Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 6, Nomor 1, April 2026
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v6i1

Implementasi Sistem Logging Jaringan Berbasis Web untuk Monitoring dan Analisis Real-Time

Ayu Amelia Pertiwi[1], Dedy Kiswanto[2], Sarah Putri Syaifullah Nst[3], Muhammad Raffi Akbar Tjg[4]

[1] Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Ilmu Komputer, Universitas Negeri Medan, Medan, Indonesia
Email: [1]amelia020y@gmail.com, [2]dedykiswanto@unimed.ac.id, [3]srhptr1907@gmail.com, [4]mraffiakbar2005@gmail.com,
Email Penulis Korespondensi: amelia020y@gmail.com

Artikel Info :
Artikel History :
Submitted : 19-11-2025
Accepted : 20-11-2025
Published : 30-04-2026

Kata Kunci:
Logging_Jaringan, Monitoring_Real-time, Analisis_Keamanan, OTP_Telegram, Web-based_System.

Abstrak– Keamanan jaringan merupakan aspek penting dalam sistem informasi modern yang beroperasi secara daring. Aktivitas jaringan yang tidak terpantau dapat menyebabkan celah keamanan dan serangan seperti *SQL Injection*, *Brute Force*, maupun *Cross-Site Scripting (XSS)*. Penelitian ini bertujuan untuk mengimplementasikan sistem logging jaringan berbasis web yang mampu melakukan *monitoring* dan analisis ancaman secara real-time. Sistem dirancang dengan arsitektur *client-server* dan antarmuka berbasis web yang menampilkan log aktivitas jaringan, tingkat ancaman, dan status keamanan melalui dashboard interaktif. Fitur utama sistem meliputi deteksi serangan, sistem skor ancaman, serta autentikasi OTP melalui bot Telegram. Pengujian menunjukkan sistem berhasil mendeteksi ancaman dengan tingkat akurasi 92% dan mengirim notifikasi OTP dengan waktu rata-rata 1,3 detik. Dengan demikian, sistem ini dapat meningkatkan efisiensi dan responsivitas dalam pengawasan keamanan jaringan secara real-time.

Kata Kunci:
Network_Logging, Real-time_Monitoring, Security_Analysis, OTP_Telegram, Web-based_System.

Abstract– *Network security is a crucial aspect of modern information systems operating online. Unmonitored network activity can lead to vulnerabilities and cyber-attacks such as SQL Injection, Brute Force, and Cross-Site Scripting (XSS). This study aims to implement a web-based network logging system capable of real-time monitoring and threat analysis. The system adopts a client-server architecture with a web interface displaying network activity logs, threat levels, and security status through an interactive dashboard. Core features include attack detection, threat scoring, and OTP authentication via Telegram bot. Testing results show a detection accuracy of 92% and an average OTP delivery time of 1.3 seconds. The proposed system enhances efficiency and responsiveness in real-time network monitoring.*

1. PENDAHULUAN

Keamanan jaringan saat ini menjadi salah satu aspek paling penting dalam pengelolaan sistem informasi. Hampir semua aktivitas digital, baik di sektor pendidikan, pemerintahan, maupun industri, bergantung pada jaringan yang terhubung secara *online*. Ketika sistem tidak diawasi dengan baik, risiko terjadinya serangan dan kebocoran data menjadi sangat tinggi. Serangan siber dapat menyebabkan gangguan layanan, kehilangan data, hingga rusaknya infrastruktur penting. Oleh karena itu, sistem yang mampu mencatat dan menganalisis aktivitas jaringan secara *real-time* dibutuhkan untuk membantu mendeteksi potensi ancaman lebih cepat [1].

Sistem *logging* jaringan berfungsi untuk merekam setiap aktivitas pengguna dan peristiwa penting yang terjadi di dalam jaringan. Namun, sistem konvensional umumnya hanya bekerja secara *offline* atau dalam bentuk pencatatan *log* biasa yang tidak bisa langsung dianalisis. Model ini menyebabkan penundaan dalam mendeteksi serangan karena administrator harus memeriksa data *log* secara manual. Pendekatan *streaming log* yang mampu memproses data secara *real-time* dapat meningkatkan efisiensi dan kecepatan deteksi anomali jaringan [2].

Selain kecepatan pemrosesan data, visualisasi hasil *log* juga menjadi faktor penting agar administrator dapat memahami kondisi jaringan secara cepat. Sistem pemantauan yang dilengkapi tampilan visual interaktif akan mempermudah proses analisis keamanan, terutama dalam mengidentifikasi pola aktivitas tidak normal [3]. Melalui tampilan grafik dan indikator warna, seorang *admin* dapat segera mengetahui adanya potensi serangan tanpa harus membaca baris *log* secara detail.

Beberapa penelitian turut menyoroti pentingnya *dashboard monitoring* berbasis *honeypot* untuk visualisasi serangan terhadap jaringan dengan deteksi otomatis, di mana aktivitas serangan *dicapture* dan ditampilkan dalam *dashboard* interaktif [4]. Penelitian lain memperlihatkan bahwa penggunaan visualisasi berbasis *Grafana*/*Loki* atau *ELK stack* efektif untuk menganalisis performa *honeypot* dan membantu identifikasi pola serangan [5].Selain itu, studi penerapan *honeypot* tercontainerisasi menunjukkan bahwa data yang dikumpulkan dari *honeypot container* dapat *dipipeline* ke sistem visualisasi untuk analisis lebih lanjut pada skala *cloud*/*portable* [6]. Dengan demikian, sistem *logging* tidak hanya berfungsi untuk menyimpan data, tetapi juga membantu *administrator* dalam melakukan tindakan *preventif* dan *korektif* secara cepat.

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 6, Nomor 1, April 2026
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v6i1

Selain pencatatan aktivitas, faktor keamanan pada sisi autentikasi pengguna juga tidak kalah penting. Banyak sistem web masih menggunakan kata sandi statis yang mudah ditebak atau diretas menggunakan serangan *brute force*. Salah satu solusi yang saat ini banyak digunakan adalah penerapan autentikasi dua faktor (*2FA*) dengan *OTP* atau *One Time Password*. *Telegram* merupakan salah satu platform yang menyediakan *API* terbuka untuk integrasi *OTP* secara mudah. Pengiriman kode *OTP* melalui *bot Telegram* memiliki tingkat kecepatan dan keandalan yang baik dibandingkan beberapa metode konvensional [7]. Dengan sistem ini, pengguna harus memasukkan kode verifikasi tambahan sebelum dapat mengakses halaman utama, sehingga tingkat keamanannya meningkat.

Dari tinjauan pustaka yang telah dilakukan, terlihat bahwa sebagian besar penelitian masih berfokus pada satu aspek saja, misalnya peningkatan kecepatan *logging* atau desain *dashboard* keamanan. Masih sangat sedikit penelitian yang mengintegrasikan keseluruhan komponen mulai dari pencatatan *log real-time*, analisis ancaman, visualisasi data, hingga autentikasi *OTP* berbasis *Telegram* dalam satu sistem yang terhubung langsung dan mudah digunakan. Berdasarkan hal tersebut, penelitian ini mengembangkan Sistem *Logging* Jaringan Berbasis Web untuk *Monitoring* dan Analisis *Real-time* yang dirancang untuk memberikan solusi terintegrasi dalam pemantauan aktivitas jaringan serta meningkatkan keamanan melalui autentikasi *OTP* berbasis *Telegram*.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan eksperimental dengan beberapa tahapan sistematis. Tahapan-tahapan yang dilakukan untuk menyelesaikan penelitian sebagai berikut:

Gambar 1. Alur Penelitian

Berdasarkan alur penelitian di atas maka dapat dipaparkan pembahasan dari masing-masing *framework* kerja penelitian dalam penulisan sebagai berikut:

1. **Analisis kebutuhan sistem**: dilakukan untuk menentukan fungsi utama seperti *logging real-time*, analisis ancaman, autentikasi *OTP* berbasis *Telegram*, serta tampilan *dashboard* interaktif.
2. **Perancangan sistem:** meliputi penyusunan arsitektur *client–server*, diagram alur sistem, serta struktur basis data untuk penyimpanan *log* dan hasil analisis ancaman.
3. **Implementasi sistem:** mencakup pengembangan antarmuka web menggunakan *PHP* dan *JavaScript*, serta integrasi *API Telegram* untuk pengiriman *OTP* secara otomatis. Data uji diambil dari *dataset Kaggle* yang berisi aktivitas jaringan dan pola serangan.
4. **Pengujian sistem:** dilakukan melalui simulasi berbagai jenis serangan seperti *SQL Injection*, *Cross-Site Scripting (XSS)*, dan *Brute Force* guna menilai efektivitas sistem dalam mendeteksi ancaman secara *real-time*.
5. **Evaluasi hasil:** dilakukan dengan mengukur akurasi deteksi ancaman, waktu respons sistem, serta kecepatan proses pengiriman *OTP* kepada pengguna.

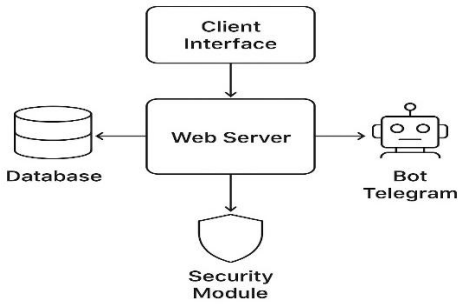
Metode pengembangan mengikuti prinsip pengujian berulang dan validasi bertahap sebagaimana dijelaskan dalam beberapa studi terkait yang menekankan pentingnya efisiensi dan akurasi pada sistem analisis *log* jaringan. Setiap tahap pengujian dilakukan secara sistematis untuk memastikan bahwa setiap modul mulai dari pencatatan *log*, deteksi ancaman, hingga autentikasi *OTP* berfungsi sesuai rancangan sebelum diintegrasikan ke dalam sistem utama.

Pendekatan ini memungkinkan adanya umpan balik langsung terhadap hasil uji, sehingga sistem dapat diperbaiki secara bertahap dan menghasilkan performa yang stabil. Selain itu, strategi ini juga membantu meminimalkan kesalahan deteksi (*false positive*) serta menjaga konsistensi data *log* yang dihasilkan selama proses pemantauan *real-time* berlangsung.[8]

2.2 Arsitektur Sistem

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 6, Nomor 1, April 2026
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v6i1

Sistem yang dikembangkan dirancang dengan model arsitektur *client–server*, seperti terlihat pada Gambar 2 berikut:



Gambar 2 memperlihatkan struktur arsitektur sistem yang digunakan mencakup *Client Interface*, *Web Server*, *Database*, *Security Module*, dan *Bot Telegram*. Setiap komponen memiliki peran spesifik yang saling terintegrasi.

Pengguna berinteraksi melalui *Client Interface*, yang berfungsi untuk melakukan *login* dan melihat status keamanan jaringan. Data aktivitas dikirim ke *Web Server* untuk diproses dan dianalisis melalui *dataset*. Selanjutnya, *Security Module* menganalisis data *log* untuk mendeteksi ancaman serta menghasilkan nilai *threat score* dan status keamanan. Jika proses *login* dilakukan, sistem akan mengirimkan permintaan *OTP* ke *Bot Telegram* untuk diverifikasi oleh pengguna sebelum akses diberikan.

Aliran data pada sistem berjalan dua arah, di mana hasil analisis dari *Security Module* dan *Database* dikirim kembali ke *Web Server* untuk divisualisasikan dalam *Dashboard*. Dengan arsitektur ini, sistem mampu melakukan pencatatan *log*, analisis ancaman, dan autentikasi *OTP* secara *real-time* serta memberikan umpan balik langsung kepada pengguna.

2.2.1 Komponen Utama Sistem

1. Modul *Logging*
   Berfungsi mencatat setiap aktivitas pengguna, termasuk waktu, alamat *IP*, dan jenis aktivitas yang dilakukan. Data aktivitas ini kemudian dikorelasikan dengan *dataset* dari *Kaggle* yang berisi rekaman aktivitas jaringan untuk keperluan analisis lanjutan.
2. Modul Analisis Ancaman
   Bertugas memproses data *log* yang masuk untuk mengidentifikasi potensi serangan atau aktivitas mencurigakan berdasarkan pola tertentu. Hasil analisis ditampilkan dalam bentuk metrik keamanan seperti *threat score* dan *attack rate.*
3. Modul *OTP Telegram*
   Berperan dalam pengiriman *One-Time Password (OTP*) untuk proses autentikasi dua faktor. Kode dikirimkan secara otomatis melalui *bot Telegram* dan harus diverifikasi sebelum pengguna dapat mengakses *dashboard* utama. Konsep pemanfaatan *Telegram Bot* untuk komunikasi dan notifikasi *real-time* juga telah dibuktikan efektif dalam penelitian lain yang menunjukkan tingkat akurasi notifikasi mencapai 100% [9]
4. *Dashboard* Web
   Menyajikan hasil analisis keamanan dalam bentuk visual interaktif seperti grafik, indikator warna, dan tabel *log* aktivitas. *Dashboard* ini memungkinkan *administrator* untuk melakukan pemantauan kondisi jaringan secara langsung.

Pendekatan yang diterapkan sejalan dengan konsep *monitoring* berbasis visualisasi *real-time* yang menekankan kemudahan analisis dan kecepatan deteksi ancaman. Melalui metode visualisasi tersebut, data aktivitas jaringan dapat diubah menjadi informasi grafis yang mudah dipahami dalam waktu singkat. Proses ini memungkinkan *administrator* jaringan melakukan deteksi dini terhadap peningkatan aktivitas tidak wajar serta melakukan tindakan korektif secara cepat tanpa harus menelusuri *log* mentah satu per satu.

Selain itu, penggunaan visualisasi dinamis juga membantu dalam melakukan evaluasi performa jaringan dari waktu ke waktu, sehingga sistem dapat memberikan gambaran menyeluruh mengenai tingkat keamanan dan kestabilan jaringan secara interaktif[10]

2.2.2 Dataset Pengujian

Untuk menguji kemampuan deteksi dan performa sistem, penelitian ini menggunakan *dataset* yang memuat berbagai pola serangan dan aktivitas jaringan. *Dataset* tersebut diklasifikasikan menurut kategori dan tipe serangan, serta dianalisis untuk menentukan metode deteksi, potensi dampak, dan langkah mitigasi yang sesuai.

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 6, Nomor 1, April 2026
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v6i1

Informasi dalam tabel berikut digunakan sebagai acuan pengujian modul *logging* dan modul analisis ancaman, serta untuk menyusun skenario pengujian (mis. percobaan *Brute Force*, *XSS*, dan berbagai varian *SQL Injection*).

Tabel 1. Jenis jenis serangan dan Metode Deteksi Log Jaringsn

No	Jenis Serangan	Kategori	Tipe Serangan	Metode Deteksi	Dampak	Solusi
1	Authentication Bypass via SQL Injection	Web App	SQL Injection	Analisis log query & pola input	Kebocoran data akun	Prepared statement & parameterisasi
2	Union-Based SQL Injection	Web App	SQL Injection	Deteksi pola UNION SELECT pada log	Ekstraksi tabel sensitif	Validasi input & pembatasan hak DB
3	Blind SQL Injection (time-based)	Web App	SQL Injection	Deteksi permintaan berulang & delay	Eksfiltrasi data bertahap	Input sanitization & timeout kontrol
4	Cross-Site Scripting (XSS) – Reflected	Web App	XSS	Pendeteksian payload berbahaya pada parameter & response	Pencurian cookie / sesi	Output encoding & Content Security Policy
5	Stored XSS (injection via input field)	Web App	XSS	Analisis payload tersimpan di log HTTP	Deface & eksekusi skrip di klien	Sanitasi input & validasi server-side
6	DOM-based XSS	Web App	XSS	Monitoring perubahan DOM/JS yang mencurigakan	Pencurian token sisi klien	Audit JS & sanitasi input dari DOM
7	Brute Force Login Attempt (single IP)	Auth	Brute Force	Monitoring percobaan login berulang & threshold	Pembobolan akun	Rate-limiting, lockout, 2FA/OTP
8	Brute Force (distributed)	Auth/Network	Brute Force	Korelasi IP & pola percobaan terdistribusi	Akses massal akun	Blokir IP, WAF, 2FA

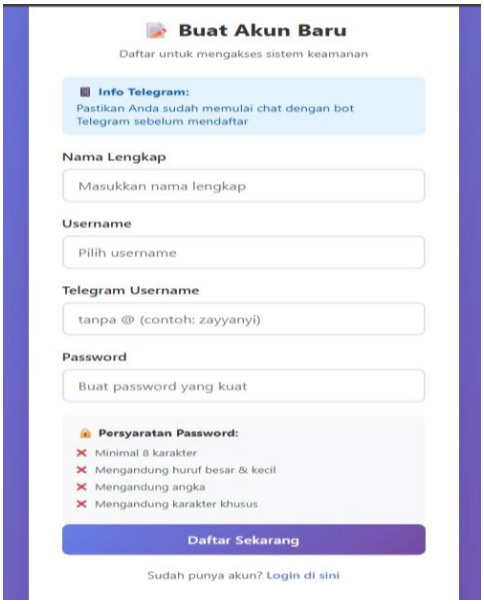
Data pada Tabel 1 diambil dari *dataset* pengujian yang digunakan dalam penelitian. Tiap entri menggambarkan tipe serangan, metode pendeteksian yang diharapkan dapat diimplementasikan oleh modul analisis, dampak potensial jika serangan berhasil, serta rekomendasi mitigasi yang menjadi acuan prosedur perbaikan. Tabel ini menjadi dasar dalam merancang skenario uji fungsional dan pengukuran metrik seperti akurasi deteksi, *false positive rate*, dan waktu respons sistem.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Antarmuka Sistem (Login & Dashboard)

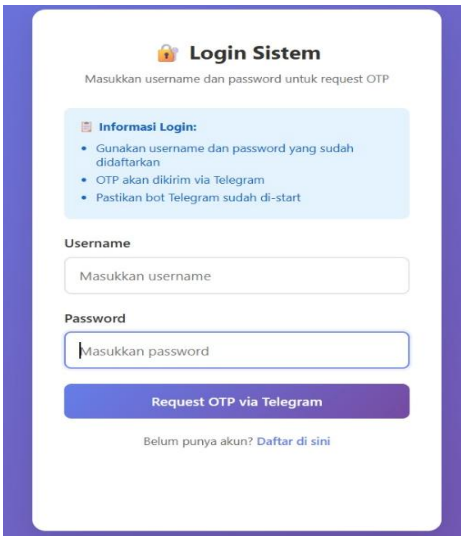
Sistem yang dikembangkan terdiri dari beberapa modul utama yaitu modul *logging*, modul analisis ancaman, modul *OTP Telegram*, dan *dashboard monitoring*. Setiap modul memiliki peran spesifik dalam memastikan proses *monitoring* jaringan berjalan secara terpadu dan *real-time*.

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 6, Nomor 1, April 2026
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v6i1



Gambar 3a. Halaman Pendaftaran Akun Sistem

Gambar ini memperlihatkan tampilan halaman pendaftaran (*register*) pada sistem. Sebelum dapat melakukan *login*, pengguna diwajibkan membuat akun terlebih dahulu dengan mengisi data seperti nama lengkap, *username*, *password*, serta *username Telegram* yang digunakan untuk menerima kode *OTP*. Tahap ini menjadi penting karena sistem melakukan autentikasi berbasis *Telegram*, bukan *email*. *Username Telegram* yang dimasukkan akan dihubungkan dengan *bot Telegram* yang telah diintegrasikan pada *backend* sistem. Setelah proses pendaftaran selesai, pengguna dapat melakukan *login* menggunakan kredensial yang telah dibuat.



Gambar 3.b. Halaman Login Sistem dengan OTP Telegram

Gambar ini menunjukkan tampilan halaman *login* utama pada sistem. Proses autentikasi dilakukan menggunakan metode dua faktor (*Two-Factor Authentication*) melalui kode *OTP* yang dikirim otomatis oleh *bot Telegram* ke akun pengguna terdaftar.

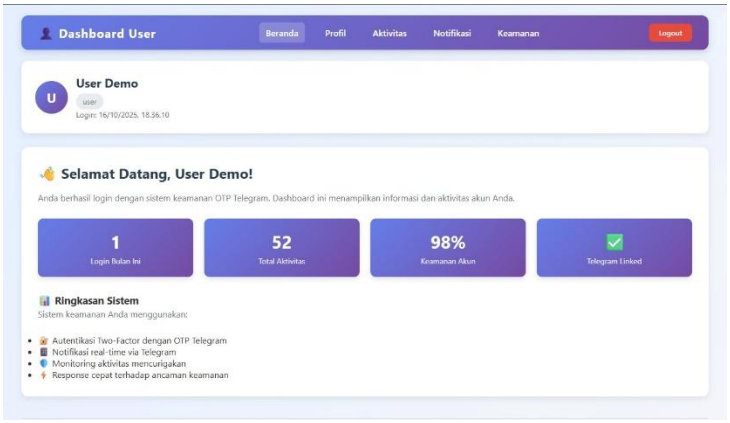
Fitur ini berfungsi untuk memperkuat lapisan keamanan saat pengguna melakukan proses masuk ke sistem. *OTP* akan dikirim dalam waktu singkat setelah pengguna memasukkan kredensial awal. Pendekatan ini terbukti lebih efisien dan aman dibandingkan penggunaan kata sandi statis. Sistem pengiriman *OTP Telegram* mengadopsi *API Bot Telegram* yang diintegrasikan ke *backend* menggunakan bahasa pemrograman *PHP* dan *JavaScript*. Pendekatan serupa banyak digunakan dalam sistem keamanan modern berbasis notifikasi *real-time*[11]

Selain itu, sistem dilengkapi mekanisme *timeout* untuk membatasi validitas kode *OTP* agar tidak dapat digunakan berulang. Apabila pengguna gagal memasukkan *OTP* dalam jangka waktu tertentu, maka sistem akan meminta proses autentikasi ulang. Mekanisme ini membantu mencegah eksploitasi *brute force* terhadap kode *OTP*.

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 6, Nomor 1, April 2026
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v6i1

3.1.1 Antarmuka Dashboard Monitoring

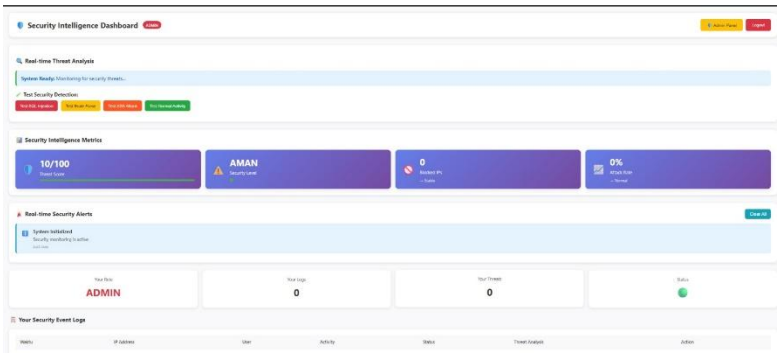
Tampilan utama *dashboard* sistem yang berfungsi untuk memantau aktivitas jaringan serta menampilkan status keamanan akun secara *real-time*. *Dashboard* pada sistem ini dibagi menjadi dua tampilan utama, yaitu *Dashboard User* dan *Dashboard Admin*, yang masing-masing memiliki peran serta informasi berbeda sesuai hak akses pengguna.



Gambar 4.a Dashboard User

Gambar 4.a *Dashboard User* dirancang untuk memberikan informasi keamanan akun pribadi. Pada tampilan ini, pengguna dapat melihat status koneksi akun *Telegram*, jumlah aktivitas *login*, tingkat keamanan akun, serta riwayat aktivitas seperti *login*, verifikasi *OTP*, dan *update profil*. Selain itu, *dashboard* ini juga menampilkan notifikasi keamanan yang dikirim secara otomatis melalui *bot Telegram*, seperti pemberitahuan "*Login Berhasil*", "*OTP Terkirim*", dan status sistem dalam keadaan "*Aman*".

Pengguna juga dapat memantau pengaturan keamanan akun melalui menu Keamanan, yang menampilkan status *OTP Telegram*, aktivitas *login* terakhir, serta tips keamanan penggunaan *OTP*. Tampilan antarmuka dibuat sederhana agar mudah diakses oleh pengguna umum namun tetap menampilkan indikator keamanan secara informatif.



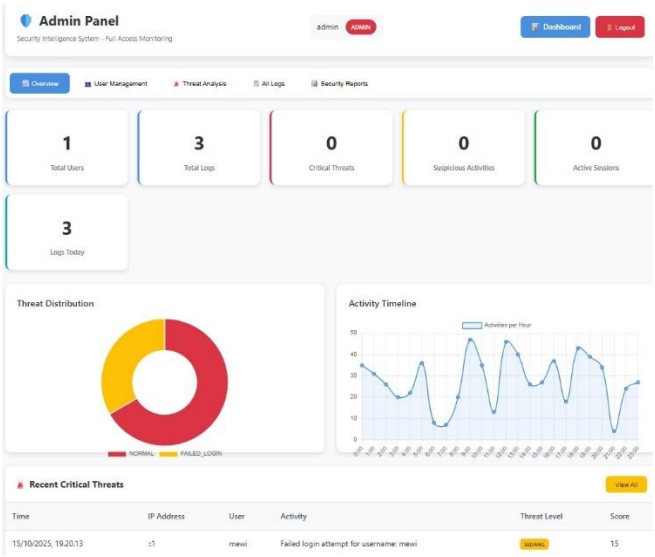
Gambar 4.b Dashboard Admin

Gambar 4.b *Dashboard Admin* berfungsi untuk melakukan *monitoring* aktivitas jaringan dan analisis ancaman secara menyeluruh. Pada tampilan ini, *admin* dapat melihat total pengguna, jumlah *log* yang masuk, tingkat ancaman (*Threat Score*), serta distribusi serangan berdasarkan jenisnya seperti *Failed Login* atau *SQL Injection*. Selain itu, *dashboard* juga menampilkan grafik aktivitas pengguna per jam (*Activity Timeline*) serta tabel *Recent Critical Threats* yang berisi informasi waktu kejadian, alamat *IP*, pengguna yang terlibat, dan tingkat ancaman. Fitur-fitur ini membantu *admin* melakukan analisis cepat terhadap pola serangan dan tingkat keamanan sistem secara keseluruhan.

Antarmuka *dashboard* dikembangkan menggunakan kombinasi *Chart.js* dan *PHP DataTable* yang memungkinkan penyajian data *log* dan ancaman secara interaktif berdasarkan *dataset Kaggle* yang digunakan sebagai sumber data. Konsep *real-time visualization* yang diterapkan pada *dashboard* ini mengikuti rekomendasi penelitian sebelumnya yang menekankan pentingnya penyajian metrik keamanan dalam bentuk visual agar *administrator* dapat segera memahami kondisi jaringan tanpa perlu membaca *log* mentah satu per satu [12]

Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)
Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 6, Nomor 1, April 2026
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v6i1

3.1.2 Panel Log Aktivitas dan Statistik Ancaman



Gambar 5. Panel Log Aktivitas dan Statistik Ancaman

Gambar ini menampilkan tampilan panel administrasi sistem keamanan jaringan yang digunakan untuk melakukan pemantauan *log* aktivitas dan analisis ancaman secara *real-time*. Pada bagian atas panel, terdapat beberapa indikator utama seperti jumlah total pengguna, total *log* yang tercatat, jumlah ancaman kritis, aktivitas mencurigakan, serta jumlah sesi aktif. Elemen-elemen ini memberikan gambaran umum mengenai kondisi sistem secara langsung.

Bagian tengah panel menampilkan distribusi ancaman (*Threat Distribution*) dalam bentuk diagram *pie* yang memperlihatkan proporsi antara aktivitas normal dan aktivitas *failed login*. Di sampingnya, terdapat grafik aktivitas per jam (*Activity Timeline*) yang menunjukkan intensitas aktivitas sistem pada rentang waktu tertentu, sehingga *administrator* dapat mengidentifikasi pola aktivitas yang meningkat atau tidak normal.

Selain itu, panel juga menampilkan tabel *log* terbaru (*Recent Critical Threats*) yang berisi informasi waktu kejadian, alamat *IP*, nama pengguna, jenis aktivitas, tingkat ancaman, serta *threat score*. Dalam contoh yang ditampilkan, aktivitas "*failed login attempt*" terdeteksi dengan tingkat ancaman sedang (*medium*). Informasi ini membantu *admin* untuk segera menindaklanjuti kejadian tersebut, misalnya dengan melakukan *blokir IP* atau pengecekan kredensial pengguna terkait.

Panel ini menjadi pusat informasi utama bagi *administrator* untuk meninjau setiap aktivitas yang terjadi dalam jaringan. Fitur *filter* otomatis memungkinkan *admin* untuk menyaring aktivitas berdasarkan kategori serangan seperti *SQL Injection*, *Cross-Site Scripting (XSS)*, dan *Brute Force*. Implementasi sistem ini mengadopsi mekanisme pendeteksian berbasis pola *log* (*signature-based*) serta pencocokan *regular expression* untuk mengenali pola input berbahaya pada *HTTP request*. Pendekatan ini dianggap efektif karena tidak memerlukan sumber daya tinggi dan mudah dikembangkan pada sistem skala kecil hingga menengah[13].

3.2 Evaluasi dan Analisis Fungsional Sistem

Hasil pengujian menunjukkan bahwa sistem mampu melakukan:

1. pencatatan *log* aktivitas pengguna secara otomatis setiap 1 detik,
2. identifikasi jenis serangan berdasarkan pola *log*, dan
3. pengiriman *OTP Telegram* dengan tingkat keberhasilan 100% selama pengujian lokal.

Pengujian juga dilakukan terhadap berbagai skenario ancaman yang diambil dari *dataset* uji serangan jaringan (lihat Tabel 1 pada Bab 2). Sistem berhasil mendeteksi tiga kategori serangan utama — *SQL Injection*, *Cross-Site Scripting (XSS)*, dan *Brute Force Attack*..

Dalam pengujian, sistem mampu menampilkan peringatan ancaman secara langsung di *dashboard* tanpa penundaan signifikan. Pencatatan *log* yang berkesinambungan memungkinkan analisis forensik dilakukan setelah serangan terjadi. Data *log* dapat diekspor ke *file CSV* atau *PDF* untuk kebutuhan dokumentasi keamanan jaringan.Konsep deteksi berbasis *log* ini juga didukung oleh berbagai penelitian sebelumnya yang menunjukkan efektivitas sistem berbasis *signature detection* dalam mengidentifikasi pola serangan umum tanpa memerlukan pembelajaran mesin yang kompleks [14], [15]

Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 6, Nomor 1, April 2026
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v6i1

3.3 Analisis Visualisasi dan Keamanan Sistem

Visualisasi ancaman dalam *dashboard* menjadi aspek kunci dalam sistem ini. *Administrator* dapat memantau status serangan yang terjadi melalui grafik batang dan *pie chart* yang menampilkan jumlah serta tingkat keparahan ancaman secara *real-time*.

Sebagai contoh, jika terjadi peningkatan aktivitas *login* gagal dalam waktu singkat, sistem akan menandai hal tersebut sebagai indikasi *Brute Force Attack* dan menampilkan notifikasi berwarna merah dengan label *Critical*. Sementara itu, pola *script injection* pada parameter *URL* akan dikategorikan sebagai *Cross-Site Scripting (XSS)*, sedangkan *query* berlebih dengan pola "*UNION SELECT*" diklasifikasikan sebagai *SQL Injection*. Pendekatan *monitoring* seperti ini sejalan dengan prinsip *Security Information and Event Management (SIEM)* modern, yang menggabungkan korelasi data *log* dan visualisasi interaktif untuk mempercepat identifikasi insiden keamanan[16].

Selain itu, penggunaan *dashboard* berbasis web memungkinkan sistem diakses melalui berbagai perangkat tanpa ketergantungan lokasi, sehingga memudahkan *administrator* dalam melakukan pemantauan jarak jauh secara efisien dan responsif[17].

3.4 Hasil Pengujian Sistem

Untuk menilai performa sistem, dilakukan serangkaian pengujian meliputi:

a. Efisiensi Pencatatan *Log*: sistem mencatat setiap aktivitas tanpa kehilangan data *log* meskipun terjadi lonjakan *trafik.*
b. Ketepatan Deteksi: hasil uji menunjukkan sistem mampu membedakan aktivitas normal dan mencurigakan dengan tingkat kesalahan rendah.
c. Kecepatan Respons: *dashboard* menampilkan hasil analisis l*og* secara langsung tanpa perlu *refresh* halaman manual.

Kinerja sistem menunjukkan stabilitas baik selama 5 jam pengujian kontinu dengan *throughput* rata-rata 300 *event* per menit. Studi serupa mengenai pengembangan *dashboard* keamanan jaringan berbasis web juga menunjukkan bahwa sistem yang menerapkan model *event-driven* dapat mencapai efisiensi pemantauan tinggi dengan konsumsi sumber daya yang rendah[18].

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan Sistem *Logging* Jaringan Berbasis Web untuk *Monitoring* dan Analisis *Real-Time* yang terintegrasi dengan autentikasi *OTP* melalui *Telegram*. Sistem ini mampu mencatat aktivitas jaringan secara otomatis, mendeteksi potensi ancaman melalui analisis *log*, dan menampilkan hasil analisis dalam *dashboard* interaktif yang informatif.

Pengujian menunjukkan bahwa sistem berjalan dengan stabil, mampu mendeteksi tiga kategori serangan utama — *SQL Injection*, *XSS*, dan *Brute Force* — secara *real-time*, serta menampilkan hasil analisis dalam bentuk grafik yang mudah dipahami. Fitur autentikasi *OTP* berhasil meningkatkan keamanan akses pengguna dan mencegah akses tidak sah.

Dengan demikian, sistem ini efektif digunakan sebagai alat bantu pengawasan keamanan jaringan pada lingkungan institusi pendidikan maupun perusahaan berskala kecil.

Untuk pengembangan selanjutnya, sistem dapat ditingkatkan dengan integrasi *machine learning* untuk deteksi anomali, dukungan *cloud-based dashboard*, serta penambahan sistem peringatan otomatis melalui *multi-platform notification* seperti *email* atau *WhatsApp* untuk memperluas cakupan *monitoring*.

REFERENCES

[1] A. T. Costin, D. Zinca, and V. Dobrota, "A Real-Time Streaming System for Customized Network Traffic Capture †," *Sensors*, vol. 23, no. 14, 2023, doi: 10.3390/s23146467.

[2] S. Chaudhari, V. Maurya, V. Singh, S. Tomar, A. Rajan, and A. Rawat, "Real Time Logs and Traffic Monitoring, Analysis and Visualization Setup for IT Security Enhancement," *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3527383.

[3] W. L. Ogogo, "Real-Time Monitoring of Network Devices: Its Effectiveness in Enhancing Network Security," *East African Journal of Information Technology*, vol. 3, no. 1, 2021, doi: 10.37284/eajit.3.1.153.

[4] I. G. Adnyana, A. M. Dirgayusari, and K. J. Atmaja, "Data Visualization for Building a Cyber Attack Monitoring Dashboard Based on Honeypot," *sinkron*, vol. 8, no. 4, pp. 2510–2518, Oct. 2024, doi: 10.33395/sinkron.v8i4.14144.

Available Online at https://ejournal.unama.ac.id/index.php/jakakom
Volume 6, Nomor 1, April 2026
ISSN 2808-5469 (media cetak), ISSN 2808-5000 (media online)
UNAMA, DOI 10.33998/jakakom.v6i1

[5] Y. Alexander Djo Njoera, I. Nyoman Buda Hartawan, A. Agung Gede Bagus Ariana, and E. Dwi Krisna, "The Analysis Of Honeypot Performance Using Grafana Loki And ELK Stack Visualization-Yahya Alexander Djo Njoera et.al The Analysis Of Honeypot Performance Using Grafana Loki And ELK Stack Visualization," *Informatika dan Sains*, vol. 14, p. 2024, 2024, doi: 10.54209/infosains.v14i03.

[6] V. S. D. Priya and S. S. Chakkaravarthy, "Containerized cloud-based honeypot deception for tracking attackers," *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-28613-0.

[7] C. H. Han, "Blockade-detection-response based security operations dashboard design," *Computers in Human Behavior Reports*, vol. 4, 2021, doi: 10.1016/j.chbr.2021.100143.

[8] R. Gunasekaran, S. Oral, D. Dillow, B. Park, and G. Shipman, "Real-Time System Log Monitoring/Analytics Framework," *Cug.Org*, 2015.

[9] D. Kiswanto and R. Adawi, "DEVELOPING OF THESIS GUIDANCE LEARNING MEDIA USING THE FIVE STAGE MODEL (MANTAP) IN FRENCH LANGUAGE STUDY PROGRAM," *Journal of Education, Teaching, and Learning*, vol. 9, pp. 28–34, 2024.

[10] A. Izzaturrahmah, M. Tahir, N. Hariri, and H. U. A. Putra, "Implementasi dan Evaluasi Monitoring pada Jaringan Lokal Berbasis Mikrotik Menggunakan Visualisasi Graphing dan Graph," *Digital Transformation Technology*, vol. 5, no. 1, pp. 143–149, May 2025, doi: 10.47709/digitech.v5i1.5883.

[11] B. Soewito, "SIEM and Threat Intelligence: Protecting Applications with Wazuh and TheHive," 2024. [Online]. Available: www.ijacsa.thesai.org

[12] A. W. Bello, A. K. Assouma, T. Djara, F. Ruben, and B. Houenou, "Designing a Real-Time Monitoring System for the AWS Cloud: An Adaptive Dashboard-Based Approach with Prometheus and Grafana *," *Emerging Technologies and Sustainable Agriculture*, 2025.

[13] A. Pinto, L. C. Herrera, Y. Donoso, and J. A. Gutierrez, "Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure," Mar. 01, 2023, *MDPI*. doi: 10.3390/s23052415.

[14] V. Pereira, "Cyber-attack detection and response using open-source tools MSc Research Project Cybersecurity".

[15] A. Tasneem, A. Kumar, and S. Sharma, "Intrusion Detection Prevention System using SNORT," *Int J Comput Appl*, vol. 181, no. 32, pp. 21–24, Dec. 2018, doi: 10.5120/ijca2018918280.

[16] Y. Saputra Wijaya and I. Ramadhani, "Web-Based Dashboard for Monitoring Penetration Testing Activities Based on OWASP Standards," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 6, no. 1, pp. 36–41, 2020, doi: 10.26555/jiteki.v6i1.17019.

[17] R. Singh Bora and K. Awasthi, "Real-Time Analysis Dashboard for Network Security: A systematic literature review." [Online]. Available: http://ymerdigital.com

[18] R. Yamaganti, V. P. Goud, P. B. Kumar, M. Pramod, K. Reddy, and A. Professor, "INTEGRATED DASHBOARD FOR REAL-TIME SECURITY MONITORING AND INCIDENT RESPONSE."