JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 14 No. 1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v14i1.8230

IMPLEMENTASI SISTEM KEAMANAN JARINGAN BERBASIS WEB: LOGGING, DETEKSI ANOMALI DAN PEMBLOKIRAN OTOMATIS

Romatua Situmorang[1], Dedy Kiswanto[2], Najwatul Khoiriah[3], Fatima Asro Harahap[4]

[1,2,3,4)] Program Studi Ilmu Komputer, Universitas Negeri Medan, Jln. William Iskandar Ps. V, Kenangan Baru, Kec. Percut Sei Tuan, Deli Serdang, Sumatera Utara.

Keywords:
Logging;
Deteksi Anomali;
Pemblokiran Otomatis;
Brute Force; Keamanan Web.

Corespondent Email:
romatua.4233250040@mhs.unimed.ac.id

Abstrak. Penelitian ini bertujuan mengimplementasikan sistem keamanan jaringan berbasis web yang mengintegrasikan kemampuan logging, deteksi anomali, dan pemblokiran otomatis terhadap aktivitas mencurigakan. Pentingnya topik ini dilandasi oleh meningkatnya ancaman siber, seperti serangan brute force dan high request rate, yang secara langsung mengancam ketersediaan sistem. Sistem dirancang menggunakan Python Flask dan PostgreSQL sebagai arsitektur server, dengan metode deteksi Rule-Based untuk mengidentifikasi pola serangan spesifik. Pengujian dengan simulasi Kali Linux membuktikan bahwa sistem 100% efektif dalam memberikan notifikasi peringatan dan memblokir IP penyerang. Keberhasilan implementasi IPS (Intrusion Prevention System) internal ini divalidasi dengan penolakan akses oleh server dan tampilan pesan "IP Anda sedang diblokir sementara" pada browser penyerang , yang menunjukkan mitigasi real-time. Implementasi ini menunjukkan bahwa sistem dapat menjadi solusi proaktif yang adaptif sebagai lapisan keamanan tambahan dalam lingkungan jaringan berbasis web.

Abstract. *This research aims to implement a web-based network security system that integrates logging, anomaly detection, and automatic blocking capabilities against suspicious activities. The significance of this topic is underpinned by the increasing cyber threats, such as brute force and high request rate attacks, which directly jeopardize system availability. The system is designed using Python Flask and PostgreSQL as the server architecture, employing a Rule-Based detection method to identify specific attack patterns. Testing with Kali Linux simulations proved that the system was 100% effective in successfully providing alerts and effectively blocking attacker IPs. The successful implementation of this internal IPS (Intrusion Prevention System) is validated by the server's denial of access and the display of the message "IP Anda sedang diblokir sementara" ("Your IP is temporarily blocked") on the attacker's browser , demonstrating real-time mitigation. This implementation shows that the system can be an adaptive, proactive solution serving as an additional layer of security within web-based network environments.*

1. PENDAHULUAN

Keamanan jaringan telah menjadi pilar fundamental dalam menjaga kerahasiaan, integritas, dan ketersediaan data pada sistem berbasis web di era digital. Ancaman spesifik yang menargetkan ketersediaan sistem, seperti serangan brute force (percobaan login berulang) dan high request rate (DDoS), dapat menyebabkan kerugian operasional dan reputasi yang signifikan. Manajemen Risiko Siber kini menjadi prioritas utama bagi setiap entitas [1].

Upaya menjaga keamanan kini memerlukan mekanisme pemantauan aktivitas jaringan secara menyeluruh. Logging berperan penting dalam mencatat aktivitas sistem sebagai bahan analisis dan audit. Berdasarkan data log yang terekam, Deteksi Anomali diperlukan untuk mengidentifikasi aktivitas mencurigakan. Selanjutnya, sistem Pemblokiran Otomatis harus memberikan respons cepat terhadap ancaman tanpa menunggu intervensi manual dari administrator jaringan [2].

Meskipun terdapat solusi canggih, seperti implementasi Machine Learning untuk deteksi anomali [3][4], penelitian terdahulu seringkali bersifat parsial. Sejumlah studi fokus pada monitoring pasif atau analisis log tanpa tindakan mitigasi [5][6]. Celah penelitian (research gap) yang diangkat adalah kebutuhan mendesak untuk mengembangkan dan memvalidasi integrasi utuh dari tiga pilar keamanan (Logging, Deteksi, dan Pemblokiran Otomatis) dalam satu platform aplikasi web, yang berfungsi sebagai Proof of Concept (PoC) untuk respons mitigasi real-time. Oleh karena itu, penelitian ini bertujuan membangun sistem berbasis aturan menggunakan Python Flask dan PostgreSQL yang langsung dan terintegrasi untuk mengamankan server dari brute force dan high request rate.

2. TINJAUAN PUSTAKA

2.1 Logging dan Manajemen Data

Logging adalah tahap fundamental dalam keamanan jaringan, berfungsi mencatat setiap aktivitas sistem sebagai bahan analisis dan audit keamanan.

a. Pencatatan Aktivitas: Logging berperan penting dalam merekam setiap request dan response pada server web, membentuk data input primer untuk deteksi anomali. Analisis log server memungkinkan pelacakan pola permintaan yang tidak normal [5].
b. Peran PostgreSQL: Database PostgreSQL merupakan salah satu alternatif solusi bagi pengguna database yang mendukung banyak platform dan bebas lisensi [7]. PostgreSQL berfungsi sebagai repositori terpusat untuk menyimpan log aktivitas (tabel logs) dan IP yang diblokir (tabel alerts). Pemilihan database ini mendukung kebutuhan sistem untuk melakukan query cepat yang esensial untuk analisis real-time berbasis aturan.

2.2 Deteksi Anomali

Deteksi anomali bertujuan untuk mengidentifikasi pola yang tidak biasa atau mencurigakan yang mungkin menunjukkan adanya serangan atau pelanggaran keamanan [8]. Deteksi dalam proyek ini berfokus pada dua jenis serangan utama:

1. Serangan Brute Force: Brute force merupakan ancaman dari penyerang yang mencoba untuk login dengan menggunakan protocol SSH dan telnet untuk mengungkap password login [9]. Serangan ini ditandai dengan tingginya frekuensi percobaan login yang gagal dari satu alamat IP dalam waktu singkat.

   Relevansi proyek: Aturan deteksi brute force dalam proyek ini ditetapkan pada ambang batas lebih dari 5 login gagal dari IP yang sama dalam periode 5 menit.
2. Serangan High Request Rate (Potensi DDoS): Melibatkan pengiriman permintaan HTTP berlebihan untuk membanjiri server target. Serangan ini bertujuan melampaui kapasitas pemrosesan server, menyebabkan layanan menjadi tidak responsif [5].

   Relevansi proyek: Aturan deteksi high request rate dalam proyek ini ditetapkan pada ambang batas lebih dari 100 request dari IP yang sama dalam periode 1 menit.

2.3 Pemblokiran Otomatis

Pemblokiran otomatis adalah tindakan sistem untuk menolak akses dari entitas yang terdeteksi melakukan aktivitas berbahaya, yang merupakan esensi dari fungsi Intrusion Prevention System (IPS). Konsep IPS: IPS menggabungkan fitur deteksi (IDS) dengan kemampuan pemblokiran (Firewall).

Penelitian ini mengimplementasikan fungsi IPS secara internal pada server Flask: setelah modul deteksi (detect_anomalies.py) mencatat IP ke tabel alerts dengan status BLOCKED, aplikasi web segera menolak request dari IP tersebut, memvalidasi keberhasilan mitigasi real-time.

2.4 Alat Pengembangan dan Pengujian

Proyek ini sangat bergantung pada alat dan bahasa pemrograman yang spesifik untuk implementasi dan simulasi.

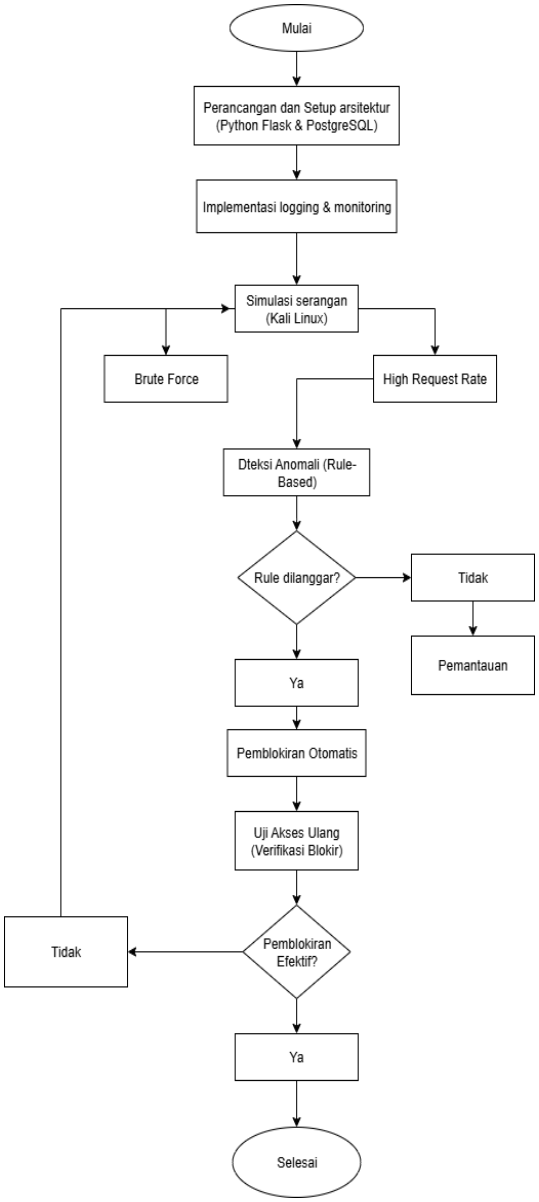
1. Python dan Flask: python merupakan salah satu bahasa pemrograman umum tingkat

tinggi yang paling populer[10]. Python adalah bahasa pemrograman utama yang digunakan. Flask merupakan micro-frameworkPython [11]. Kerangka kerja Flask digunakan untuk membangun server web minimalis yang menangani request HTTP, logging, dan logika pemblokiran (app.py).

2. Visual Studio Code (VS Code): VS Code merupakan editor kode sumber yang dikembangkan oleh Microsoft untuk sistem multiplatform [12]. Digunakan sebagai editor kode sumber (code editor) untuk pengembangan semua script Python (app.py dan detect_anomalies.py) dalam proyek ini.
3. Kali Linux: Digunakan sebagai platform penyerang. Kali linux adalah salah satu sistem operasi yang sering digunakan dalam melakukan penetration testing serta untuk audit keamanan jaringan computer dari keluarga Linux tingkat lanjut yang kembangkan oleh offensive Security[13]. Kali Linux digunakan untuk mensimulasikan serangan brute force dan high request rate terhadap server yang diimplementasikan.

3. METODE PENELITIAN

Metode penelitian ini menggunakan pendekatan Implementasi dan Pengujian Sistem (System Implementation and Testing). Penelitian ini bertujuan untuk memvalidasi konsep integrasi logging, deteksi anomali rule-based, dan pemblokiran otomatis pada lingkungan server aplikasi web. Alur penelitian ditunjukkan pada gambar 1. Berikut:



Gambar 1. Alur Penelitian

3.1 Lingkungan Implementasi dan Alat Penelitian

Sistem keamanan jaringan ini dikembangkan di atas lingkungan server minimalis, menggunakan teknologi open-source spesifik untuk implementasi PoC (Proof of Concept).

Aspek	Detail Teknis	Keterangan fungsional
Server Web	Python Flask	Berfungsi sebagai server utama, menangani request http, logging aktivitas dan

		pemeriksaan status pemblokiran.
Basis Data	PostgreSQL	Digunakan untuk menyimpan tabel logs (aktivitas) dan alerts (IP terblokir)
Modul Analisis	Python Script	Beroperasi sebagai background process yang menganalisis tabel logs dan menerapkan rule-based anomaly detection.
Alat Pengembangan	Visual Studio Code (VS Code)	Digunakan sebagai code editor utama untuk pengembangan semua script python.
Alat Pengujian	Kali Linux dan perintah curl/hping3	Digunakan untuk simulasi serangan brute force dan high request rate.

Tabel 3.1 Lingkungan Implementasi Dan Alat Penelitian

3.2 Arsitektur Sistem dan Metode Deteksi

Arsitektur sistem mengimplementasikan fungsi Intrusion Prevention System (IPS) secara internal, di mana respons pencegahan didasarkan pada logika yang dikodekan (rule-based).

1. Logging: Setiap request yang masuk ke server (app.py) akan dicatat ke dalam tabel logs di PostgreSQL, termasuk alamat IP sumber, aksi (LOGIN FAILED/LOGIN SUCCESS), dan timestamp.
2. Deteksi Anomali Berbasis Aturan (Rule-Based): Modul detect_anomalies.py menjalankan fungsi analisis berdasarkan ambang batas logis berikut:
   a. Brute Force: Deteksi dipicu ketika IP yang sama menghasilkan lebih dari 5 login gagal dalam periode 5 menit.
   b. High Request Rate: Deteksi dipicu ketika IP yang sama mengirimkan lebih dari 100 request dalam periode 1 menit.
3. Pemblokiran Otomatis: Jika deteksi anomali berhasil, IP penyerang dicatat ke tabel alerts dengan status BLOCKED. Aplikasi (app.py) akan melakukan pengecekan status IP di tabel alerts pada setiap request yang diterima; jika IP terblokir, request tersebut ditolak, dan browser klien menerima pesan penolakan.

3.3 Prosedur Pengujian dan Evaluasi

Pengujian dilakukan untuk memvalidasi fungsionalitas dan efektivitas sistem dalam mendeteksi dan merespons ancaman, sesuai dengan scenario serangan yang telah ditentukan.

3.3.1 Uji Simulasi Serangan

a. Uji Brute Force digunakan untuk memastikan medeteksi percobaan login berulang yang melanggar rule. Dilaksanakan dengan menggunakan script curl di Kali Linux untuk mengirimkan percobaan login gagal (>5 kali).

b. Uji Hight Request Rate digunakan untuk memastikan sistem mendeteksi lonjakan trafik tinggi yang melanggar rule. Dilaksanakan dengan menggunakan script curl atau hping3 paralel di Kali Linux untuk menghasilkan request yang sangat tinggi (>50 kali/menit).

3.3.2 Kriteria Evaluasi Keberhasilan

No	Tahap Evaluasi	Kriteria keberhasilan yang diukur
1	Deteksi dan Pencatatan Alert	Output Konsol Server mencatat ALERT ACTION dan IP penyerang harus tercatat di tabel alerts dengan status BLOCKED.
2	Efektivitas Pemblokiran	Penyerang mencoba mengakses kembali aplikasi. Browser penyerang harus menampilkan pesan penolakan. “IP Anda sedang diblokir sementara”.

Tabel 3.2 Kriteria Evaluasi Hasil

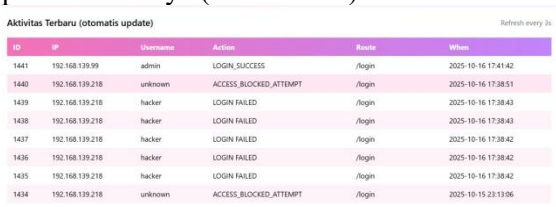
4. HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Bagian ini menyajikan temuan dari implementasi sistem dan pengujian fungsionalitas logging, deteksi anomali, dan pemblokiran otomatis.

4.1.1 Implementasi Fungsionalitas Logging dan Monitoring

Implementasi sistem berhasil menyediakan fungsi Logging dengan mencatat setiap aktivitas login dan request ke tabel logs di PostgreSQL. Data real-time ini ditampilkan melalui dashboard yang memberikan visibilitas penuh pada aktivitas jaringan. Struktur database mencakup tabel alerts yang secara khusus berfungsi menyimpan IP yang melanggar aturan (rule) dan status pemblokirannya (BLOCKED).



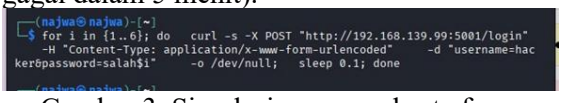
Gambar 2. Tampilan Dashboard

4.1.2 Hasil Uji Deteksi Anomali Berbasis Aturan

Pengujian dari Kali Linux memvalidasi kemampuan sistem untuk mendeteksi dua pola serangan:

1. Uji Brute Force:

Simulasi berhasil menghasilkan login gagal berulang, melampaui ambang batas rule deteksi yang ditetapkan (lebih dari 5 login gagal dalam 5 menit).



Gambar 3. Simulasi serangan brute force

Sistem secara real-time menampilkan notifikasi alert otomatis di konsol server dan mencatat alert BRUTE_FORCE di tabel alerts.

Gambar 4. Notifikasi serangan brute force

2. Uji High Request Rate:

Simulasi request frekuensi tinggi berhasil menghasilkan lonjakan trafik (tercatat 60 request dalam 1 menit pada pengujian).

Gambar 5. Simulasi Serangan High request rate

Sistem menampilkan notifikasi alert otomatis dan mencatat ALERT NOTIFY HIGH_REQUEST_RATE, yang segera memicu tindakan pemblokiran.

Gambar 6. Notifikasi serangan high request rate

4.1.3 Hasil Efektivitas Pemblokiran Otomatis

Tahap pengujian pemblokiran memverifikasi respons sistem setelah deteksi anomali:

a. Verifikasi Akses: Setelah IP penyerang dicatat ke tabel alerts dengan status BLOCKED, upaya akses lanjutan dari browser Kali Linux ditolak.
b. Pesan Penolakan: Browser penyerang secara konsisten menampilkan pesan penolakan akses: "IP Anda sedang diblokir sementara". Hasil ini memvalidasi keberhasilan fungsi mitigasi.

Gambar 7. Tampilan web setelah IP diblokir

4.2 Pembahasan

Pembahasan ini menginterpretasikan hasil pengujian dan mengaitkannya dengan tujuan penelitian serta kontribusi pada literatur keamanan jaringan.

4.2.1 Kontribusi terhadap Integrasi Keamanan (IPS Internal)

Keberhasilan pengujian membuktikan tercapainya tujuan utama penelitian, yaitu integrasi utuh dari tiga pilar keamanan. Fungsi logging (Pilar 1) menyediakan input data ke modul deteksi rule-based (Pilar 2), yang kemudian memicu fungsi pemblokiran (Pilar 3) secara real-time. Integrasi ini secara fungsional menciptakan Intrusion Prevention System (IPS) internal yang bekerja di lapisan aplikasi [2]. Mekanisme pemblokiran yang terjadi sebelum request diproses oleh logika aplikasi utama menjamin respons yang cepat dan efisien. Hal ini mengatasi research gap yang sering ditemukan dalam studi terdahulu yang hanya fokus pada deteksi tanpa mitigasi otomatis.

4.2.2 Analisis Efektivitas Rule-Based Detection

Penggunaan Deteksi Berbasis Aturan terbukti efektif untuk jenis serangan yang memiliki pola frekuensi yang jelas, seperti brute force dan high request rate. Meskipun akurasi deteksi ini tidak sebanding dengan algoritma Machine Learning tingkat lanjut (misalnya Random Forest atau XGBoost yang mencapai akurasi >99%; [3][14], metode rule-based ini menawarkan keunggulan dalam hal:

1. Keterbacaan dan Implementasi Cepat: Logika deteksi sederhana dan mudah diimplementasikan langsung pada server Flask (detect_anomalies.py).
2. Biaya Komputasi Rendah: Analisis yang dilakukan melalui query SQL sederhana jauh lebih ringan dibandingkan training dan inferencing model DL/ML, menjadikannya ideal untuk Proof of Concept dan lingkungan server skala kecil.

Namun, keterbatasan rule-based adalah ketidakmampuannya mengenali serangan yang memiliki pola baru atau tersembunyi (zero-day), yang memerlukan pengembangan menuju solusi berbasis AI di masa depan [4].

4.2.3 Implikasi Praktis

Implikasi dari temuan ini adalah bahwa sistem ini dapat diterapkan sebagai lapisan keamanan tambahan yang proaktif dalam lingkungan berbasis web. Keberhasilan pemblokiran (yang dikonfirmasi oleh pesan penolakan yang terlihat di Kali Linux) menunjukkan bahwa sistem ini memberikan nilai tambah yang signifikan di atas pertahanan pasif semata. Ke depannya, model ini dapat diperluas untuk mengintegrasikan Web Application Firewall (WAF) dan filtering yang lebih kompleks [15].

5. KESIMPULAN

Berdasarkan implementasi dan pengujian sistem keamanan jaringan berbasis web, beberapa poin penting dapat disimpulkan:

a. Integrasi Tiga Pilar Keamanan Berhasil Divalidasi: Sistem berhasil mengimplementasikan dan memvalidasi integrasi fungsi logging, deteksi anomali rule-based, dan pemblokiran otomatis dalam satu platform aplikasi web. Konsep ini menciptakan fungsi Intrusion Prevention System (IPS) internal yang terintegrasi penuh.

b. Efektivitas Deteksi dan Pemblokiran: Sistem terbukti efektif dalam mengenali pola serangan brute force dan high request rate berdasarkan ambang batas yang ditetapkan. Fungsi pemblokiran otomatis berjalan secara real-time, dibuktikan dengan penolakan akses oleh server yang menampilkan pesan pemblokiran kepada IP penyerang.

5.1 Kelebihan dan Kekurangan Sistem

a. Kelebihan:

1. Respons Real-Time: Sistem mampu memberikan respons mitigasi (pemblokiran) secara instan, mengatasi research gap pada studi yang terhenti pada tahap deteksi pasif.
2. Biaya Komputasi Rendah: Penggunaan metode rule-based pada aplikasi Flask menjadikannya solusi ringan dan efisien untuk server skala kecil.
3. Kontrol Penuh: Implementasi IPS secara internal memberikan kontrol penuh terhadap logika logging dan pemblokiran, tanpa ketergantungan pada firewall eksternal untuk filtering dasar.

b. Kekurangan:

Deteksi Kaku: Deteksi Rule-Based yang digunakan bersifat kaku dan terbatas; sistem tidak mampu mendeteksi serangan yang memiliki pola baru (zero-day) atau serangan

yang lebih canggih yang tidak tercakup dalam ambang batas logis.

5.2 Pengembangan Selanjutnya

Peningkatan Akurasi dengan AI: Pengembangan sistem selanjutnya harus mengintegrasikan algoritma Machine Learning (seperti Random Forest atau Jaringan Neural) untuk meningkatkan akurasi deteksi dan kemampuan mengenali pola anomali yang kompleks.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih yang setulus-tulusnya kepada Bapak Dedy Kiswanto, S.Kom., M.Kom., selaku Dosen Pengampu Mata Kuliah Keamanan Data dan Jaringan, atas bimbingan, arahan, dan dukungan yang tak terhingga selama proses penelitian dan penyusunan naskah ini. Kami juga berterima kasih kepada seluruh pihak yang telah memberikan dukungan moral dan fasilitas, sehingga riset mengenai implementasi sistem keamanan jaringan berbasis web ini dapat terselesaikan.

DAFTAR PUSTAKA

[1] R. C. Tarumingkeng, *Manajemen Risiko Siber*. Rudyct E-Press, 2025

[2] A. Kurniawan And L. M. Silalahi, "Analisis Keamanan Jaringan Menggunakan Intrusion Prevention System (Ips) Dengan Metode Traffic Behavior," *Electrician - Jurnal Rekayasa Dan Teknologi Elektro*, Vol. 11, No. 1, Pp. 1–16, 2023

[3] D. Kiswanto, F. Ramadhani, N. M. Surbakti, And N. 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.

[4] R. Nursiaga, N. Mulyana, H. Sanjaya, And G. Santoso, "Model Jaringan Neural Untuk Deteksi Anomali Pada Sistem Keamanan (Siber): Rancangan, Implementasi, Dan Analisis," *Jarekom: Jurnal Jaringan Dan Rekayasa Komputer*, Vol. 1, No. 1, Pp. 1–11, 2025.

[5] A. R. Nisa, A. D. Wijayanto, A. P. J. Priana, And A. Setiawan, "Analisis Log Server Untuk Mendeteksi Serang Ddos Pada Keamaan Jaringan Di Website," *Journal Of Internet And Software Engineering*, Vol. 1, No. 3, Pp. 1–17, 2024.

[6] J. U. Usla And A. Ikhwan, "Web Based Social Assistance Distribution Monitoring System Using Waterfall Method," *Journal Of Computer Networks, Architecture And High Performance Computing*, Vol. 5, No. 1, 2023.

[7] S. Munawaroh, "Mengeksplorasi Database Postgresql Dengan Pgadmin Iii", *Jurnal Teknologi Informasi Dinamik,* Vol. X, No. 2, Pp. 103-107, 2005.

[8] P. A. C. Setiawan, I. A. D. Giriantari, And N. Indra, "Tinjauan Literatur: Deteksi Anomali Berbasis Analisis Waktu Pada Can Bus Kendaraan Listrik", *Jurnal Ilmiah Teknik Elektro,* Vol. 6, No. 1, Pp. 72-84, 2025.

[9] B. Arifwidodo, Y. Syuhada, And S. Ikhwan, "Analisis Kinerja Mikrotik Terhadap Serangan Brute Force Dan Ddos," *Techno.Com*, Vol. 20, No. 3, Pp. 392–399, Aug. 2021.

[10] A.C. Darmawan, "Pengembangan Aplikasi Berbasis Web Dengan Python Flask Untuk Klasifikasi Data Menggunakan Metode Decision Tree C4.5", Uii, Yogyakarta, 2023.

[11] Y. T. Bota, N. Setiawati, "Pengembangan Sistem Informasi Perantara Bisnis Menggunakan Framework Flask", Vol.3, No.2, Pp.79-93, 2022.

[12] A. F. Oklilas And S. Pangestu, "Dashboard Monitoring Perangkat It Berbasis Website Pada Pt Kpi Ru Iii Plaju," *Jitet (Jurnal Informatika Dan Teknik Elektro Terapan)*, Vol. 12, No. 3, Pp 3665-3674, 2024.

[13] T. Yusnanto, M. A. Muin, And S. Wahyudiono, "Analisa Infrastruktur Jaringan Wireless Dan Local Area Network (Wlan) Meggunakan Wireshark Serta Metode Penetration Testing Kali Linux", *Journal On Education,* Vol.4, No.04, pp 1470-1476, 2022.

[14] A. M. A. Rudianto, E. S. Pramukantoro, And D. Kurnianingtyas, "Implementasi Sistem Deteksi Anomali Pada Jaringan Komputer Dengan Pendekatan Xgboost Dan Data Snmp," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, Vol. 9, 2025.

[15] Nurhayati, Atthariq, And Aswandi, "Implementasi Keamanan Jaringan Dengan Metode Web Application Firewall (Waf)," *Jurnal Teknologi Rekayasa Informasi Dan Komputer*, Vol. 8, No. 2, Pp. 48–56, 2022.