Teknik: Jurnal Ilmu Teknik dan Informatika
Volume 6 Nomor 2 Oktober 2026
e-ISSN: 2798-2513; p-ISSN: 2808-8751, Hal *87-97*
DOI: https://doi.org/10.51903/teknik.v6i2.1271
Available online at: https://journal.stiestekom.ac.id/index.php/TEKNIK

Penerapan DFA untuk Deteksi Serangan TCP SYN dan Ping Flood dengan Visualisasi pada Lingkungan GNS3

Yohanes Gerardus Haga Zai[1], Zulfahmi Indra[2], Sevta Triwana Simamora[3], Elga Sari Tanjung[4]
[1-3]Ilmu Komputer, Fakultas Matematika dan Pengetahuan Alam, Universitas Negeri Medan

Korespondensi penulis: yohanesgerardushagazai@gmail.com

Abstract. *The rapid development of computer network technology has increased communication efficiency but also raised the risk of cyberattacks, particularly Denial of Service (DoS) attacks such as TCP SYN Flood and ICMP Flood. Conventional Intrusion Detection Systems (IDS) generally rely on static log analysis and lack responsive real-time monitoring visualization. This research aims to design and implement an Intrusion Detection System based on Deterministic Finite Automaton (DFA) for detecting TCP SYN Flood and ICMP Flood attacks in a virtualized network environment. The research method used is an experimental approach through network simulation using Graphical Network Simulator 3 (GNS3). The system utilizes MikroTik Router with TaZmen Sniffer Protocol (TZSP) for packet mirroring, Python with Scapy library for packet analysis and DFA state transition processing, Supabase as a real-time cloud database, and ReactJS as a monitoring dashboard interface. The DFA model classifies network conditions into three states, namely normal, alert, and intrusion, based on the number of packets received per second. The experimental results show that the proposed IDS successfully detects abnormal traffic generated by TCP SYN Flood and ICMP Flood attacks and performs state transitions accurately in real-time. In addition, the integration of Supabase and ReactJS enables the monitoring dashboard to display network security status dynamically without requiring manual page refresh. The implementation of a decoupled architecture also improves system flexibility, responsiveness, and scalability. The findings indicate that the proposed system can provide a lightweight and interactive real-time intrusion monitoring solution for computer network security environments.*
Keywords: *Decoupled Architecture; Deterministic Finite Automaton; Intrusion Detection System; Network Security; Real-Time Monitoring*

Abstrak. Perkembangan teknologi jaringan komputer yang semakin pesat telah meningkatkan efisiensi komunikasi, namun juga menimbulkan ancaman keamanan siber yang lebih kompleks, khususnya serangan Denial of Service (DoS) seperti TCP SYN Flood dan ICMP Flood. Intrusion Detection System (IDS) konvensional umumnya masih bergantung pada analisis log statis dan belum mampu memberikan visualisasi pemantauan secara responsif dalam waktu nyata. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem deteksi intrusi berbasis Deterministic Finite Automaton (DFA) untuk mendeteksi serangan TCP SYN Flood dan ICMP Flood pada lingkungan jaringan tervirtualisasi. Metode penelitian yang digunakan adalah metode eksperimen dengan pendekatan simulasi jaringan menggunakan Graphical Network Simulator 3 (GNS3). Sistem memanfaatkan Router MikroTik dengan protokol TaZmen Sniffer Protocol (TZSP) untuk proses port mirroring, Python dengan library Scapy untuk analisis paket dan pemrosesan transisi state DFA, Supabase sebagai basis data cloud real-time, serta ReactJS sebagai antarmuka dashboard pemantauan. Model DFA mengklasifikasikan kondisi jaringan ke dalam tiga state, yaitu normal, waspada, dan intrusi berdasarkan jumlah paket yang diterima setiap detik. Hasil pengujian menunjukkan bahwa sistem IDS yang dikembangkan mampu mendeteksi trafik anomali yang dihasilkan oleh serangan TCP SYN Flood dan ICMP Flood serta melakukan transisi state secara akurat dan real-time. Integrasi Supabase dan ReactJS juga memungkinkan dashboard menampilkan status keamanan jaringan secara dinamis tanpa memerlukan refresh halaman secara manual. Penerapan arsitektur decoupled memberikan peningkatan fleksibilitas, responsibilitas, dan skalabilitas sistem. Temuan penelitian ini menunjukkan bahwa sistem yang diusulkan mampu menjadi solusi monitoring intrusi jaringan yang ringan, interaktif, dan responsif pada lingkungan keamanan jaringan komputer.
Kata kunci: Arsitektur Decoupled; Deterministic Finite Automata; Keamanan Jaringan; Monitoring Real-Time; Sistem Deteksi Intrusi

Diterima: 06 Juni 2026; Direvisi: 20 Juli 2026; Disetujui: 05 September 2026; Tersedia Online: 21 September 2026; Diterbitkan: 06 Oktober 2026.

1. LATAR BELAKANG

Perkembangan teknologi jaringan komputer dan internet telah meningkatkan efisiensi komunikasi dan pertukaran data di berbagai bidang. Namun, perkembangan tersebut juga diiringi dengan meningkatnya ancaman keamanan jaringan, khususnya serangan Denial of Service (DoS) seperti TCP SYN Flood dan ICMP Flood yang dapat melumpuhkan layanan jaringan dengan membanjiri target menggunakan paket data dalam jumlah besar. Kondisi ini menyebabkan penurunan performa jaringan bahkan mengakibatkan layanan tidak dapat diakses oleh pengguna yang sah. Oleh karena itu, diperlukan mekanisme keamanan yang mampu memantau lalu lintas jaringan dan mendeteksi aktivitas mencurigakan secara dini.

Salah satu metode yang digunakan dalam menjaga keamanan jaringan adalah Intrusion Detection System (IDS). IDS berfungsi mendeteksi pola serangan maupun aktivitas abnormal pada jaringan sehingga administrator dapat segera melakukan penanganan. Penelitian Purba dan Efendi (2020) menunjukkan bahwa implementasi IDS menggunakan Snort mampu membantu mendeteksi aktivitas mencurigakan secara otomatis. Suhendi dan Cahyo (2021) juga membuktikan bahwa penerapan Snort sebagai Intrusion Prevention System (IPS) efektif dalam meningkatkan keamanan jaringan melalui deteksi serangan secara real-time. Selain itu, Satwika, Suryanto, dan Nugroho (2020) menerapkan IDS berbasis Snort pada Raspberry Pi untuk mendeteksi serangan DoS dan memberikan notifikasi kepada administrator. Penelitian lain oleh Purnama (2023) mengintegrasikan IDS dengan media notifikasi WhatsApp dan Telegram, sedangkan Mulyanto dkk. (2024) menunjukkan bahwa kombinasi IDS dan firewall mampu meningkatkan perlindungan jaringan terhadap ancaman siber.

Pendekatan logika formal seperti Deterministic Finite Automaton (DFA) dinilai mampu memodelkan perubahan kondisi sistem secara terstruktur. Menurut Suprapto, Dandy, dan Fauziah (2020), DFA memiliki kemampuan merepresentasikan perpindahan state secara pasti berdasarkan input tertentu sehingga sesuai diterapkan pada sistem keamanan jaringan.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan pengembangan Intrusion Detection System berbasis Deterministic Finite Automaton (DFA). Pendekatan DFA digunakan untuk memetakan kondisi jaringan ke dalam tiga state utama, yaitu Normal (q_0), Waspada (q_1), dan Intrusi (q_2) berdasarkan jumlah paket per detik yang

diterima sistem. Untuk mendukung monitoring real-time, sistem dikembangkan menggunakan arsitektur decoupled dengan memisahkan modul deteksi berbasis Python dan dashboard berbasis React. Paket jaringan dari simulator GNS3 diproses melalui protokol TZSP, kemudian hasil deteksi dikirim ke Supabase sebagai media sinkronisasi data real-time. Kebaruan penelitian ini terletak pada integrasi logika DFA dengan dashboard monitoring real-time berbasis React dan Supabase sehingga menghasilkan sistem IDS yang lebih ringan, responsif, dan interaktif dalam mendeteksi serangan TCP SYN Flood dan ICMP Flood.

2. KAJIAN TEORITIS

Intrusion Detection System (IDS) merupakan sistem keamanan jaringan yang berfungsi mendeteksi aktivitas mencurigakan maupun serangan pada jaringan komputer. IDS bekerja dengan memantau lalu lintas jaringan dan menganalisis paket data untuk menemukan pola serangan atau aktivitas abnormal. Menurut Purba dan Efendi (2020), IDS membantu administrator dalam mengidentifikasi ancaman keamanan secara lebih cepat melalui proses monitoring otomatis. IDS dibedakan menjadi Network-based Intrusion Detection System (NIDS) dan Host-based Intrusion Detection System (HIDS). Suhendi dan Cahyo (2021) menjelaskan bahwa IDS berbasis jaringan mampu meningkatkan keamanan sistem melalui analisis paket secara real-time. Selain itu, Satwika, Suryanto, dan Nugroho (2020) menyatakan bahwa IDS efektif mendeteksi serangan Denial of Service (DoS) karena mampu mengenali lonjakan trafik yang tidak normal.

Deterministic Finite Automaton (DFA) merupakan model komputasi matematika yang digunakan untuk merepresentasikan perubahan state secara deterministik berdasarkan input tertentu. Menurut Suprapto, Dandy, dan Fauziah (2020), DFA memiliki karakteristik sederhana dan terstruktur sehingga sesuai diterapkan pada sistem yang membutuhkan pengambilan keputusan secara pasti. Pada penelitian ini, DFA digunakan untuk menentukan kondisi jaringan ke dalam tiga state, yaitu normal (q_0), waspada (q_1), dan intrusi (q_2). Ketika jumlah paket meningkat melewati ambang batas normal, sistem akan berpindah dari q_0 menuju q_1, kemudian ke q_2 apabila trafik memenuhi kriteria serangan. Menurut Alfaeni dkk. (2023), pendekatan state machine sesuai digunakan pada sistem keamanan jaringan karena mampu merepresentasikan perubahan kondisi secara jelas.

Penelitian ini menggunakan serangan TCP SYN Flood dan ICMP Echo Flood sebagai skenario pengujian. TCP SYN Flood memanfaatkan kelemahan proses Three-Way Handshake dengan mengirimkan paket SYN dalam jumlah besar sehingga server mengalami banyak koneksi setengah terbuka. Menurut Mulyanto dkk. (2024), serangan ini dapat meningkatkan penggunaan CPU dan memori server secara signifikan. Sementara itu, ICMP Echo Flood dilakukan dengan membanjiri target menggunakan paket ICMP Echo Request sehingga bandwidth dan sumber daya jaringan menjadi terbebani. Menurut Purnama (2023), serangan ICMP Flood dapat menyebabkan penurunan performa layanan jaringan.

TaZmen Sniffer Protocol (TZSP) merupakan protokol yang digunakan untuk meneruskan salinan paket hasil monitoring menuju sistem IDS melalui UDP port 37008. Router melakukan proses port mirroring dan mengirimkan paket menuju sistem pemantau tanpa mengganggu jalur komunikasi utama jaringan. Menurut Rahman dan Wijaya (2022), teknik port mirroring memungkinkan monitoring dilakukan secara pasif sehingga tidak mempengaruhi performa jaringan.

Arsitektur decoupled digunakan dengan memisahkan backend dan frontend agar sistem lebih fleksibel dan responsif. Menurut Nugraha dan Kurniawan (2021), pendekatan ini mempermudah pengembangan dan sinkronisasi data real-time. Backend pada penelitian ini dibangun menggunakan Python untuk pemrosesan paket dan logika DFA, sedangkan frontend menggunakan React sebagai dashboard monitoring. Kedua komponen dihubungkan menggunakan Supabase sebagai basis data cloud dan message broker. Supabase memanfaatkan mekanisme publish-subscribe dan WebSocket untuk sinkronisasi data secara real-time. Menurut Hidayat dan Saputra (2023), penggunaan WebSocket mampu mengurangi latensi pertukaran data dibandingkan metode polling konvensional.

Beberapa penelitian sebelumnya telah membahas implementasi IDS menggunakan Snort maupun sistem notifikasi keamanan jaringan. Purba dan Efendi (2020) meneliti IDS berbasis Snort untuk mendeteksi aktivitas mencurigakan, sedangkan Suhendi dan Cahyo (2021) mengembangkan Snort sebagai Intrusion Prevention System (IPS). Satwika, Suryanto, dan Nugroho (2020) menerapkan IDS pada Raspberry Pi dengan notifikasi SMS, sementara Purnama (2023) mengintegrasikan notifikasi WhatsApp dan Telegram. Penelitian Suprapto, Dandy, dan Fauziah (2020) menunjukkan bahwa DFA mampu

memodelkan perubahan state secara konsisten, namun belum diterapkan pada sistem keamanan jaringan. Berdasarkan penelitian tersebut, belum terdapat integrasi DFA dengan dashboard monitoring real-time berbasis React dan Supabase menggunakan arsitektur decoupled. Oleh karena itu, penelitian ini dikembangkan untuk menghasilkan sistem IDS yang lebih ringan, responsif, dan interaktif.

3. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimen dengan pendekatan simulasi jaringan komputer untuk menguji kemampuan Intrusion Detection System (IDS) berbasis Deterministic Finite Automaton (DFA) dalam mendeteksi serangan jaringan secara real-time (Setiawan & Pratama, 2022). Lingkungan pengujian dibangun menggunakan Graphical Network Simulator 3 (GNS3) sehingga seluruh proses penelitian dapat dilakukan secara virtual tanpa perangkat jaringan fisik (Rahman & Hidayat, 2021).

Skenario penelitian terdiri atas Router MikroTik, host korban, host penyerang, dan sistem IDS. Router MikroTik berfungsi sebagai gerbang jaringan sekaligus melakukan port mirroring menggunakan TaZmen Sniffer Protocol (TZSP) (Saputra & Nugroho, 2023). Host korban direpresentasikan menggunakan VPCS, sedangkan Kali Linux digunakan sebagai attacker yang menjalankan tools *hping3* untuk melakukan serangan TCP SYN Flood dan ICMP Flood (Firmansyah dkk., 2022).

Sistem IDS ditempatkan secara pasif dan hanya melakukan penyadapan paket (*sniffing*) terhadap salinan trafik dari router menggunakan protokol TZSP (Prakoso & Aditya, 2021). Paket yang diterima dianalisis menggunakan logika DFA untuk menentukan kondisi jaringan berdasarkan jumlah paket per detik. Hasil klasifikasi dikirim ke Supabase dan ditampilkan pada dashboard ReactJS secara real-time (Wijaya & Kurniawan, 2024).

Perangkat lunak yang digunakan meliputi GNS3, Router MikroTik CHR, Kali Linux, VPCS, dan Parrot OS sebagai mesin IDS. Sistem dikembangkan menggunakan Python dengan library Scapy untuk melakukan packet sniffing dan analisis paket jaringan (Prasetyo & Wahyudi, 2022). Dashboard monitoring dibangun menggunakan ReactJS dan Supabase sebagai media sinkronisasi data real-time (Sari & Maulana, 2023).

Tahapan penelitian dimulai dari konfigurasi topologi jaringan, pengaturan alamat IP, dan routing pada Router MikroTik (Rahman & Hidayat, 2021). Selanjutnya dilakukan

konfigurasi Packet Sniffer MikroTik untuk mengirimkan salinan trafik menuju mesin IDS melalui UDP port 37008 menggunakan protokol TZSP (Saputra & Nugroho, 2023).

Implementasi logika DFA menggunakan Python dirancang memiliki tiga state utama, yaitu q_0 sebagai kondisi normal, q_1 sebagai kondisi waspada, dan q_2 sebagai kondisi intrusi. Perubahan state dilakukan berdasarkan jumlah paket per detik yang diterima sistem (Setiawan & Pratama, 2022). Selain itu, digunakan thread latar belakang untuk mengembalikan sistem ke state normal ketika trafik kembali stabil (Herlambang & Siregar, 2023).

Pengujian dilakukan dengan menjalankan serangan TCP SYN Flood dan ICMP Flood menggunakan *hping3* dari Kali Linux menuju host korban (Firmansyah dkk., 2022). Sistem IDS kemudian mencatat perubahan state DFA dari $q_0 \rightarrow q_1 \rightarrow q_2$ dan mengirimkan hasil monitoring ke Supabase untuk ditampilkan pada dashboard ReactJS tanpa refresh halaman (Sari & Maulana, 2023).

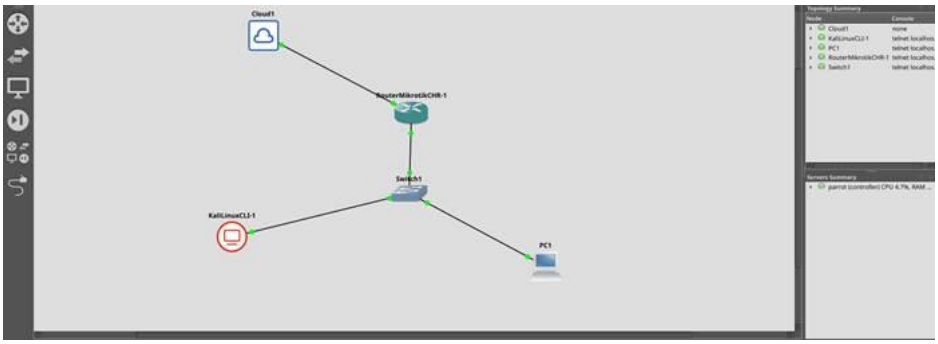
Teknik pengumpulan data dilakukan melalui observasi dan eksperimen langsung pada lingkungan simulasi jaringan (Nasution & Fauzi, 2021). Data yang dikumpulkan berupa jumlah paket jaringan, perubahan state DFA, log serangan, dan respons dashboard monitoring. Teknik analisis data dilakukan secara deskriptif dengan membandingkan kondisi jaringan normal dan kondisi saat terjadi serangan untuk mengevaluasi kemampuan sistem IDS dalam mendeteksi intrusi secara real-time (Wijaya & Kurniawan, 2024).

4. HASIL DAN PEMBAHASAN

Realisasi Topologi Infrastruktur Jaringan dan Konfigurasi Sistem

Implementasi dan pengujian sistem deteksi intrusi berbasis Deterministic Finite Automaton (DFA) pada penelitian ini dilakukan dalam lingkungan jaringan tervirtualisasi menggunakan perangkat lunak GNS3. Skenario jaringan dirancang untuk merepresentasikan serangan langsung terhadap infrastruktur inti, di mana peretas menargetkan router untuk melumpuhkan layanan jaringan secara keseluruhan.

Topologi jaringan yang dibangun terdiri dari empat entitas utama yakni, Router MikroTik yang bertindak sebagai gerbang jaringan utama sekaligus target serangan, node Kali Linux sebagai host penyerang (attacker), Virtual PC Simulator (VPCS) sebagai client internal, dan node Parrot OS sebagai mesin Intrusion Detection System (IDS) tempat logika DFA berjalan.



Gambar 1. Topologi Jaringan Simulasi pada GNS3

Untuk memastikan konektivitas antar node berjalan dengan baik dan mesin IDS dapat menerima data streaming dengan tepat, dilakukan konfigurasi pengalamatan IP (IP Addressing) secara statis. Berikut menguraikan detail konfigurasi pada masing-masing perangkat.

Tabel 1. Perbandingan Nilai Fitur Sebelum dan Setelah Normalisasi

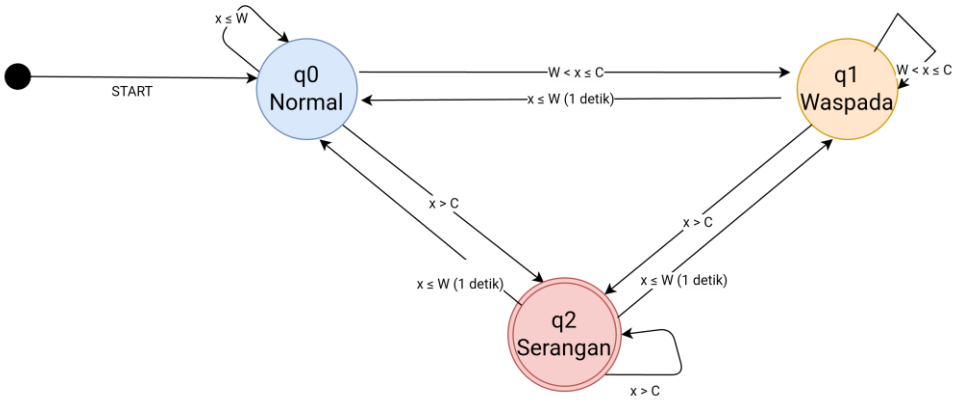
Nama Perangkat	Peran Perangkat	Antarmuka (Interface)	IP Address / Prefix
Router Mikrotik CHR-1	*Gateway* & Target Korban	ether1 (arah Cloud1 / IDS)	192.168.99.1
		ether2 (arah Switch1)	192.168.98.1
KaliLinux CLI-1	Penyerang (*Attacker*)	eth0	192.168.98.8
PC1	Klien Internal	eth0	192.168.98.2
Cloud1 (Parrot OS)	Mesin IDS (DFA Python)	tap0t	192.168.99.2

Mekanisme akuisisi data pada sistem deteksi ini adalah pemanfaatan protokol TaZmen Sniffer Protocol (TZSP). Router MikroTik dikonfigurasi menggunakan fitur Packet Sniffer untuk menangkap seluruh paket data anomali yang masuk dari arah Switch1 (antarmuka LAN).

Implementasi Logika *State Machine* DFA untuk Deteksi Intrusi

Mekanisme utama dalam mendeteksi anomali lalu lintas jaringan pada sistem ini dibangun menggunakan model matematika Deterministic Finite Automaton (DFA).

DFA dipilih karena sifatnya yang deterministik, di mana sistem hanya dapat berada pada satu status (state) tertentu pada satu waktu, dan transisi antar status ditentukan secara mutlak oleh jumlah paket masuk (sebagai input). Dalam perancangan ini, himpunan status didefinisikan menjadi tiga kondisi utama:



Gambar 2. Diagram Transisi State Logika DFA

A. **q_0 (Normal):** Status awal sistem. Kondisi jaringan dianggap aman dengan intensitas lalu lintas di bawah ambang batas kewaspadaan.

B. **q_1 (Waspada):** Status transisi awal ketika sistem mendeteksi adanya lonjakan trafik yang mencurigakan, namun belum cukup masif untuk diklasifikasikan sebagai serangan.

C. **q_2 (Intrusi):** Status kritis yang menandakan bahwa serangan *Denial of Service* (*TCP SYN Flood* atau *ICMP Flood*) telah terkonfirmasi.

Antarmuka *Dashboard* Pemantauan

Sebagai hasil akhir dari sistem deteksi intrusi, dibangun sebuah antarmuka berbasis web. Dashboard ini dirancang untuk memberikan kesadaran situasional (situational awareness) secara instan kepada administrator jaringan mengenai status keamanan infrastruktur yang dipantau.

Antarmuka dashboard terbagi menjadi dua komponen visual utama: representasi grafis state machine DFA dan tabel rekam jejak (log) serangan.



Gambar 3. *Antarmuka Dashboard Pemantauan IDS*

Visualisasi menggunakan reduksi dimensi PCA menunjukkan bahwa sebagian besar cluster dapat terpisah dengan cukup baik di ruang dua dimensi. Beberapa cluster terlihat membentuk kelompok yang kompak, sementara cluster lainnya memiliki penyebaran yang lebih luas, yang mengindikasikan variasi internal yang lebih tinggi dalam cluster tersebut.

Fungsi utama dari dashboard pemantauan ini adalah menerjemahkan data mentah berupa log ancaman menjadi informasi visual yang intuitif. Pada antarmuka komponen DFA, setiap status direpresentasikan melalui simpul (node) yang memiliki indikator warna dinamis. Perubahan warna ini beroperasi sebagai sistem peringatan dini (early warning system) dengan skenario visual sebagai berikut:

A. Status $\mathbf{q_0}$ (Normal - Indikator Hijau)**,** ketika jaringan berada dalam kondisi aman, sistem berada pada status $\mathbf{q_0}$. Indikator ini tidak hanya muncul saat sistem pertama kali dihidupkan, tetapi juga saat serangan telah benar-benar berhenti.

B. Status $\mathbf{q_1}$ (Waspada - Indikator Kuning)**,** Apabila mesin IDS mendeteksi adanya lonjakan paket (TCP SYN atau ICMP) yang melewati batas kewaspadaan namun belum mencapai titik kritis serangan masif

C. Status $\mathbf{q_2}$ (Intrusi - Indikator Merah)**,** kondisi paling kritis terjadi ketika intensitas paket serangan tinggi, memaksa sistem masuk ke status $\mathbf{q_2}$. Representasi merah ini bertindak sebagai konfirmasi visual bahwa serangan *Denial of Service* (*Flood*) sedang aktif terjadi dan menargetkan gerbang jaringan, menuntut intervensi keamanan sesegera mungkin.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa sistem Intrusion Detection System (IDS) berbasis Deterministic Finite Automaton (DFA) berhasil dirancang dan diimplementasikan untuk mendeteksi serangan TCP SYN Flood dan ICMP Flood secara real-time pada lingkungan simulasi jaringan berbasis GNS3. Sistem mampu melakukan klasifikasi kondisi jaringan ke dalam tiga state utama, yaitu normal (q0), waspada (q1), dan intrusi (q2) berdasarkan jumlah paket yang diterima. Implementasi protokol TZSP pada Router MikroTik dan pemanfaatan Python berbasis Scapy berhasil mendukung proses monitoring lalu lintas jaringan secara pasif tanpa mengganggu komunikasi utama jaringan. Selain itu, integrasi Supabase dan dashboard

berbasis ReactJS mampu menampilkan perubahan status keamanan jaringan secara real-time tanpa memerlukan refresh halaman sehingga meningkatkan efektivitas pemantauan oleh administrator jaringan. Penelitian ini juga menunjukkan bahwa penerapan arsitektur decoupled memberikan sistem yang lebih fleksibel, ringan, dan responsif dibandingkan IDS konvensional yang masih bergantung pada tampilan log statis. Meskipun demikian, penelitian ini masih memiliki keterbatasan karena pengujian hanya dilakukan pada lingkungan simulasi virtual dan terbatas pada serangan TCP SYN Flood serta ICMP Flood. Oleh karena itu, penelitian selanjutnya disarankan untuk mengimplementasikan sistem pada jaringan nyata, menambahkan variasi jenis serangan lainnya, serta mengembangkan metode deteksi yang lebih adaptif menggunakan pendekatan machine learning atau artificial intelligence agar kemampuan deteksi sistem dapat ditingkatkan secara lebih optimal.

DAFTAR REFERENSI

Alfaeni, Z. D., dkk. (2023). Deteksi serangan DDoS pada jaringan RT/RW-Net menggunakan metode Intrusion Detection System. *Seminar Nasional Teknologi Informasi*.

Firmansyah, R., Hadi, S., & Prabowo, A. (2022). Analisis serangan TCP SYN Flood menggunakan Kali Linux pada jaringan virtual. *Jurnal Teknologi Informatika dan Komputer*, 8(2), 88–96.

Herlambang, D., & Siregar, M. (2023). Implementasi packet sniffing menggunakan Python Scapy pada sistem monitoring jaringan. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 5(1), 44–52.

Hidayat, A., & Saputra, R. (2023). Implementasi WebSocket pada sistem monitoring real-time berbasis web. *Jurnal Teknologi Informasi dan Komputer*.

Mulyanto, Y., Susanto, E. S., Akbar, M. I., & Idifitriani, F. (2024). Analisis keamanan jaringan komputer menggunakan metode Intrusion Detection System (IDS) dan firewall. *Digital Transformation Technology, 3*(2), 864–870. https://doi.org/10.47709/digitech.v3i2.3402

Nasution, A., & Fauzi, R. (2021). Metode eksperimen pada simulasi keamanan jaringan komputer berbasis virtual. *Jurnal Sistem Informasi dan Teknologi*, 3(2), 73–81.

Nugraha, D., & Kurniawan, F. (2021). Implementasi arsitektur decoupled pada aplikasi web modern. *Jurnal Informatika dan Sistem Informasi*.

Prakoso, B., & Aditya, R. (2021). Analisis teknik sniffing pasif pada monitoring keamanan jaringan komputer. *Jurnal Ilmiah Informatika*, 6(1), 55–63.

Prasetyo, E., & Wahyudi, T. (2022). Implementasi Python dan Scapy untuk analisis paket jaringan secara real-time. *Jurnal Teknologi dan Sistem Komputer*, 10(3), 120–128.

Purnama, T. (2023). Implementasi Intrusion Detection System (IDS) Snort sebagai sistem keamanan menggunakan WhatsApp dan Telegram sebagai media notifikasi. *Jurnal Teknologi Informasi dan Komunikasi Profesional*.

Purba, W. W., & Efendi, R. (2020). Perancangan dan analisis sistem keamanan jaringan komputer menggunakan SNORT. *Aiti: Jurnal Teknologi Informasi, 17*(2), 143–158. https://doi.org/10.24246/aiti.v17i2.143-158

Rahman, I., & Hidayat, N. (2021). Simulasi jaringan komputer menggunakan Graphical Network Simulator 3 (GNS3). *Jurnal Informatika Mulawarman, 16*(2), 89–97.

Rahman, M., & Wijaya, A. (2022). Analisis penggunaan port mirroring untuk monitoring jaringan komputer. *Jurnal Sistem dan Teknologi Informasi*.

Saputra, D., & Nugroho, A. (2023). Implementasi port mirroring dan TZSP pada MikroTik untuk monitoring jaringan. *Jurnal Teknologi Informasi dan Komunikasi*, 11(1), 15–23.

Sari, P., & Maulana, Y. (2023). Pengembangan dashboard monitoring real-time berbasis ReactJS. *Jurnal Media Informatika Budidarma, 7*(4), 1801–1810.

Satwika, I., Suryanto, & Nugroho, A. (2020). Intrusion Detection System (IDS) menggunakan Raspberry Pi berbasis Snort dengan notifikasi SMS. *SCAN: Jurnal Teknologi Informasi dan Komunikasi*.

Setiawan, A., & Pratama, R. (2022). Implementasi Intrusion Detection System berbasis state machine pada keamanan jaringan komputer. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 6*(3), 512–519.

Suhendi, H., & Cahyo, W. D. (2021). Perancangan dan implementasi keamanan jaringan menggunakan Snort sebagai Intrusion Prevention System (IPS) pada jaringan internet STEI ITB. *NARATIF: Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika, 3*(2), 60–68.

Suprapto, A., Dandy, D., & Fauziah. (2020). Implementasi Finite State Automata pada mesin abstrak DFA dan NFA berbasis Android. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi), 5*(1), 28–36.

Wijaya, F., & Kurniawan, H. (2024). Implementasi Supabase sebagai backend real-time pada aplikasi monitoring berbasis web. *Jurnal Teknologi Informasi dan Ilmu Komputer, 11*(1), 101–109.