Spectrum: Multidisciplinary Journal
Volume 2 Nomor 3, Juli 2025
e-ISSN 3032-5838

SISTEM MONITORING SERANGAN DOS DENGAN METODE INTRUSION DETECTION SYSTEM (IDS) SNORT MENGGUNAKAN APLIKASI BERBASIS PYTHON PADA SISTEM OPERASI LINUX

Heru Gunawan[1*], Ardijan Handijono[2], Ari Putra[3], Afrizal Zein[4]
Program Studi Sistem Informasi, Universitas Pamulang, Indonesia
E-mail: [1]herugunawan791@gmail.com, [2]dosen00853@unpam.ac.id,
[3]dosen02815@unpam.ac.id, [4]dosen01495@unpam.ac.id

Abstrak

Perkembangan teknologi informasi telah meningkatkan ketergantungan terhadap sistem jaringan komputer, namun juga memunculkan ancaman keamanan, salah satunya serangan Denial of Service (DoS). Serangan ini membanjiri jaringan dengan lalu lintas berlebih hingga layanan tidak tersedia. Penelitian ini bertujuan membangun sistem pemantauan serangan DoS menggunakan Snort Intrusion Detection System (IDS) pada Ubuntu Server 20.04 LTS dan aplikasi berbasis Python. Snort digunakan untuk menganalisis paket jaringan, sedangkan Python memproses log dan menampilkannya dalam dashboard interaktif menggunakan Flask. Metode yang digunakan adalah Waterfall dengan tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Hasil pengujian menunjukkan sistem mampu mendeteksi serangan DoS secara efektif, menampilkan informasi seperti IP sumber, jenis serangan, prioritas, dan tren waktu. Penggunaan CPU meningkat saat lonjakan log namun tetap dalam batas wajar. Sistem bekerja stabil tanpa kegagalan dalam pengolahan data. Sistem ini memberikan kontribusi nyata dalam pemantauan lalu lintas jaringan secara real-time dan menjadi dasar pengembangan sistem keamanan yang lebih luas.

Kata kunci: Intrusion Detection System, Snort, Serangan DoS, Python, Monitoring Jaringan

Abstract

The rapid advancement of information technology has increased reliance on computer networks, but also introduced security threats, including Denial of Service (DoS) attacks. These attacks flood networks with excessive traffic, rendering services unavailable. This study aims to develop a DoS attack monitoring system using the Snort Intrusion Detection System (IDS) on Ubuntu Server 20.04 LTS and a Python-based application. Snort analyzes network packets, while Python processes logs and presents them in an interactive dashboard using Flask. The Waterfall method is applied, consisting of requirement analysis, design, implementation, testing, and maintenance. Testing results show the system effectively detects DoS attacks, displaying information such as source IPs, attack types, priority levels, and time trends. CPU usage increases during log spikes but remains within acceptable limits. The system runs stably without data processing failures. This system provides significant value in real-time network traffic monitoring and serves as a foundation for developing more comprehensive security solutions.

Keywords: Intrusion Detection System, Snort, DoS Attack, Python, Network Monitoring

PENDAHULUAN

Pertumbuhan pesat teknologi informasi telah mendorong peningkatan ketergantungan berbagai sektor, seperti pemerintahan, perbankan, dan industri digital, terhadap sistem jaringan komputer. Namun, seiring dengan pemanfaatan jaringan yang semakin luas, risiko keamanan yang dihadapi juga semakin kompleks. Salah satu ancaman yang paling umum dan berdampak signifikan adalah serangan Denial of Service (DoS), yang bertujuan untuk melumpuhkan layanan dengan membanjiri sistem target menggunakan lalu lintas jaringan berlebih. Farhannullah dan Hardjianto [1] menunjukkan bahwa serangan DoS menyebabkan sumber daya server seperti CPU, memori, dan bandwidth mengalami kelebihan beban, sehingga layanan menjadi lambat atau tidak dapat diakses sama sekali.

Dampaknya tidak hanya mencakup kerugian finansial, tetapi juga gangguan operasional, penurunan kepercayaan pelanggan, hingga potensi kebocoran data apabila serangan tersebut dikombinasikan dengan teknik lain seperti Distributed Denial of Service (DDoS) [2]. Wang et al. [3] mencatat bahwa jumlah serangan DDoS meningkat sebesar 314% pada tahun 2021 dibandingkan tahun sebelumnya, dengan sebagian besar menargetkan sektor keuangan dan layanan cloud.

Selain itu, serangan DoS telah menyebabkan gangguan besar pada layanan perbankan online, e-commerce, dan penyedia layanan internet, yang berdampak pada kerugian finansial, kehilangan data, dan rusaknya reputasi layanan [4], [5]. Shamsi et al. [6] melaporkan bahwa perusahaan dapat mengalami kerugian hingga $1,3 juta per jam akibat layanan yang terganggu oleh serangan DoS. Zargar et al. [7] menambahkan bahwa serangan DoS sering kali menargetkan penyedia layanan digital, perbankan, dan platform e-commerce, sehingga sangat mengganggu operasional bisnis. Selain kerugian ekonomi, Kumar dan Gupta [5] menegaskan bahwa kepercayaan pengguna juga dapat menurun drastis akibat gangguan yang berulang. Kandias et al. [8] menjelaskan bahwa serangan DoS bahkan dapat digunakan sebagai pintu masuk untuk melakukan eksploitasi tambahan seperti pencurian data sensitif.

Untuk mengatasi hal tersebut, sistem keamanan jaringan yang andal dan mampu mendeteksi serangan secara efektif sangat dibutuhkan. Salah satu pendekatan yang umum digunakan adalah pemanfaatan Intrusion Detection System (IDS) seperti Snort. Namun, tantangan yang sering muncul adalah bagaimana menyajikan hasil deteksi tersebut dalam bentuk yang mudah dipahami oleh pengguna. Jin et al. [9] mengusulkan pengembangan sistem pemantauan berbasis web yang mengintegrasikan Snort dengan aplikasi Python menggunakan framework Flask untuk menampilkan data log dalam visualisasi interaktif.

Dalam penelitian ini, sistem yang dikembangkan akan memanfaatkan pendekatan real-time log analysis, di mana data serangan yang direkam oleh

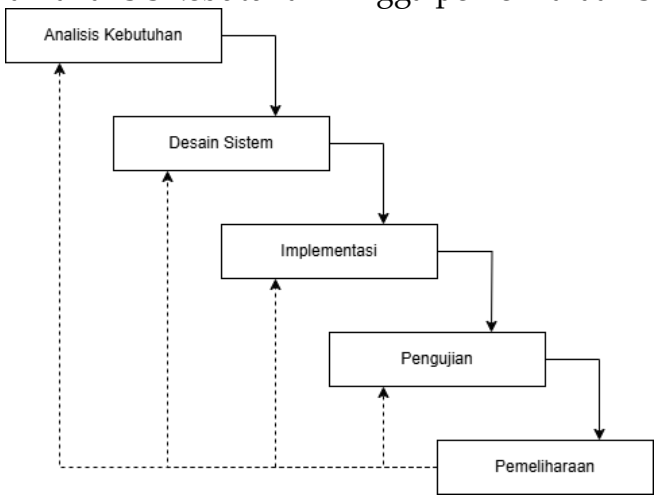
Snort akan diolah dan disajikan dalam bentuk dashboard. Pendekatan ini bertujuan memberikan wawasan mendalam terhadap aktivitas jaringan, membantu proses investigasi keamanan, dan memungkinkan tindakan preventif terhadap ancaman siber [10]. Ezenwe et al. [10] menunjukkan bahwa pemantauan serangan berbasis log secara real-time dapat meningkatkan kesadaran terhadap serangan dan memperkuat sistem terhadap potensi kerentanan.

.

METODE PENELITIAN

2.1 Model Pengembangan Sistem

Metode penelitian yang digunakan dalam pengembangan sistem monitoring serangan DoS ini adalah metode pengembangan perangkat lunak model Waterfall. Model ini dipilih karena memiliki tahapan yang jelas dan sistematis, dimulai dari analisis kebutuhan hingga pemeliharaan sistem.



Gambar 1. Metode Waterfall

2.2 Analisis Kebutuhan Sistem

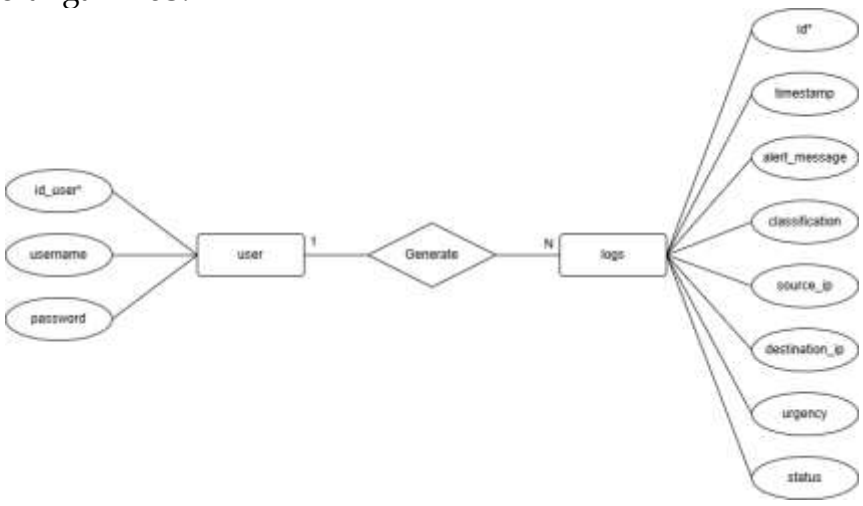
Analisis kebutuhan dilakukan untuk menentukan fitur dan komponen yang diperlukan dalam sistem. Kebutuhan ini meliputi:

1. Perangkat keras seperti laptop/komputer dengan spesifikasi minimum Intel Core i3, RAM 8 GB, dan SSD 128 GB;
2. Perangkat lunak seperti sistem operasi Linux (Ubuntu 20.04), Snort, Python 3, MySQL, dan Flask;
3. Fungsionalitas sistem yang mencakup deteksi serangan DoS, visualisasi log serangan, dan manajemen log.

2.3 Perancangan Sistem

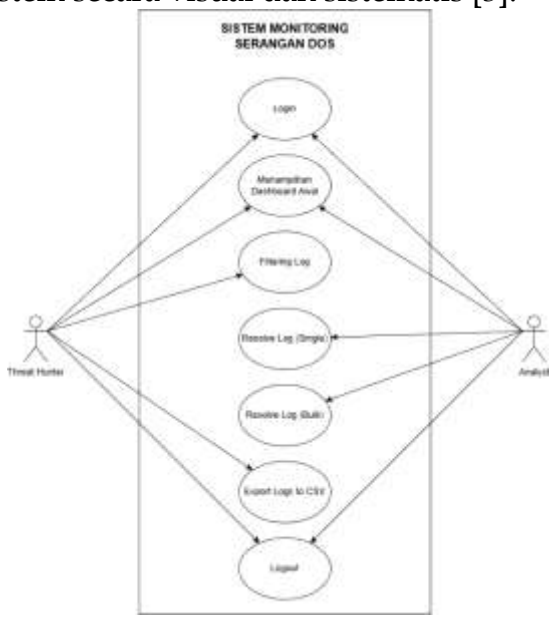
Perancangan sistem mencakup pembuatan Entity Relationship Diagram (ERD), Logical Record Structure (LRS), dan normalisasi data hingga bentuk Third Normal Form (3NF) untuk memastikan efisiensi dan integritas basis data [1]. *Gambar 2* menampilkan ERD dari sistem yang menunjukkan hubungan antara entitas User dan Logs, yang menjadi komponen utama dalam sistem

monitoring serangan DoS.



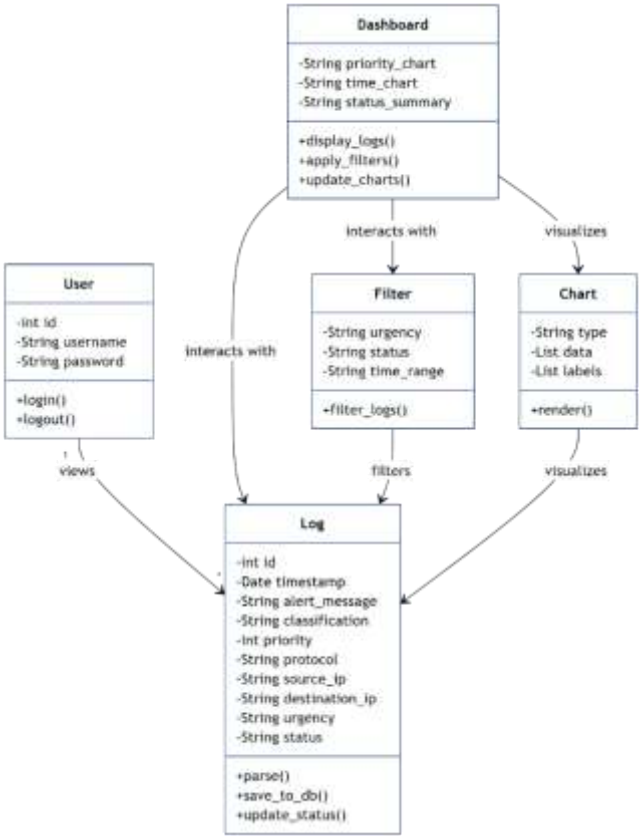
Gambar 2. Entity Relationship Diagram (ERD) Sistem Monitoring DoS

Perancangan antarmuka pengguna dilakukan dengan pendekatan Unified Modeling Language (UML), termasuk *use case diagram* dan *activity diagram*, yang digunakan untuk menggambarkan alur aktivitas serta interaksi antara pengguna dan sistem secara visual dan sistematis [3].



Gambar 3. Use Case Diagram

Selain itu, struktur tabel log dirancang untuk menyimpan informasi penting seperti waktu kejadian (*timestamp*), pesan peringatan (*alert message*), klasifikasi serangan, tingkat prioritas, protokol yang digunakan, alamat IP sumber, serta alamat IP tujuan. Proses normalisasi dilakukan untuk menghindari redundansi data dan menjaga konsistensi antar entitas [2].



Gambar 4. Class Diagram

2.4 Implementasi Sistem

Implementasi dilakukan dengan menginstal dan mengkonfigurasi Snort sebagai Intrusion Detection System (IDS) pada sistem operasi Linux untuk mendeteksi serangan DoS berbasis jaringan [6]. Antarmuka sistem dikembangkan menggunakan bahasa pemrograman Python dengan framework Flask untuk membangun aplikasi berbasis web yang ringan dan fleksibel [7]. Data log disimpan pada sistem manajemen basis data MySQL yang mendukung struktur relasional dan integrasi dengan aplikasi [4].

Dashboard monitoring mencakup beberapa fitur utama, seperti autentikasi pengguna (login), filter log berdasarkan status dan tingkat urgensi, visualisasi data serangan dalam bentuk grafik, ekspor data log ke format CSV, serta tombol aksi untuk melakukan resolusi terhadap log yang terdeteksi sebagai anomali. Fitur-fitur ini dirancang untuk mendukung proses analisis keamanan yang cepat dan efisien.

2.5 Pengujian Sistem

Pengujian dilakukan untuk memastikan bahwa sistem yang dikembangkan dapat berjalan sesuai dengan kebutuhan fungsional. Metode pengujian yang digunakan adalah black box testing, yaitu pengujian berdasarkan masukan dan keluaran tanpa memeriksa struktur internal kode program [8]. Setiap fitur diuji berdasarkan skenario penggunaan untuk mengevaluasi apakah sistem memberikan respons yang sesuai terhadap input

yang diberikan.

Hasil pengujian menunjukkan bahwa sistem berhasil mendeteksi serangan Denial of Service (DoS) secara real-time, mencatat log ke dalam basis data MySQL, serta menampilkan data pada dashboard secara akurat. Seluruh fungsi utama, termasuk login, filter log, visualisasi grafik, ekspor CSV, dan resolusi log, berjalan dengan baik sesuai dengan spesifikasi yang telah dirancang.

2.6 Pemeliharaan Sistem

Tahap pemeliharaan dilakukan untuk memastikan sistem tetap berjalan optimal setelah implementasi. Pemeliharaan meliputi perbaikan bug, pembaruan rule set pada Snort secara berkala, serta penyesuaian sistem berdasarkan kebutuhan pengguna [9]. Proses ini penting untuk menjaga akurasi deteksi terhadap pola serangan yang terus berkembang.

Pengembangan lanjutan sistem dapat dilakukan dengan menambahkan fitur notifikasi otomatis melalui email atau pesan instan, integrasi dengan platform Security Information and Event Management (SIEM), serta kemampuan untuk mendeteksi tipe serangan lainnya seperti Distributed Denial of Service (DDoS) dan port scanning. Dengan demikian, sistem dapat menjadi solusi yang lebih komprehensif dalam memantau keamanan jaringan.

HASIL DAN PEMBAHASAN

3.1 Implementasi

Tabel 1. Spesifikasi Implementasi Perangkat Keras (Hardware)

No	Komponen	Spesifikasi
1	*Processor*	Intel Core i3-10110U @ 2.10GHz (4 CPUs), 10th Gen
2	*Memory RAM*	8.00 GB DDR4 2666 MHz (Single Channel)
3	SSD	128 GB SATA III SSD (Kingston SA400S37128G)

Tabel 1 menunjukkan spesifikasi perangkat keras yang digunakan dalam penelitian ini. Perangkat yang digunakan memiliki prosesor Intel Core i3-10110U generasi ke-10 dengan kecepatan 2.10 GHz dan 4 inti pemrosesan (*CPU*), didukung oleh RAM sebesar 8 GB DDR4 berkecepatan 2666 MHz dalam konfigurasi *single channel*. Media penyimpanan yang digunakan adalah SSD SATA III sebesar 128 GB untuk mempercepat proses baca tulis data dan memperlancar kinerja sistem secara keseluruhan.

Tabel 2. Implementasi Perangkat Lunak (Software)

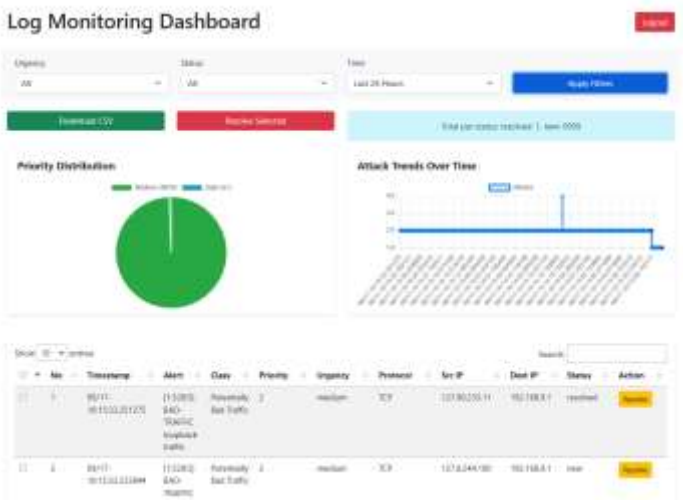
No	Perangkat Lunak	Keterangan
1	Sistem Operasi Host	*Windows 11 Pro 64-bit (Build 22631)*

2	Sistem Operasi Guest	*Ubuntu Server 20.04.6 LTS (*diinstal dalam *VirtualBox)*
3	*IDS*	*Snort 3.1.60.0*
4	Bahasa Pemrograman	*Python 3.10.12*
5	*Framework Web*	*Flask 2.3.3*
6	*Browser Web*	*Google Chrome 116 / Mozilla Firefox 115 ESR*
7	Editor Kode	*Visual Studio Code 1.102.3*

Tabel 2 memuat rincian perangkat lunak yang digunakan. Sistem operasi host menggunakan Windows 11 Pro 64-bit dengan *build* 22631, sedangkan sistem operasi *guest* menggunakan Ubuntu Server 20.04.6 LTS yang diinstal melalui aplikasi *VMware*. Untuk sistem deteksi intrusi, digunakan Snort versi 3.1.60.0, yang dikonfigurasi untuk memantau lalu lintas jaringan.

3.2 Implementasi Antarmuka (User Interface)

Sistem yang dikembangkan menyediakan antarmuka berbasis web untuk memantau serangan DoS secara real-time. Antarmuka ini menampilkan hasil analisis log Snort dalam bentuk tabel dan grafik. Dashboard memungkinkan pengguna memfilter log berdasarkan status, urgensi, dan waktu. Selain itu, tersedia fitur ekspor CSV dan penandaan status resolusi pada tiap log.



Gambar 3. Tampilan Dashboard Monitoring Log

Antarmuka dibuat menggunakan HTML, CSS, JavaScript, dan didukung oleh backend Python Flask. Tujuan dari desain ini adalah untuk menyajikan informasi secara informatif dan mudah dipahami, serta meningkatkan respons administrator terhadap insiden keamanan jaringan.

3.2 Pengujian Sistem

Tabel 3. Hasil Pengujian Unit

No	Komponen yang Diuji	Hasil
1	Snort (Deteksi DoS)	Berhasil mendeteksi serangan

2	Parser Log Python	Data berhasil ditransformasikan
3	Penyimpanan ke MySQL	Data log tersimpan dengan baik
4	Dashboard (Web Flask)	Data tampil di halaman utama
5	Fitur Export CSV	File CSV berhasil dibuat
6	Filter Log dan Status	Filter dan update status berfungsi

Tabel 3 menunjukkan hasil pengujian unit dari masing-masing komponen sistem. Seluruh fungsi utama, mulai dari deteksi serangan oleh Snort, transformasi log dengan Python, penyimpanan ke MySQL, tampilan dashboard, fitur export CSV, hingga filter dan update status log, berhasil berjalan dengan baik sesuai harapan.

3.2 Analisis Sistem

Tabel 4. Analisa Hasil Pengujian

No	Deskripsi	Hasil Yang Diharapkan	Hasil Akhir
1	Terjadinya burst traffic dari satu IP/subnet	Sistem mendeteksi dan mencatat aktivitas mencurigakan secara berulang	Sistem berhasil mencatat serangan *SYN Flood* dari IP yang memiliki class sama dalam waktu singkat
2	Serangan terjadi dalam rentang waktu tertentu	Sistem mampu merekam dan menampilkan tren waktu serangan	Grafik menunjukkan lonjakan serangan pada waktu tertentu
3	Variasi prioritas serangan dalam *Log*	Sistem menampilkan distribusi tingkat ancaman berdasarkan level prioritas *Snort*	Pie chart berhasil menggambarkan dominasi serangan dengan prioritas
4	Serangan dari beberapa IP berbeda	Sistem mengelompokkan IP berdasarkan frekuensi serangan	Teridentifikasi banyak melibatkan IP dari subnet yang sama dalam jumlah besar pada waktu singkat (burst traffic).
5	Serangan DoS berbasis protokol (SYN/*UDP Flood*)	Sistem mengenali jenis protokol dalam setiap serangan yang terdeteksi	Sistem berhasil mengklasifikasi serangan berdasarkan protokol
6	Serangan mempengaruhi performa server	Sistem tetap responsif meski terjadi lonjakan *Log*	*CPU* dan *RAM* meningkat

Berdasarkan Tabel 4, hasil analisa menunjukkan bahwa sistem mampu merespons berbagai jenis skenario serangan DoS sesuai dengan harapan. Sistem

berhasil mencatat aktivitas burst traffic dari satu IP, menampilkan tren serangan dalam rentang waktu tertentu, serta mengelompokkan serangan berdasarkan prioritas, IP sumber, dan jenis protokol. Selain itu, sistem juga mampu mendeteksi peningkatan penggunaan CPU dan RAM saat terjadi lonjakan log, yang menunjukkan sensitivitas sistem terhadap performa server saat terjadi serangan.

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, sistem monitoring serangan DoS berbasis IDS Snort dan aplikasi Python Flask pada Ubuntu Server 20.04 LTS berhasil mendeteksi serangan secara real-time, mencatat aktivitas mencurigakan, dan menyajikan data log dalam antarmuka web yang interaktif. Sistem mampu mengenali pola serangan seperti burst traffic, variasi prioritas serangan, serta serangan dari berbagai IP dengan akurasi yang baik. Visualisasi data dalam bentuk grafik dan pie chart membantu pengguna memahami tren dan distribusi tingkat ancaman. Pengujian menunjukkan bahwa sistem tetap responsif meskipun terjadi lonjakan log, dengan penggunaan CPU dan RAM yang masih dalam batas wajar untuk skala sistem minimum.

Keberhasilan ini menunjukkan bahwa integrasi Snort dengan Python Flask merupakan pendekatan efektif untuk implementasi sistem deteksi serangan ringan dan fleksibel. Meski demikian, sistem masih memiliki ruang untuk pengembangan lebih lanjut, seperti penambahan deteksi serangan lain, integrasi notifikasi otomatis, dan peningkatan efisiensi sumber daya. Pengujian di lingkungan jaringan nyata juga disarankan untuk memastikan kestabilan sistem dalam kondisi lalu lintas tinggi dan kompleks.

DAFTAR PUSTAKA

Farhannullah, M., & Hardjianto, H. (2020). Analisis dan simulasi serangan denial of service (DoS) menggunakan Metasploit Framework. *Jurnal Teknologi dan Sistem Komputer, 8*(3), 426–431.

Kumar, A., & Gupta, B. (2021). An integrated framework for detection and prevention of DDoS attacks in cloud environment. *Journal of Ambient Intelligence and Humanized Computing, 12*(3), 3287–3306.

Wang, Y., Liu, J., Zhang, X., & Chen, Z. (2021). DDoS attack detection based on CNN-BiLSTM hybrid neural network. *IEEE Access, 9*, 27365–27373.

Zargar, P., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys & Tutorials, 15*(4), 2046–2069.

Kumar, A., & Gupta, B. (2020). Threats in cloud computing: An analysis and survey. *Computer Networks, 169*, 1–22.

Shamsi, M., Anjum, N., & Mir, F. (2018). Cyber security threats in the financial sector: A case study on DoS attacks. *International Journal of Computer Applications, 179*(14), 25–30.

Kandias, S., Mylonas, A., & Gritzalis, D. (2014). Social engineering attacks exploiting the trust of users on social networks. In *Proceedings of the 6th*

IEEE International Conference on Cyber Conflict (CyCon) (pp. 1–18). Tallinn, Estonia.

Zargar, M., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against DDoS flooding attacks. *IEEE Communications Surveys & Tutorials, 15*(4), 2046–2069.

Jin, X., Hu, Y., & Wang, K. (2019). Real-time web-based network monitoring system based on Snort. In *Proceedings of the IEEE 5th International Conference on Computer and Communications (ICCC)* (pp. 1221–1226). Chengdu, China.

Ezenwe, C., Onibere, A., & Eze, M. (2021). Enhancing network security using real-time intrusion detection system (IDS). *International Journal of Computer Applications, 183*(47), 15–21