,1% dan memori 83,7% saat terjadi serangan. Walaupun firewall Fortigate mampu mempertahankan layanan saat diserang, keterbatasan spesifikasi virtualisasi (maksimal 2 GB memory dan 1 core CPU) menjadi hambatan dalam

mensimulasikan kondisi jaringan besar. Selain itu, penelitian ini hanya berfokus pada satu jenis serangan DDoS tanpa menguji.

Penelitian "Implementasi Security Information and Event Management (SIEM) untuk Deteksi dan Analisa Insiden Keamanan pada Web Server" oleh (Hadi dkk., 2023). Penelitian ini bertujuan untuk membangun sistem deteksi insiden keamanan menggunakan kombinasi Wazuh dan Suricata pada web server. Metode yang digunakan adalah eksperimen dengan skenario serangan nyata, seperti SQL Injection, XSS, Directory Traversal, dan DoS. Hasil penelitian menunjukkan bahwa sistem SIEM berhasil mendeteksi serangan secara real-time dan mengirimkan notifikasi melalui Telegram. Penelitian ini melanjutkan dan mengembangkan penelitian sebelumnya dengan fokus pada aplikasi web server dan integrasi sistem notifikasi cepat untuk meningkatkan respon terhadap insiden.

Penelitian "Implementasi Keamanan Jaringan Komputer Menggunakan Fortigate Sebagai Firewall pada Lab Komputer IAIN Bengkulu" oleh (Jaya dkk, 2021). Penelitian ini bertujuan mengamankan server dari serangan ICMP Flooding dan Port Scanning menggunakan Fortigate. Metode yang digunakan adalah Network Development Life Cycle (NDLC) melalui tahapan analisis, perancangan, implementasi, dan monitoring. Hasil penelitian menunjukkan bahwa Fortigate efektif dalam mendeteksi dan memblokir serangan, menjaga kestabilan server dari potensi kerusakan akibat serangan tersebut. Penelitian ini terbatas pada dua jenis serangan, sehingga disarankan untuk mengembangkan pengujian terhadap ancaman lainnya di masa mendatang.

Penelitian "Implementasi Security Information and Event Management (SIEM) Menggunakan Wazuh pada Politeknik Caltex Riau" oleh (Zahira, 2023). Penelitian ini membahas kesulitan dalam menganalisis log dari berbagai endpoint seperti server yang memakan waktu lama dan memerlukan akses server satu per satu. Tujuan penelitian ini adalah untuk mempermudah pengumpulan log dari banyak endpoint, sehingga menghemat waktu dan memudahkan analisis log yang dapat mencegah insiden serangan siber. Metode yang digunakan adalah penerapan SIEM. Hasil penelitian menunjukkan bahwa Wazuh efektif dalam memantau server sebagai HIDS dengan performa stabil pada serangan seperti

BruteForce dan SQL Injection. Namun, terjadi penurunan kinerja pada serangan DoS Attack dengan SYN Flood. Secara keseluruhan, Wazuh menjadi solusi efektif dalam mendeteksi serangan dan menyajikan data serangan dalam log bagi administrator. Namun, Wazuh tidak dapat secara otomatis menanggapi serangan keamanan dan memerlukan integrasi dengan sistem IDS lain untuk menanggapi serangan tersebut.

Sebagai tinjauan lebih mendalam terhadap perkembangan terkini (*state of the art*) pada bidang integrasi SIEM dan SOAR, dua penelitian berikut dikaji secara khusus karena arsitekturnya paling mendekati sistem yang dibangun pada penelitian ini, yaitu kombinasi Wazuh sebagai SIEM dan Shuffle sebagai SOAR.

Sagita (2024), dalam penelitian berjudul "*Implementasi Security Orchestration, Automation, and Response dengan Wazuh & Shuffle untuk Otomasi Respon SOC*", mengintegrasikan Wazuh, Suricata, Shuffle, dan Iris untuk meningkatkan kinerja tim *Security Operations Center* (SOC) di Departemen Teknologi Informasi Institut Teknologi Sepuluh Nopember. Raspberry Pi digunakan sebagai perangkat eksekutor pemblokiran, dengan pengujian mencakup serangan berbasis jaringan (*reconnaissance, bruteforce, denial of service, DHCP starvation*) maupun serangan aplikasi web (*SQL injection, cross-site scripting*). Sistem yang dibangun terbukti mampu mendeteksi *reconnaissance* dalam waktu sekitar 35 detik, *bruteforce* dalam 10 detik, serta memblokir sebagian besar serangan terdeteksi dalam waktu kurang dari 40 detik. Meskipun demikian, penggunaan Raspberry Pi sebagai eksekutor membatasi kapasitas pemrosesan dan skalabilitas sistem dibandingkan perangkat firewall enterprise, serta belum dilaporkan adanya pengujian terhadap tingkat *false positive* dari mekanisme respons otomatis tersebut.

Sementara itu, Ridwan & Yudi Kurniawan (2025), melalui penelitian "*Implementasi dan Evaluasi Sistem Keamanan Siber Berbasis Wazuh, Shuffle, dan Yara di Pusat Data Diskominfo Kota Tangerang dengan Metode PPDIOO*", mengintegrasikan Wazuh, Shuffle, dan YARA untuk memantau serta merespons ancaman secara *real-time* pada lebih dari seratus server di pusat data pemerintah. YARA ditambahkan sebagai mekanisme deteksi *malware* berbasis *signature* yang

terintegrasi dengan Wazuh, sehingga berkas mencurigakan dapat dideteksi dan dikarantina secara otomatis, dengan metode PPDIOO (*Prepare, Plan, Design, Implement, Operate, Optimize*) sebagai kerangka kerja implementasi. Namun, penelitian ini tidak menyertakan perangkat firewall sebagai eksekutor tindakan pemblokiran berbasis jaringan, serta tidak melaporkan pengujian terhadap pola serangan jaringan seperti *port scanning* atau *SSH brute force*.

Kedua penelitian tersebut menunjukkan bahwa integrasi SIEM dan SOAR mampu mendukung otomatisasi deteksi dan respons insiden secara efektif, namun perangkat eksekutor yang digunakan (Raspberry Pi maupun tanpa eksekutor jaringan sama sekali) belum merepresentasikan kapabilitas firewall enterprise, dan evaluasi *false positive* belum menjadi fokus pengujian pada kedua aspek yang menjadi dasar pengembangan pada penelitian ini.

Tabel 2. 1 Penelitian Terdahulu

Peneliti	Judul	Tools	Hasil
(Heluka dkk., 2023)	Perancangan dan Implementasi Security Information and Event Management (SIEM) pada Layanan Virtual Server	SIEM	Penelitian menunjukkan bahwa layanan Virtual Server berhasil menggunakan Wazuh dalam memonitor log secara efektif, dengan kolaborasi aktif tim keamanan dalam mengatur aturan SIEM yang berperan penting
(Ibrahim dkk., 2023)	Implementasi dan Analisis Profil Sistem pada Virtualisasi Fortigate Firewall Berdasarkan Metrik Sumber Daya Komputasi	Fortigate	Fortigate mampu mempertahankan layanan saat terjadi serangan DDoS, meskipun konsumsi CPU mencapai 97,1% dan memori 83,7%.
(Hadi dkk., 2023)	Implementasi Security Information and Event Management (SIEM) untuk Deteksi dan Analisa Insiden	SIEM	Sistem SIEM yang dibangun dengan Wazuh dan Suricata berhasil mendeteksi serangan pada web server secara real-time dan mengirimkan notifikasi melalui Telegram, sehingga

Peneliti	Judul	Tools	Hasil
	Keamanan pada Web Server		meningkatkan kecepatan respon terhadap insiden keamanan.
(Jaya dkk, 2021)	Implementasi Keamanan Jaringan Komputer Menggunakan Fortigate Sebagai Firewall pada Lab Komputer IAIN Bengkulu	Fortigate	Fortigate efektif dalam mendeteksi dan memblokir serangan ICMP Flooding dan Port Scanning, serta mampu menjaga kestabilan server, meskipun pengujian masih terbatas pada dua jenis serangan.
(Zahira, 2023)	Implementasi Security Information and Event Management (SIEM) Menggunakan Wazuh pada Politeknik Caltex Riau	SIEM	Pempermudah Pengumpulan log dari banyak endpoint sehingga menghemat waktu dan memudahkan menganalisis log yang dapat mencegah insiden serangan siber.
(Sagita, 2024)	Implementasi <i>Security Orchestration, Automation, and Response</i> dengan Wazuh & Shuffle untuk Otomasi Respon SOC	SIEM, SOAR	Sistem berhasil mendeteksi <i>reconnaissance</i> dalam ± 35 detik, <i>bruteforce</i> dalam ± 10 detik, serta memblokir sebagian besar serangan dalam waktu kurang dari 40 detik menggunakan Raspberry Pi sebagai eksekutor
(Ridwan & Kurniawan, 2025)	Implementasi dan Evaluasi Sistem Keamanan Siber Berbasis Wazuh, Shuffle, dan Yara di Pusat Data Diskominfo Kota Tangerang	SIEM, SOAR, YARA	Wazuh berhasil memantau anomali <i>server</i> secara <i>real-time</i> dan diteruskan ke Shuffle; YARA berhasil mendeteksi dan mengkarantina <i>file</i> mencurigakan secara otomatis

Dibandingkan dengan kedua penelitian terkini tersebut, penelitian ini memiliki perbedaan mendasar pada penggunaan *firewall enterprise* (FortiGate) sebagai eksekutor kebijakan keamanan berbasis *address group*, alih-alih Raspberry Pi (Sagita, 2024) atau tanpa eksekutor jaringan sama sekali (Ridwan & Kurniawan, 2025). Selain itu, kedua penelitian tersebut belum melaporkan adanya pengujian

terhadap potensi *false positive* pada mekanisme respons otomatis yang dibangun aspek yang menjadi salah satu fokus evaluasi pada penelitian ini.

Penelitian ini memiliki keunggulan dari sisi integrasi dan otomasi. Proyek akhir ini tidak hanya mengimplementasikan Wazuh sebagai platform SIEM, tetapi juga mengintegrasikannya dengan Shuffle sebagai platform SOAR untuk mengotomatiskan respon terhadap ancaman yang terdeteksi. Respon tersebut kemudian dieksekusi secara langsung oleh perangkat FortiGate, sehingga membentuk alur deteksi hingga pemblokiran ancaman yang berjalan otomatis dan *real-time*. Dengan pendekatan ini, penelitian ini diharapkan mampu memberikan kontribusi nyata dalam membangun sistem keamanan jaringan yang proaktif, efisien, dan siap diterapkan pada skala yang lebih luas.