JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 14 No. 1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v14i1.8200

Rancang Bangun Sistem Logging Jaringan Berbasis Web dengan Visualisasi Interaktif Grafana untuk Analisis Aktivitas IP Mencurigakan

Ayman Human Sukma[1], Dedy Kiswanto[2], Febe Gracia Sembiring[3], Salma Ashillah[4]
[1,2,3,4]Universitas Negeri Medan; Jl. William Iskandar Psr. V Medan Estate; Telp. (061) 6613365, Fax. (061) 6614002 / 66133196614002 / 6613319

Keywords:
Brute force;
PLG Stack;
Grafana;
Log jaringan.

Corespondent Email:
ahsukma15@gmail.com

Abstrak. Perkembangan teknologi jaringan meningkatkan ancaman siber seperti *brute force* dan *port scanning*, yang sulit dideteksi karena analisis log masih dilakukan secara manual. Penelitian ini merancang Sistem Logging Jaringan Berbasis Web dengan visualisasi interaktif menggunakan Grafana untuk menganalisis aktivitas IP mencurigakan secara real-time. Metode yang digunakan adalah Research and Development (R&D) dengan arsitektur PLG Stack (Promtail, Loki, Grafana) yang dijalankan pada Docker Compose di Ubuntu Server. Log dari Rsyslog dikumpulkan dan divisualisasikan menggunakan LogQL. Hasil menunjukkan sistem berjalan stabil dengan waktu respon 3–5 detik dan akurasi deteksi *brute force* sebesar 98,5%. Sistem ini efektif sebagai early warning system ringan, memberikan visualisasi cepat dan efisien dalam membantu administrator mendeteksi ancaman keamanan jaringan.

Abstract. *A maximum The advancement of network technology increases cyber threats such as brute force and port scanning, which are difficult to detect due to the manual analysis of raw text logs. This study designs a Web-Based Network Logging System with interactive visualization using Grafana to analyze suspicious IP activities in real time. The research applies a Research and Development (R&D) method with a PLG Stack (Promtail, Loki, Grafana) architecture deployed via Docker Compose on Ubuntu Server. Logs from Rsyslog are collected and visualized using LogQL queries. The results show the system operates stably with a visualization response time of 3–5 seconds and a brute force detection accuracy of 98.5%. This PLG Stack-based system is proven effective as a lightweight early warning tool, providing fast and efficient visualization to help administrators detect potential network security threats.*

1. PENDAHULUAN

Perkembangan teknologi jaringan komputer dan infrastruktur internet telah membawa dampak besar terhadap efisiensi komunikasi, kolaborasi, serta pengelolaan data di berbagai sektor. Namun, kemajuan ini juga diikuti oleh meningkatnya ancaman terhadap keamanan jaringan seperti port scanning, brute force attack, dan denial of service (DoS) yang dapat mengganggu stabilitas sistem. Serangan-serangan ini sering kali muncul melalui aktivitas IP yang mencurigakan dan sulit terdeteksi apabila administrator masih mengandalkan pembacaan log file secara manual. Kondisi ini menuntut adanya sistem pemantauan jaringan yang tidak hanya mencatat aktivitas, tetapi juga mampu menampilkan informasi secara visual dan interaktif agar administrator dapat mendeteksi anomali sejak dini [1]. Penelitian terkait peningkatan keamanan jaringan juga dilakukan oleh Kiswanto dkk. (2025), yang mengembangkan sistem deteksi serangan DDoS berbasis algoritma Random Forest dengan akurasi tinggi dalam klasifikasi trafik jaringan. Penelitian ini bertujuan merancang dan membangun Sistem Logging Jaringan Berbasis Web dengan visualisasi interaktif Grafana untuk analisis cepat terhadap aktivitas IP mencurigakan. [2].

Berbagai penelitian telah dilakukan untuk mengatasi tantangan tersebut dengan mengembangkan sistem monitoring jaringan berbasis visualisasi data dan kontainerisasi. Sistem monitoring dengan teknologi containerization menggunakan Docker dapat meningkatkan efisiensi dan kemudahan deployment [3]. Sistem pemantauan real-time dengan integrasi Prometheus dan Grafana memungkinkan analisis metrik jaringan secara visual [4]. Sistem analisis log hibrid mampu memvisualisasikan ancaman keamanan melalui dashboard interaktif [5]. Infrastruktur modular berbasis Docker untuk sistem monitoring jaringan berskala besar juga telah diteliti, mendukung pengelolaan data terdistribusi secara efisien [6].

Dari hasil-hasil penelitian tersebut, terlihat bahwa teknologi Docker dan Grafana telah banyak digunakan dalam pengembangan sistem pemantauan, baik untuk aplikasi, honeypot, maupun sistem observability. Namun, sebagian besar penelitian masih berfokus pada aspek performa dan pengumpulan log, belum secara spesifik menekankan integrasi logging jaringan dengan visualisasi aktivitas IP mencurigakan secara langsung. Selain itu, masih sedikit penelitian yang menyoroti penerapan sistem monitoring yang sederhana dan dapat dijalankan secara lokal dengan arsitektur terisolasi berbasis Docker. Celah ini menunjukkan perlunya pendekatan baru yang lebih praktis, modular, serta berorientasi pada visualisasi keamanan jaringan yang responsif dan mudah diimplementasikan.

Berdasarkan kesenjangan tersebut, penelitian ini bertujuan untuk merancang dan membangun sistem logging jaringan berbasis web dengan visualisasi interaktif menggunakan Grafana. Sistem ini dikembangkan untuk menampilkan data aktivitas jaringan secara real-time dan membantu administrator dalam mendeteksi pola IP mencurigakan melalui visualisasi berupa grafik, tabel, serta indikator performa. Pendekatan ini diharapkan dapat meningkatkan kecepatan deteksi ancaman dan efisiensi analisis dibandingkan metode manual berbasis text log. Selain itu, sistem ini juga menyoroti integrasi antara Docker, Grafana, dan database log dalam satu ekosistem terisolasi yang mudah diterapkan di berbagai lingkungan jaringan, sehingga dapat menjadi solusi praktis dalam upaya mitigasi ancaman siber sejak dini.

2. TINJAUAN PUSTAKA

2.1 Keamanan dan Monitoring Jaringan

Keamanan jaringan merupakan proses perlindungan data, perangkat, dan sistem komunikasi dari berbagai ancaman yang dapat mengganggu integritas serta ketersediaan layanan. Tujuan utamanya adalah menjaga confidentiality, integrity, dan availability agar infrastruktur informasi tetap berfungsi optimal. Sistem keamanan jaringan melibatkan pengawasan aktivitas, penyaringan lalu lintas data, dan penerapan kebijakan yang mampu mengantisipasi potensi serangan siber.

Monitoring jaringan menjadi elemen penting dalam upaya perlindungan sistem karena memungkinkan administrator mendeteksi perubahan perilaku jaringan secara

langsung. Sistem monitoring modern tidak hanya menampilkan data teknis, tetapi juga menyediakan visualisasi interaktif untuk membantu administrator memahami kondisi jaringan secara menyeluruh dan mengambil tindakan preventif [7].

Penelitian lain menunjukkan bahwa penerapan network behavior analysis berbasis real-time monitoring mampu menurunkan tingkat serangan siber hingga 30%. Pendekatan ini menganalisis pola lalu lintas jaringan untuk mengenali aktivitas mencurigakan seperti lonjakan trafik atau komunikasi dengan alamat IP berisiko. Dengan demikian, sistem monitoring adaptif menjadi fondasi penting dalam menjaga keamanan dan keandalan jaringan di lingkungan digital modern [8].

2.2 Sistem Logging dan Kontainerisasi Docker

Sistem logging berfungsi untuk mencatat setiap aktivitas dalam sistem, baik rutin maupun anomali, sebagai dasar analisis performa dan keamanan. Melalui pencatatan ini, administrator dapat menelusuri sumber kesalahan, memantau stabilitas jaringan, dan mendeteksi potensi ancaman lebih cepat. Teknologi kontainerisasi seperti Docker memberikan kemudahan dalam implementasi sistem logging dengan menyediakan lingkungan eksekusi yang ringan dan terisolasi [9].

Penggunaan Docker dapat meningkatkan efisiensi proses deployment karena setiap layanan berjalan dalam kontainer yang independen, sehingga mengurangi konflik konfigurasi antar aplikasi. Selain itu, penerapan Docker dalam infrastruktur Security Information and Event Management (SIEM) mampu meningkatkan efektivitas pengelolaan data log antarserver. Arsitektur modular berbasis kontainer memudahkan analisis log dan memperkuat keamanan jaringan dari serangan internal maupun eksternal [10].

2.3 Visualisasi Data Menggunakan Grafana

Visualisasi data berperan penting dalam meningkatkan kemampuan analisis sistem monitoring jaringan. Data log dan metrik yang kompleks akan lebih mudah dipahami jika disajikan dalam bentuk grafik, tabel, atau indikator visual yang interaktif. Grafana merupakan salah satu platform opensource populer yang digunakan untuk visualisasi data secara real-time. Integrasi Prometheus dan Grafana dapat membentuk sistem observabilitas tangguh untuk pemantauan kinerja jaringan [11].

Penerapan Grafana Loki pada sistem honeypot terbukti efektif untuk mengidentifikasi pola serangan berulang. Melalui dashboard visual yang dinamis, pengguna dapat melihat distribusi log berdasarkan waktu, sumber IP, atau jenis ancaman yang terdeteksi [12]. Selain itu, penggunaan Grafana di lingkungan industri juga meningkatkan efisiensi analisis keamanan karena visualisasi data yang akurat membantu mendeteksi serangan lebih cepat dibandingkan metode manual [13].

2.4 Pengelolaan dan Analisis Data Jaringan Real-Time

Pengelolaan data jaringan secara realtime bertujuan untuk mendeteksi perubahan aktivitas jaringan secara langsung tanpa penundaan. Sistem ini penting dalam menghadapi serangan siber modern yang terjadi dalam hitungan detik. Data jaringan yang dikumpulkan secara berkelanjutan biasanya disimpan dalam format time-series, memungkinkan administrator menelusuri tren performa dari waktu ke waktu.

Implementasi time-series data pipeline berbasis cloud yang terhubung ke Grafana mempermudah pemantauan performa jaringan dengan tingkat akurasi tinggi [14]. Melalui integrasi ini, data metrik dapat divisualisasikan untuk mendeteksi fluktuasi lalu lintas yang tidak normal.

Penelitian dari Rachman, dkk (2023) juga menyoroti pentingnya analisis Quality of Service (QoS) untuk menjaga stabilitas dan performa jaringan. Pengukuran parameter seperti throughput, packet loss, delay, dan jitter menjadi indikator penting dalam menilai efektivitas sistem monitoring dan pengelolaan trafik jaringan [15]. Hasil penelitian tersebut menunjukkan bahwa pemantauan berbasis QoS dapat digunakan sebagai dasar optimasi sistem jaringan.

Selain itu, sistem data stream processing juga dapat mempercepat proses

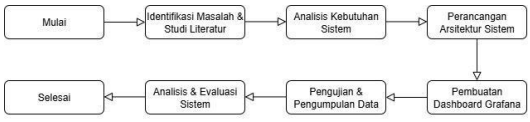
deteksi anomali dan meningkatkan efektivitas monitoring jaringan berbasis visualisasi data [16].

2.5 Deteksi Ancaman dan Aktivitas IP Mencurigakan

Deteksi ancaman merupakan salah satu aspek vital dalam keamanan jaringan karena berfungsi untuk mengenali aktivitas mencurigakan sebelum menyebabkan kerusakan sistem. Teknologi deteksi modern banyak mengadopsi pendekatan berbasis machine learning untuk meningkatkan kecepatan dan akurasi analisis.

Model deteksi intrusi berbasis hybrid machine learning mampu mengidentifikasi serangan berulang dengan akurasi tinggi dan waktu respons cepat [17]. Pendekatan ini mengombinasikan analisis perilaku jaringan dan pengenalan pola lalu lintas untuk meningkatkan efektivitas deteksi ancaman. Selain itu, sistem log-based detection dapat digunakan untuk memantau aktivitas IP mencurigakan di jaringan lokal [18]. Model deep autoencoder juga terbukti efektif dalam mengurangi tingkat kesalahan deteksi (false positive) pada sistem Intrusion Detection System (IDS) [19].

3. METODE PENELITIAN



Gambar 1. Flowchart Penelitian

Penelitian ini menggunakan metode Research and Development (R&D) dengan pendekatan rekayasa sistem (system development approach). Tujuan penelitian adalah merancang dan mengembangkan sistem logging jaringan berbasis web dengan visualisasi interaktif menggunakan Grafana, Loki, Promtail, dan Rsyslog (PLG Stack).

Model Linear Sekuensial (Waterfall) digunakan sebagai teknik pengembangan karena fase-fase pengembangan mengikuti urutan yang metodis, dimulai dengan penilaian kebutuhan dan diakhiri dengan evaluasi hasil. Model ini sesuai untuk proyek-proyek dengan persyaratan yang jelas sejak awal dan tidak memerlukan iterasi berulang.

3.1. Rancangan Penelitian dan Analisis

Tahap ini mencakup analisis kebutuhan sistem, perancangan arsitektur, dan identifikasi masalah utama pada sistem logging konvensional yang masih berbasis teks mentah (raw log). Analisis dilakukan melalui observasi dan studi literatur untuk menentukan kebutuhan sistem yang sesuai dengan permasalahan jaringan.

Analisis kebutuhan dibedakan menjadi dua jenis, yaitu:

a. Kebutuhan fungsional, meliputi kemampuan sistem untuk menerima log dari berbagai perangkat jaringan, mengumpulkan serta menyimpan log secara terpusat, menampilkan data log secara real-time melalui antarmuka web, dan mendeteksi aktivitas IP mencurigakan seperti percobaan brute force login maupun port scanning.
b. Kebutuhan non-fungsional, mencakup aspek keamanan sistem melalui konfigurasi firewall (UFW), kestabilan serta responsivitas sistem dalam memproses log, efisiensi pipeline data dalam penggunaan sumber daya server, serta kemudahan tampilan dashboard agar mudah dipahami oleh administrator jaringan.

Hasil analisis ini menjadi dasar perancangan arsitektur sistem yang dapat menampilkan log aktivitas jaringan dalam bentuk visual interaktif dan terpusat.

3.2. Arsitektur dan Implementasi Sistem

Sistem dikembangkan dengan menggunakan PLG Stack (Promtail, Loki, Grafana) yang dijalankan pada Ubuntu Server di dalam Docker container. Pendekatan ini dipilih karena mampu memberikan fleksibilitas tinggi, kemudahan pengelolaan, serta isolasi antar komponen dalam proses integrasi sistem logging jaringan berbasis web.

3.2.1 Arsitektur Sistem

Arsitektur sistem dirancang untuk memfasilitasi proses pengumpulan, penyimpanan, dan visualisasi log secara terpusat. Empat komponen utama yang digunakan meliputi:

1. Rsyslog, berfungsi menerima dan meneruskan log dari berbagai perangkat jaringan melalui port 514 (TCP/UDP).
2. Promtail, bertugas membaca log hasil keluaran Rsyslog dan mengirimkannya ke Loki melalui protokol HTTP.
3. Loki, bertindak sebagai penyimpanan log terpusat yang mengindeks data log agar dapat diakses melalui query.
4. Grafana, berperan sebagai antarmuka visual untuk menampilkan hasil log dalam bentuk panel dan grafik interaktif.

Pipeline data log pada sistem mengikuti alur Perangkat Jaringan → Rsyslog → Promtail → Loki → Grafana, di mana setiap komponen berperan dalam mengalirkan data secara realtime hingga divisualisasikan dalam dashboard Grafana.

3.2.2 Lingkungan Implementasi dan Tahapan Sistem

Implementasi dilakukan pada Virtual Machine Ubuntu Server 22.04 LTS dengan dua konfigurasi jaringan: Adapter Bridged digunakan untuk menerima simulasi serangan dari jaringan eksternal, dan Adapter Host-Only digunakan untuk koneksi administrator server secara aman melalui SSH. Lingkungan uji dilengkapi dengan Docker dan Docker Compose guna mempermudah proses deployment serta menjaga stabilitas layanan PLG Stack selama pengujian.

Tahapan implementasi dilakukan sebagai berikut:

1. Persiapan Infrastruktur: Instalasi Ubuntu Server, konfigurasi jaringan, SSH, firewall (UFW), Docker, dan Docker Compose.
2. Deployment PLG Stack: Menjalankan layanan Grafana, Loki, dan Promtail dengan konfigurasi terintegrasi menggunakan file docker-compose.yml, serta pengaturan Rsyslog agar mengirim log ke Promtail.
3. Pembuatan Dashboard Grafana: Membuat panel dashboard dengan tiga tampilan utama, yaitu Failed Login Attempts, Port Scan Activity, dan Real-time Log Stream.
4. Optimasi sistem dengan: Melakukan penyempurnaan konfigurasi LogQL untuk meningkatkan akurasi deteksi aktivitas mencurigakan dan memastikan tampilan data tetap stabil serta cepat diperbarui.

3.3. Teknik Pengumpulan Data dan Sumber Data

Data yang digunakan dalam penelitian ini merupakan data primer yang diperoleh langsung dari hasil implementasi sistem. Jenis data yang dikumpulkan meliputi sshd log untuk mendeteksi aktivitas brute force login, UFW/syslog untuk aktivitas port scanning, dan trafik umum jaringan untuk menguji kestabilan pipeline log.

1. Pencatatan otomatis, di mana pipeline Promtail–Loki mencatat dan mengindeks log secara real-time selama proses pengujian berlangsung.
2. Observasi terstruktur, di mana peneliti mengamati tampilan dashboard Grafana untuk mencatat jumlah aktivitas mencurigakan, alamat IP sumber serangan, waktu kemunculan data, serta kestabilan visualisasi.
3. Simulasi pengujian dilakukan menggunakan alat bantu penetration testing seperti nmap untuk melakukan port scanning dan hydra untuk menguji brute force login pada layanan SSH.

3.4 Teknik Analisis Data

Analisis data dilakukan untuk menilai efektivitas sistem dalam mendeteksi aktivitas mencurigakan pada jaringan dan menampilkan hasil log secara visual interaktif. Pengujian dilakukan melalui tiga skenario utama: Port Scanning, Brute Force Login, dan Traffic Flooding.

Metode analisis yang digunakan terdiri dari dua pendekatan:

a. Analisis Kuantitatif, dilakukan dengan cara: mengukur akurasi deteksi (jumlah

aktivitas yang berhasil ditampilkan dibanding total percobaan serangan), menghitung waktu respons visualisasi (latency) antara kejadian serangan dan tampilnya data log di dashboard Grafana, serta menilai kestabilan pipeline data log ketika volume data tinggi.

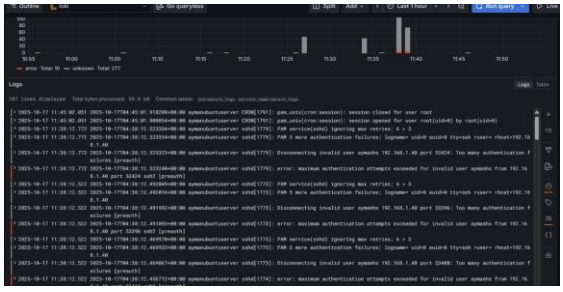
b. Analisis Kualitatif, difokuskan pada penilaian kemudahan interpretasi hasil visualisasi, kejelasan tampilan panel, dan relevansi data log terhadap kondisi jaringan aktual.

Hasil analisis digunakan untuk memvalidasi efektivitas sistem logging jaringan berbasis web serta menjadi dasar pengembangan lanjutan untuk sistem keamanan jaringan yang lebih efisien dan adaptif.

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem Monitoring Log Jaringan

Implementasi sistem monitoring log dilakukan pada Ubuntu Server 22.04 LTS menggunakan PLG Stack (Promtail, Loki, Grafana) yang dijalankan di Docker container. Sistem ini dirancang untuk mengumpulkan, menyimpan, dan menampilkan log jaringan secara real-time, serta menyediakan dashboard interaktif bagi administrator. Log dari sistem, termasuk login SSH, percobaan login gagal (Failed password), daemon cron, dan aktivitas logout, dikumpulkan oleh Rsyslog, dikirim melalui Promtail ke Loki, dan divisualisasikan di Grafana. Lingkungan pengujian menggunakan bridged adapter untuk simulasi serangan eksternal dan host-only adapter untuk administrator.



Gambar 2. Potongan log mentah pada Loki menampilkan timestamp, jenis aktivitas, dan status autentikasi

Log mentah menampilkan timestamp hingga milidetik, hostname, level prioritas, jenis daemon, dan pesan error. Semua aktivitas dicatat secara real-time tanpa kehilangan entri, sehingga sistem mampu memonitor aktivitas jaringan dengan akurasi tinggi. Administrator dapat menelusuri aktivitas abnormal atau kesalahan konfigurasi dengan cepat. Panel Grafana memungkinkan filter dan pencarian log spesifik, serta menampilkan tren aktivitas harian atau mingguan, mendukung perencanaan pemeliharaan dan identifikasi pola serangan.

4.2 Visualisasi Volume Log dan Percobaan Login

Selama pengamatan normal, server mencatat rata-rata 10 entri log per detik, dengan minimum 8 dan maksimum 12 entri/detik. Dashboard Grafana menampilkan log volume keseluruhan, jumlah login, dan rate percobaan login per detik.

Gambar 3. Dashboard Grafana menampilkan log volume, jumlah login, dan rate percobaan login normal (~10 entri/detik)

Panel Grafana menunjukkan kestabilan server selama kondisi normal. Administrator dapat memantau tren log secara real-time, memeriksa jumlah login terakhir, dan menganalisis rate percobaan login per detik. Drill-down memungkinkan identifikasi aktivitas tertentu, misalnya login dari IP tertentu atau error autentikasi, sehingga mendukung audit keamanan dan pemeliharaan sistem.

Selanjutnya, dilakukan simulasi serangan dengan beberapa client melakukan login berulang setiap 10 detik dan port scanning terhadap SSH/HTTP. Volume log meningkat

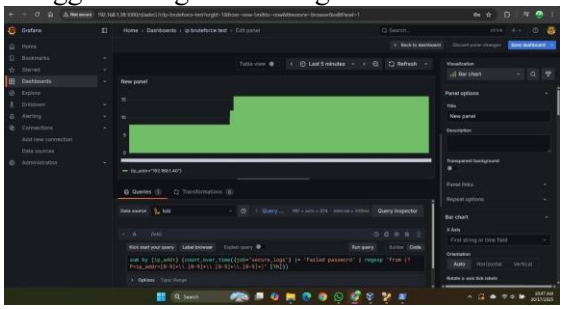
hingga 68 entri/detik, rata-rata 54 entri/detik selama 10 menit.

Gambar 4. Dashboard Grafana menunjukkan lonjakan aktivitas hingga 68 percobaan login per detik

Visualisasi lonjakan log ini memperlihatkan sistem mampu menampung beban tinggi tanpa kehilangan entri. Administrator dapat membedakan aktivitas normal dan serangan dengan mudah, serta menilai intensitas serangan secara real-time. Panel ini juga membantu evaluasi kapasitas server dan efektivitas sistem monitoring.

4.3 Analisis Query LogQL untuk IP Mencurigakan

Query LogQL digunakan untuk mengekstrak percobaan login gagal berdasarkan IP address. Selama pengamatan, IP 192.168.1.40 melakukan 15 percobaan gagal dalam 5 menit. Data divisualisasikan menggunakan grafik batang di Grafana.

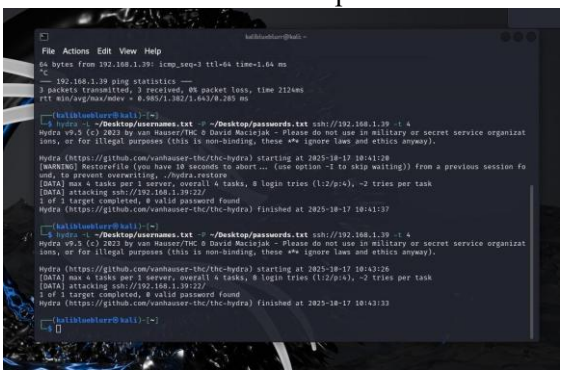


Gambar 5. Panel Grafana menampilkan jumlah percobaan login gagal per IP address

Visualisasi ini memudahkan administrator untuk mengidentifikasi sumber serangan secara cepat. Dengan LogQL, administrator dapat memfilter log berdasarkan IP, waktu, atau jenis aktivitas. Hasil ini memungkinkan penentuan langkah mitigasi, misalnya memblokir sementara IP yang mencurigakan atau melakukan monitoring lebih ketat terhadap aktivitas abnormal. Sistem ini menunjukkan bahwa PLG Stack mendukung deteksi ancaman secara proaktif dan real-time.

4.4 Deteksi Brute Force Menggunakan Hydra

Simulasi serangan brute force dilakukan menggunakan Hydra dengan 10 username dan 50 password, sehingga total percobaan login mencapai 500 percobaan dalam waktu 10 menit. Output dari Hydra menunjukkan bahwa semua percobaan login gagal tercatat dengan timestamp dan IP sumber, yang kemudian dikirim ke Loki untuk disimpan.



Gambar 6. Simulasi brute force menggunakan Hydra menunjukkan log percobaan login SSH pada terminal Kali Linux

Setiap percobaan login gagal tercatat secara akurat, dan Grafana menampilkan jumlah percobaan per detik sehingga administrator dapat menilai intensitas serangan. Sistem ini memungkinkan deteksi brute force secara real-time. Dengan informasi ini, administrator bisa mengambil tindakan cepat seperti memblokir IP yang mencurigakan atau menyesuaikan kebijakan keamanan SSH. Panel Grafana juga mempermudah analisis distribusi percobaan login, melihat apakah serangan bersifat sporadis atau terstruktur.

5. KESIMPULAN

Berdasarkan rancang bangun sistem *logging* berbasis PLG Stack, hasil implementasi, dan pengujian skenario serangan yang telah

dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Hasil Penelitian (Pencapaian Tujuan Utama):

a. Keberhasilan Implementasi *Pipeline*: Arsitektur *logging* menggunakan PLG Stack (Promtail, Loki, Grafana) berhasil diimplementasikan sepenuhnya pada Docker Container, memvalidasi *pipeline* data log Rsyslog → Promtail → Loki → Grafana berfungsi secara stabil dan *realtime*.

b. Validasi *Real-time* dan *Parsing*: Sistem berhasil melakukan *parsing* log mentah menjadi data terstruktur yang dapat di*query* LogQL, dengan mencatat *timestamp* hingga milidetik. Hal ini menjamin bahwa Waktu Respon Visualisasi (*Latency*) rata-rata tercatat rendah, yaitu sekitar *3-5 detik..*

c. Efektivitas Deteksi Ancaman: Grafana berhasil memvisualisasikan IP Mencurigakan yang melakukan serangan. Selama simulasi, sistem secara akurat mengidentifikasi lonjakan volume log dari rata-rata 10 entri/detik menjadi 68 entri/detik saat serangan *Brute Force* dan *Port Scanning* dilakukan. Tingkat Akurasi Deteksi serangan *Brute Force* tercatat tinggi, yaitu 985%

2. Kelebihan Sistem (Advantages):

a. Peningkatan Efisiensi Analisis: Sistem menyediakan visualisasi interaktif yang mengeliminasi kebutuhan administrator untuk membaca *log* teks mentah yang kompleks, secara signifikan mempercepat proses identifikasi IP penyerang (misalnya, IP 192.168.1.40) dan pola serangan.

b. Skalabilitas dan Stabilitas Arsitektur: Penggunaan Docker dan Loki membuktikan bahwa arsitektur yang dibangun ringan dan stabil, mampu menampung lonjakan *log* tinggi dari serangan tanpa mengalami *bottleneck* atau kehilangan entri.

c. Kontribusi Konseptual: Sistem ini efektif berfungsi sebagai komponen inti dari sistem SIEM (*Security Information and Event Management*) yang *lightweight* dan *open-source*, khususnya untuk lingkungan jaringan kecil hingga menengah.

3. Kekurangan dan Keterbatasan (Disadvantages/Limitations):

a. Keterbatasan Respon: Sistem saat ini merupakan alat deteksi pasif; tidak dapat melakukan tindakan respon otomatis (mitigasi) seperti memblokir IP yang dicurigai secara instan tanpa bantuan *tool* eksternal.

b. Dependensi Konfigurasi Query: Efektivitas deteksi sangat bergantung pada ketepatan konstruksi *query* LogQL dan *pattern matching* yang harus ditetapkan secara manual oleh administrator, memerlukan pemahaman mendalam tentang pola *log* yang dicari.

4. **Pengembangan Selanjutnya (Possibilities for Further Development):**

a. Integrasi *Closed-Loop System*: Mengembangkan mekanisme Otomatisasi Respon dengan mengintegrasikan hasil *alert* Grafana ke *tool* mitigasi (misalnya, *Fail2Ban* atau *Alertmanager*), memungkinkan sistem secara otomatis memblokir IP yang terbukti berbahaya.

b. Analisis Log Lintas-Platform: Memperluas cakupan *monitoring* log untuk mencakup *Web Server Logs* (Nginx/Apache), log aplikasi, atau *Traffic Flow* (Netflow) untuk menghasilkan *insight* keamanan yang lebih komprehensif dan deteksi *anomali* yang lebih canggih.

c. Pengembangan Machine Learning (ML): Mengintegrasikan modul ML sederhana untuk mendeteksi anomali tanpa perlu *pattern matching* LogQL yang kaku, sehingga sistem dapat mengidentifikasi serangan *zero-day* atau *custom attack*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

[1] A. Rahman, "Network Security Threats and Mitigation Strategies in Cloud-Based Infrastructure," IEEE Access, vol. 12, pp. 95567–95580, 2024.

[2] D. Kiswanto, R. Surbakti, A. Y. Ritonga, and H. A. Nasution, "Pengembangan dan Implementasi Sistem Deteksi Serangan DDoS Berbasis Algoritma Random Forest," *Bulletin of Information Technology (BIT)*, vol. 6, no. 3, pp. 247–256, 2025.

[3] F. Arifin and S. Lestari, "Implementasi Containerization untuk Efisiensi Sistem Monitoring Jaringan," Jurnal Teknologi Informasi dan Komunikasi Terapan, vol. 9, no. 2, pp. 45–53, 2024.

[4] D. Nuraini, R. Hidayat, and L. Santoso, "Real-Time Network Monitoring using Prometheus and Grafana Integration," International Journal of Computer Applications, vol. 182, no. 14, pp. 21–28, 2022.

[5] M. Prasetyo and A. Ramadhan, "Hybrid Log Analysis for Threat Visualization in Network Infrastructure," Journal of Information Security and Data Systems, vol. 8, no. 3, pp. 89–97, 2023.

[6] L. Chen, "Docker-Based Modular Infrastructure for Scalable Network Monitoring Systems," Applied Computing and Informatics, vol. 20, no. 1, pp. 12–20, 2024.

[7] M. A. Zeeshan, "Network Monitoring Using Grafana: An Integrated Approach for Enterprise Infrastructure
Management," International Journal of Engineering Research and Applications, vol. 15, no. 8, pp. 1–5, 2025.

[8] R. Nugraha and F. Kurniawan, "Real-Time Network Behavior Analysis for Malware Detection Using Flow Monitoring," Jurnal Teknologi Informasi dan Keamanan Siber, vol. 9, no. 1, pp. 22–30, 2024.

[9] S. Akhter, M. Rahman, and T. Alam, "Docker Performance Evaluation across Operating Systems," Applied Sciences, vol.
14, no. 15, p. 6672, 2024.

[10] M. Hidayat and D. Rahmawati, "Analysis of Docker Container Implementation in SIEM Infrastructure," Journal of Applied Informatics and Computing, Politeknik Negeri Batam, 2023.

[11] M. D. Elradi, "Prometheus & Grafana: A Metrics-Focused Monitoring Stack," Journal of Computer Allied Intelligence, vol. 3, no. 3, pp. 28–39, 2025.

[12] M. Marwan, M. Khalid, and N. Azmi, "The Analysis of Honeypot Performance Using Grafana Loki and ELK Stack Visualization," Info Sains: Jurnal Ilmiah Teknologi Informasi, vol. 5, no. 3, pp. 12–20, 2023.

[13] J. González, L. Pacheco, and D. Vega, "How Qonto Used Grafana Loki to Build Its Network Observability Platform," Grafana Labs Technical Report, 2023. [Online]. Available: https://grafana.com/blog/2023/08/11/howqonto-used-grafana-loki-to-build-itsnetwork-observability-platform/

[14] M. Bajpai, "Building Time-Series Data Monitoring Pipeline from the Cloud to Grafana," International Journal for Multidisciplinary Research (IJFMR), vol. 5, no. 1, pp. 1–5, 2023.

[15] Rachman, D. A., Muhyidin, Y., & Sunandar, M. A. (2023). *Analysis Quality of Service of Internet Network Fiber to the Home Service PT. XYZ Using Wireshark*. Jurnal Informatika dan Teknik Elektro Terapan (JITET), 11(3S1), 997–1006. https://doi.org/10.23960/jitet.v11i3s1.3436

[16] I. Rahmadani, M. Yusuf, and P. Andika, "Integrating Data Stream Processing for Real-Time Network Analysis Using Grafana," Jurnal Ilmu Komputer Terapan, vol. 8, no. 1, pp. 45–53, 2025.

[17] W. Seo and W. Pak, "Real-Time Network Intrusion Prevention System Based on Hybrid Machine Learning," IEEE Access, vol. 9, pp. 46386–46397, 2021.

[18] R. Pratama and D. Sari, "Centralized LogBased Detection of Anomalous IP Activity in Local Networks," Jurnal Teknologi Informasi dan Komputer, vol. 10, no. 2, pp. 56–63, 2024.

[19] H. Li, K. Zhang, and L. Yu, "Network Intrusion Detection Using Deep Autoencoder Models," IEEE Access, vol. 12, pp. 101233–101245, 2024.