

ANALISIS PERFORMA INTRUSION DETECTION SYSTEM SNORT DAN SURICATA TERHADAP SERANGAN SQL INJECTION

Fabian Ramot Argenta Pasaribu¹, Andi Maslan²

¹Mahasiswa Program Studi Teknik Informatika, Universitas Putera Batam

²Dosen Program Studi Teknik Informatika, Universitas Putera Batam

email: pb210210016@upbatam.ac.id

ABSTRACT

Web application security is becoming increasingly important due to the rise of threats such as SQL Injection, which exploits vulnerabilities to access sensitive data. As one of the most severe types of attacks, SQL Injection compromises the confidentiality, integrity, and access control of a system. Intrusion Detection Systems such as Snort and Suricata are used to detect and mitigate this. This study compares the effectiveness of Snort and Suricata in detecting SQL Injection using an experimental setup. The vulnerable web application (DVWA) was installed on Ubuntu, while attacks were launched from Kali Linux. Both IDS tools were configured to monitor network traffic and detect intrusions based on predefined rules. Performance was evaluated using accuracy, precision, recall, and F1 score. Suricata outperformed Snort in all metrics, Suricata also demonstrated faster detection. These results indicate that Suricata is more accurate and efficient at detecting SQL injection attacks in the test environment.

Keywords: *Intrusion Detection System, Network Security, Snort, SQL Injection, Suricata.*

PENDAHULUAN

Di era digital saat ini, aplikasi *web* sangat penting untuk kehidupan sehari-hari. Banyak orang menggunakannya untuk melakukan banyak hal, seperti berbelanja *online*. Situs *web* biasanya menggunakan *database* di *backend* untuk menyimpan informasi pengguna. *Database* yang menyimpan informasi pribadi seperti kata sandi, nomor kartu kredit, dan nomor jaminan sosial sering menjadi target serangan siber. (Kareem et al., 2021). Peningkatan jumlah pengguna *website* secara signifikan membuka celah bagi pelaku serangan siber untuk mengeksploitasi kerentanan sistem, salah satunya melalui serangan *SQL Injection*.

Berdasarkan dari dokumen “Lanskap Keamanan Siber Indonesia Tahun 2024” oleh BSSN. Indonesia mencatat lebih dari 330.000.000 jumlah trafik anomali, jenis anomali terbanyak yaitu serangan *Mirai Botnet* 81 juta aktivitas. Selain itu ditemukan juga aktivitas lainnya seperti serangan *Advanced Persistent Threat* sebanyak 2,4 juta, *Ransomware* sebanyak 514 ribu, dan *Phishing* sebanyak 26 juta. BSSN menerima laporan siber dari *stakeholder*, dengan jenis insiden terbanyak adalah kebocoran data. BSSN menemukan 241 kebocoran data, serta 56 juta data yang terekspos di *darknet* dan berdampak pada *stakeholder* di Indonesia.

Serangan *Structured Query Language Injection* dianggap sebagai serangan paling berbahaya dalam kategori injeksi karena dapat mengganggu layanan keamanan utama, yaitu kerahasiaan, autentikasi, otorisasi, dan integritas (Jemal et al., 2021). *Intrusion Detection System* dan *Intrusion Prevention System* merupakan dua pendekatan penting dalam keamanan jaringan. *IDS* berfungsi sebagai untuk memantau dan memberikan peringatan terhadap aktivitas mencurigakan tanpa mengambil tindakan langsung, sedangkan *IPS* tidak hanya mendeteksi tetapi juga secara otomatis memblokir serangan (Safana Hyder Abbas et al., 2023). Penelitian ini difokuskan pada *IDS* karena tidak mengganggu lalu lintas jaringan secara langsung.

2 *IDS open source* yang paling umum digunakan adalah *Snort* dan *Suricata*. Keduanya memiliki kemampuan mendeteksi pola serangan berbasis *signature*, namun masing – masing memiliki pendekatan arsitektur dan performa yang berbeda. *Snort* dikenal sebagai *IDS* ringan berbasis *signature* yang dapat dikustomisasi, sedangkan *Suricata* hadir dengan fitur lebih fleksibel, seperti deteksi berbasis skrip dan kemampuan *multithread* yang lebih baik (Hu et al., 2020).

Metode yang digunakan dalam penelitian ini, digunakan pendekatan *testbed-based evaluation* untuk menguji performa sistem deteksi intrusi *Snort* dan *Suricata* dalam ruang lingkungan virtual. Pendekatan ini dipilih untuk menghindari permasalahan hukum, etika, dan ketersediaan yang mungkin timbul jika pengujian dilakukan pada sistem langsung (de Santana et al., 2024). Melalui pendekatan ini, peneliti bertujuan untuk mengevaluasi dan

membandingkan performa *Snort* dan *Suricata* dalam mendeteksi serangan *SQL Injection*. Aspek yang dianalisis meliputi tingkat deteksi, akurasi, dan efisiensi sumber daya.

KAJIAN TEORI

2.1 Intrusion Detection System

Intrusion Detection System adalah sebuah mekanisme keamanan jaringan yang dirancang untuk mengidentifikasi dan memantau aktivitas tidak biasa, mencurigakan, atau berpotensi berbahaya dalam sistem jaringan komputer. Sistem ini bekerja dengan cara menganalisis lalu lintas data dan memberikan peringatan secara otomatis kepada administrator jaringan jika terdeteksi adanya ancaman atau penyimpangan dari pola yang dianggap normal (Abdulganiyu et al., 2023).

2.2 Host based Intrusion Detection System

Host based Intrusion Detection System bekerja dengan memeriksa file *log* dari lalu lintas *server* berdasarkan *database* yang ada di *server*, dan akan mengirimkan laporan kepada administrator sistem terkait ketika terdeteksi serangan yang tidak terduga. Sistem ini dipasang pada server khusus dan bertugas mendeteksi atau mencegah serangan dengan memantau file konfigurasi, mencatat aktivitas sistem, memeriksa perubahan pada integritas sistem, serta mencegah penyalahgunaan. Kompatibilitas dengan sistem operasi menjadi hal penting, karena sifat sistem operasi yang berbeda dapat memengaruhi efektivitas *HIDS* (Efe & Abaci, 2022).

2.3 Snort

Snort adalah perangkat lunak *open source* yang digunakan untuk mendeteksi dan mencegah intrusi pada jaringan komputer. *Snort* memiliki kemampuan untuk menganalisis lalu lintas jaringan secara *real-time* dan melakukan *logging* paket-paket data yang mencurigakan, serta mampu mendeteksi berbagai jenis serangan seperti *buffer overflow*, *stealth port scans*, *CGI attacks*, *SMB probes*, dan *OS fingerprinting attempts*, (Barends et al., 2022).

2.4 Suricata

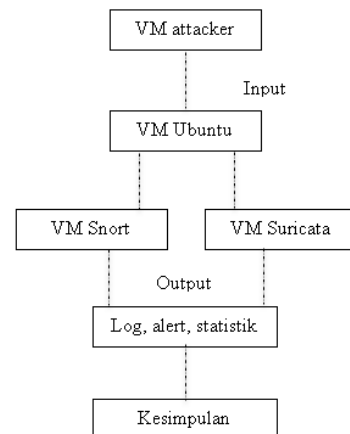
Suricata merupakan sistem berbasis aturan yang menggunakan kumpulan aturan yang dikembangkan secara eksternal untuk memantau lalu lintas jaringan dan memperingatkan sistem administrator saat terjadi peristiwa yang mencurigakan seperti paket yang dikirimkan secara terus menerus seperti indikasi serangan *Denial of Service Attack*. *Suricata* dapat mempercepat deteksi dan respons terhadap ancaman siber, sehingga memungkinkan administrator untuk lebih cepat merespon ancaman keamanan dan melindungi jaringan mereka dengan lebih efektif. *Suricata* adalah *tools* pendeteksi yang sudah mengandung *Intrusion Detection System (IDS)*, sehingga dapat menghentikan serangan dengan melakukan *drop packet* yang dicurigai. *Suricata* berbeda dari *Snort* namun *Suricata* memiliki dukungan untuk *rules* yang luas sehingga memiliki sintaks yang serupa dengan bahasa *rules Snort* (Hoover & Thompson, 2022).

2.5 SQL Injection

Menurut (Chen et al., 2021), *Structured Query Language* adalah bahasa pemrograman yang digunakan untuk *query database*. *SQL* didasarkan pada aljabar relasional dan dirancang untuk mengelola data dalam Sistem Manajemen Basis Data Relasional (*RDBMS*). Beberapa perintah *SQL* yang paling penting adalah *create*, *insert*, *update*, *delete*, *select* dan *drop*.

2.6 Kerangka Pemikiran

Kerangka pemikiran berfungsi untuk menjelaskan alur logis yang digunakan dalam upaya menyelesaikan permasalahan yang menjadi fokus penelitian.



Gambar 1. Kerangka pemikiran
(Sumber: Data Penelitian, 2025)

Berikut ini merupakan tahapan – tahapan yang terdapat dalam kerangka pemikiran:

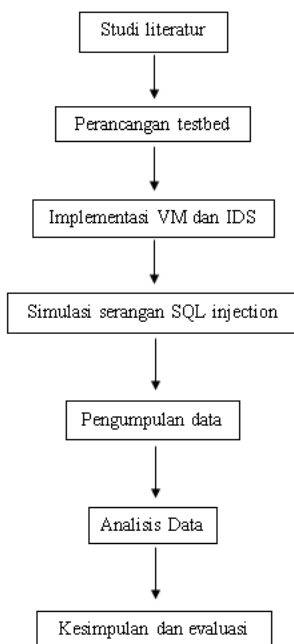
1. *VM Attacker*: bertindak sebagai sumber serangan yang mengirimkan *traffic* ke sistem target.
2. *VM Ubuntu*: berperan sebagai *server* utama yang menampung dua sistem deteksi intrusi.

3. *VM Snort* dan *VM Suricata*: kedua sistem *IDS* diimplementasikan secara paralel untuk memonitor dan menganalisis lalu lintas jaringan dari *VM Attacker*.
4. *Output*: kedua *IDS* menghasilkan *output* berupa *log*, *alert*, dan statistik.
5. Kesimpulan: *data output* dianalisis untuk menarik kesimpulan mengenai efektivitas dan performa dari kedua *IDS*.

METODE PENELITIAN

3.1 Desain Penelitian

Agar pelaksanaan penelitian berlangsung secara terstruktur, dibutuhkan suatu rancangan penelitian. Tahapan – tahapan dalam rancangan sebagai berikut:



Gambar 2. Desain Penelitian
(Sumber: Data Penelitian, 2025)

1. Studi literatur

Tahap pertama adalah studi literatur, yang bertujuan memahami topik dengan menganalisis yang ada dalam mendeteksi serangan *SQL Injection* serta untuk mengidentifikasi teori, metode, dan hasil penelitian terdahulu yang berkaitan.

2. Perancangan testbed

Tahap kedua adalah mempersiapkan perangkat keras dan lunak yang dibutuhkan untuk pelaksanaan penelitian. Dengan satu unit laptop dengan spesifikasi: *RAM 16 GB*, *SSD 240 GB*, dan prosesor *AMD Quad – Core*. Perangkat lunak *IDS Snort* dan *Suricata*, *virtual machine attacker* dan *victim*.

3. Implementasi VM dan IDS

Tahap ketiga adalah proses konfigurasi sistem sesuai kebutuhan penelitian. Tahap ini mencakup instalasi *software*, pengaturan lingkungan, serta penyesuaian parameter sistem agar mendukung metode dan alat analisis yang akan digunakan.

4. Simulasi serangan SQL Injection

Tahap keempat adalah implementasi *Virtual Machine* dan *Intrusion Detection System*, selanjutnya melakukan simulasi serangan menggunakan *SQL Injection manual* di *VM attacker*.

5. Data analisis

Tahap kelima adalah pengambilan hasil data dari sistem yang telah dikonfigurasi sebelumnya, kemudian dilanjutkan dengan proses analisis data. Analisis dilakukan untuk menginterpretasikan hasil yang diperoleh, mengevaluasi kinerja sistem, serta menarik kesimpulan berdasarkan tujuan penelitian.

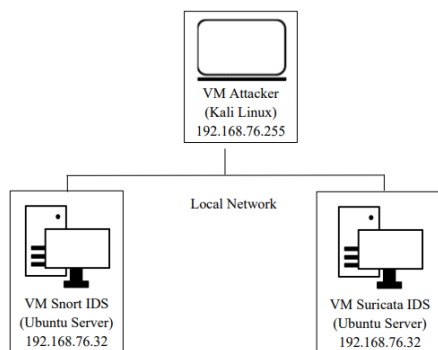
6. Kesimpulan dan evaluasi

Tahap keenam adalah membahas hasil performa dari kedua *IDS* yaitu *Snort* dan *Suricata* dalam mendeteksi serangan

SQL Injection, serta evaluasi untuk peneliti kedepannya.

3.2 Topologi jaringan lokal

Simulasi ini terdiri dari tiga komponen utama: *VM Attacker (Kali Linux)* sebagai pihak penyerang, *VM Snort IDS (Ubuntu Server)* dan *VM Suricata IDS (Ubuntu Server)* sebagai sistem deteksi intruksi, yang terhubung dalam jaringan internal untuk menguji efektivitas kedua *IDS* tersebut.



Gambar 3. Topologi
(Sumber: Data Penelitian, 2025)

Topologi yang digunakan dalam proses implementasi mencakup *Virtual Machine (VM) attacker* dan *VM victim* yang saling terhubung dalam jaringan lokal. Sistem *IDS* dioperasikan melalui software *VM* dengan sistem operasi *Ubuntu*. Pada *VM attacker* akan diinstal *Kali Linux*, dan serangan dilakukan secara manual menggunakan Teknik *SQL Injection* manual. Target serangan adalah situs web *DVWA*, yang dikonfigurasi dengan aturan (rules) tertentu.

3.3 Metode pengumpulan data

Metode pengumpulan data penelitian ini menggunakan metode eksperimen, dengan menggunakan deteksi intrusi, yaitu *Snort* dan *Suricata*. Kedua sistem ini

digunakan untuk memantau lalu lintas jaringan dan menangkap berbagai jenis data seperti *packet data*, *log* aktivitas, serta *alert* terhadap potensi ancaman. Penelitian ini menggunakan *ground truth* sebagai acuan kebenaran dalam mengklasifikasikan hasil deteksi. *Ground truth* diperoleh melalui proses klasifikasi manual terhadap 30 jenis paket data yang digunakan dalam pengujian. Setiap paket dianalisis dan diberi label berdasarkan apakah paket tersebut mengandung serangan *SQL Injection* atau tidak. Dengan adanya *ground truth* ini, setiap alert yang dihasilkan oleh sistem *IDS* dapat diklasifikasikan ke dalam empat kategori utama, yaitu *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*, dan *True Negative (TN)*, sehingga memungkinkan perhitungan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* secara objektif dan terukur.

3.4 Kebutuhan perangkat keras

Berikut merupakan rincian perangkat keras yang digunakan dalam penelitian ini:

Tabel 1. Kebutuhan perangkat keras

Komponen	Spesifikasi minimum
Prosesor	AMD A12 – 9720P Radeon R7, 12 Compute Cores 4C + 8G 2,70GHz
RAM	16 GB
Penyimpanan	SSD 256 GB
Jaringan	Koneksi lokal / internal VM

3.5 Kebutuhan perangkat lunak

Berikut merupakan daftar kebutuhan perangkat lunak yang digunakan dalam penelitian ini:

Tabel 2. Kebutuhan perangkat lunak

Sistem operasi	Aplikasi dan tool
----------------	-------------------

Ubuntu server	Snort 2.9
---------------	-----------

20.04 LTS	
-----------	--

Kali linux 2023.1	Suricata 6.0
-------------------	--------------

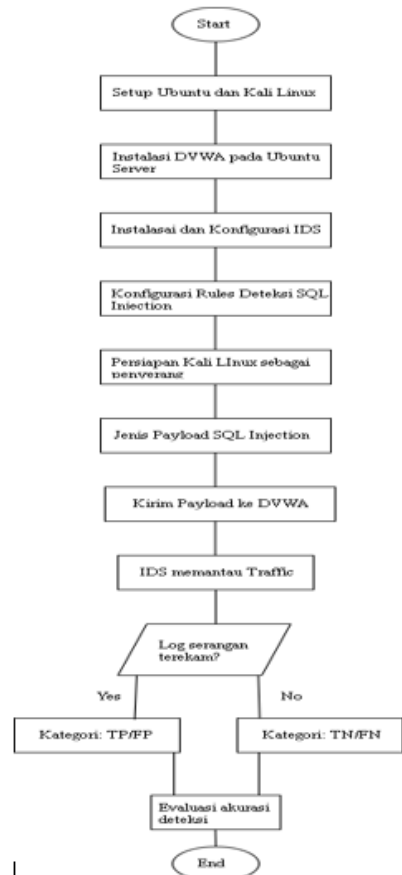
VirtualMachine 7.0	DVWA 1.10
--------------------	-----------

3.6 Kebutuhan jaringan

Semua *Virtual Machine* dikonfigurasi dalam satu jaringan *virtual internal* (*internal network/host-only network*) sehingga komunikasi antar perangkat dapat dipantau oleh *IDS*. *IDS* dapat diletakkan secara *inline* atau menerima salinan *traffic* menggunakan *mode port mirroring*.

3.7 Metode Simulasi Serangan SQL Injection

Dalam metode simulasi serangan *SQL Injection* menggunakan *setup Testbed* yang telah diatur, menyertakan perbedaan terminologi, fokus, dan detail teknis dalam simulasi serangan *SQL Injection* dengan *IDS*. Perubahan ini bertujuan untuk meningkatkan akurasi dan isolasi lingkungan pengujian.



Gambar 4. Metode serangan *SQL Injection*

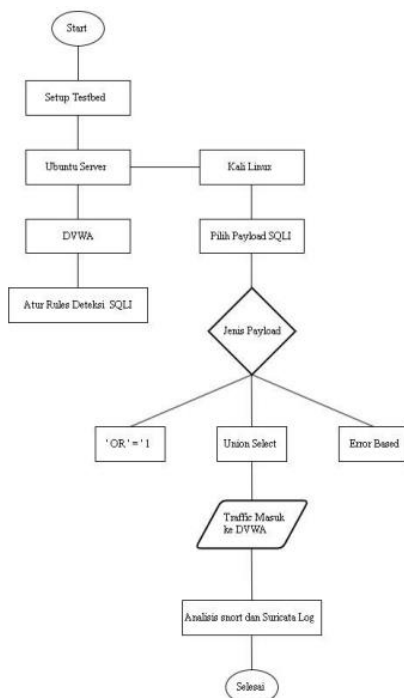
(Sumber: Data Penelitian, 2025)

Metode simulasi serangan *SQL Injection* dilakukan dengan menyiapkan dua mesin *virtual*, *Ubuntu server* yang menjalankan *DVWA* sebagai target, dan *Kali Linux* sebagai penyerang. *IDS Snort* dan *Suricata* diinstal serta dikonfigurasi di *Ubuntu server* dengan *rules* khusus untuk mendeteksi *SQL Injection*. Penyerang menyiapkan *payload* seperti "*OR*"="*1*" dan *UNION SELECT*, lalu mengirimkannya ke *DVWA*. *IDS* memantau lalu lintas jaringan dan mencatat *log* serangan yang

kemudian diklasifikasikan sebagai *TP*, *FP*, *TN*, dan *FN*. Terakhir, dilakukan evaluasi akurasi deteksi menggunakan metrik seperti akurasi, presisi, *recall* dan *F1-score*.

3.8 Metode deteksi serangan

Metode deteksi serangan *SQL Injection* yang menggambarkan interaksi antara *Kali Linux* (*attacker*) dan *Ubuntu Server* (*victim*). Simulasi ini melibatkan *DVWA* sebagai target, *Snort* dan *Suricata* sebagai sistem deteksi intrusi (*IDS*), serta serangan manual dari *Kali Linux* untuk menguji efektivitas deteksi.



Gambar 5. Metode deteksi serangan
(Sumber: Data Penelitian, 2025)

Penelitian ini menggunakan *testbed* berupa *Ubuntu server* dengan *DVWA*

sebagai target dan *Kali Linux* sebagai mesin penyerang untuk melakukan serangan *SQL Injection* secara manual. Penyerang memilih dan mengirimkan *payload* seperti '*OR*'='1', *UNION SELECT*, atau *Teknik Error Based* ke *DVWA*. Lalu lintas serangan ini dipantau oleh *IDS* untuk mendeteksi dan mencatat aktivitas yang mencurigakan, yang kemudian dianalisis oleh *log*.

3.9 Metode evaluasi

Berikut adalah penjelasan singkat mengenai metrik evaluasi yang digunakan untuk menilai kinerja sistem deteksi intrusi,

$$akurasi = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Presisi = \frac{TP}{TP + FN}$$

$$Recall = \frac{TP}{TP + FP}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Snort dan *Suricata* merupakan *IDS* yang dirancang untuk mendeteksi ancaman jaringan seperti serangan *SQL Injection*. Kinerjanya dievaluasi melalui empat parameter: *True Positive (TP)* saat serangan berhasil terdeteksi, *True Negative (TN)* saat tidak ada serangan dan tidak ada peringatan, *False Positive (FP)* saat aktivitas normal salah terdeteksi sebagai serangan, dan *False Negative (FN)* saat serangan tidak terdeteksi. Parameter ini digunakan untuk menilai akurasi dan keandalan *IDS* dalam mengidentifikasi serangan.

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Bagian ini memaparkan hasil eksperimen untuk menganalisis kinerja *IDS Snort* dan

Suricata dalam mendeteksi serangan *SQL Injection*. Pengujian dilakukan di lingkungan *virtual* dengan skenario serangan pada aplikasi *web* rentan *DVWA*. Data yang dikumpulkan meliputi jumlah peringatan, jenis serangan terdeteksi, dan akurasi masing-masing *IDS* untuk menilai efektivitas keduanya dalam mengenali dan merespons serangan *SQL Injection*.

```

root@kali:~# sudo cat /var/log/suricata/suricata.json | head -n 20
{"timestamp": "2023-07-08T06:15:10.311000+0000",
  "flow_id": 1234567890123456,
  "event_type": "http",
  "src_ip": "192.168.76.255",
  "src_port": 48766,
  "dest_ip": "192.168.76.32",
  "dest_port": 80,
  "proto": "TCP",
  "http": {
    "hostname": "192.168.76.32",
    "url": "/vulnerabilities/sql/",
    "http_user_agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36",
    "http_content_type": "application/x-www-form-urlencoded",
    "http_method": "POST",
    "http_referer": "http://192.168.76.32/vulnerabilities/sql/",
    "length": 76
  },
  "alert": {

```

Gambar 6. Implementasi *Snort*
(Sumber: Data Penelitian, 2025)

```

root@kali:~# sudo cat /var/log/suricata/suricata.json | head -n 20
{"timestamp": "2023-07-08T06:15:10.311000+0000",
  "flow_id": 1234567890123456,
  "event_type": "http",
  "src_ip": "192.168.76.255",
  "src_port": 48766,
  "dest_ip": "192.168.76.32",
  "dest_port": 80,
  "proto": "TCP",
  "http": {
    "hostname": "192.168.76.32",
    "url": "/vulnerabilities/sql/",
    "http_user_agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36",
    "http_content_type": "application/x-www-form-urlencoded",
    "http_method": "POST",
    "http_referer": "http://192.168.76.32/vulnerabilities/sql/",
    "length": 76
  },
  "alert": {

```

Gambar 7. Implementasi *Suricata*
(Sumber: Data Penelitian, 2025)



Gambar 8. Web *DVWA* di Kali Linux
(Sumber: Data Penelitian, 2025)

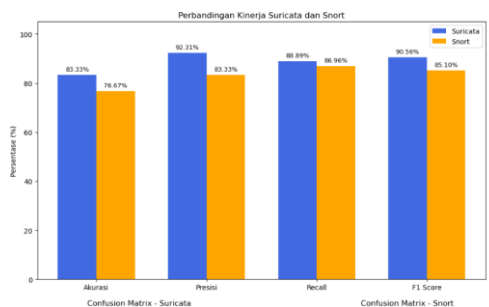
4.2 Evaluasi hasil deteksi

Setelah melakukan implementasi *Snort* dan *Suricata* serta simulasi serangan *SQL injection*, langkah selanjutnya adalah menganalisis hasil deteksi dari kedua sistem *IDS* tersebut. Analisis ini bertujuan untuk mengevaluasi efektivitas masing-masing *tools* dalam mengidentifikasi serangan, termasuk tingkat akurasi, kecepatan deteksi, serta kemunculan *false positive* atau *false negative*.

Tabel 3. Confusion Metrix

Metrix	Suricata	Snort
True Positives	24	20
False Positives	2	4
False Negatives	3	3
True Negatives	1	3

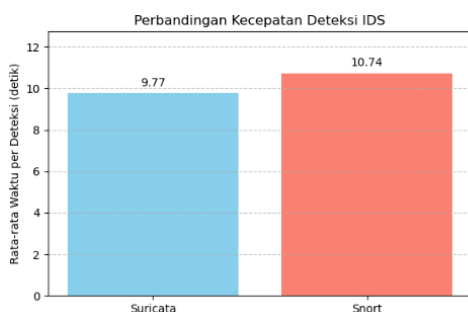
(Sumber: Data Penelitian, 2025)



Gambar 9. Evaluasi hasil deteksi
(Sumber: Data Penelitian, 2025)

Perbandingan akurasi deteksi *IDS* menunjukkan bahwa *Suricata* unggul pada gambar diatas. *Snort* di semua metrik evaluasi utama: akurasi, presisi, *recall*, dan *F1 score*. *Suricata* mencatat akurasi 83,33%, presisi 92,31%, *recall* 88,89%, dan *F1 score* 90,56%, sedangkan *Snort* masing-masing 76,67%, 83,33%, 86,96%, dan 85,10%. Hasil ini

menunjukkan bahwa *Suricata* lebih konsisten, akurat, dan efektif dalam mendeteksi serangan *SQL Injection* dengan tingkat kesalahan yang lebih rendah dibandingkan *Snort*.



Gambar 10. Deteksi Kecepatan
(Sumber: Data Penelitian, 2025)

SIMPULAN

Penelitian ini menunjukkan bahwa *Snort* dan *Suricata* berhasil diimplementasikan untuk mendeteksi serangan *SQL Injection* menggunakan aturan berbasis *signature*. Dari hasil pengujian, *Suricata* terbukti lebih unggul dibandingkan *Snort*, baik dari segi akurasi, efektivitas deteksi, maupun tingkat kesalahan yang lebih rendah. Selain itu, *Suricata* mampu melakukan deteksi lebih cepat, menjadikannya pilihan yang lebih tepat untuk lingkungan yang membutuhkan respons secara waktu nyata.

DAFTAR PUSTAKA

Abdulganiyu, O. H., Ait Tchakouch, T., & Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (IDS). *International Journal of Information Security*, 22(5), 1125–1162. <https://doi.org/10.1007/s10207-023-00682-2>

4.3 Pembahasan

Hasil implementasi menunjukkan bahwa *Suricata* lebih unggul dibandingkan *Snort* dalam mendeteksi serangan *SQL Injection*. *Suricata* mencatat akurasi 83,33% dan *F1 score* 90,56%, dengan lebih banyak *true positive* 24 dan lebih sedikit *false positive* 2, serta waktu deteksi yang lebih cepat, yaitu 9,77 detik per insiden. Sementara itu, *Snort* mencatat akurasi 76,67% dan *F1 score* 85,10%, dengan 20 *true positive* dan 4 *false positive*. Keunggulan *Suricata* didukung oleh konfigurasi yang lebih fleksibel, dukungan analisis berbasis koneksi, dan kemampuan mendeteksi lebih banyak varian *SQLi* secara akurat, menjadikannya lebih cocok untuk lingkungan produksi yang menuntut respons cepat dan andal

- Barends, J. K., Dewanta, F., & Karna, N. B. A. (2022). Perancangan dan Analisis Intrusion Prevention System Berbasis SNORT dan IPTABLES dengan Integrasi Honeypot pada Arsitektur Software Defined Network. *Multinetics*, 7(2), 163–176. <https://doi.org/10.32722/multinetics.v7i2.4276>
- Chen, D., Yan, Q., Wu, C., & Zhao, J. (2021). SQL Injection Attack Detection and Prevention Techniques Using Deep Learning. *Journal of Physics: Conference Series*, 1757(1). <https://doi.org/10.1088/1742-6596/1757/1/012055>
- de Santana, K. G. Q., Schwarz, M., & Wangham, M. S. (2024). Cybersecurity Testbeds for IoT:

- A Systematic Literature Review and Taxonomy. *Journal of Internet Services and Applications*, 15(1), 450–473. <https://doi.org/10.5753/jisa.2024.4363>
- Efe, A., & Abaci, İ. N. (2022). Comparison of the Host Based Intrusion Detection Systems and Network Based Intrusion Detection Systems. *Celal Bayar Üniversitesi Fen Bilimleri Dergisi*, 18(1), 23–32. <https://doi.org/10.18466/cbayarfbe.832533>
- Hoover, C., & Thompson, D. R. (2022). *Comparative Study of Snort 3 and Suricata Intrusion Detection Systems Click here to let us know how this document benefits you . by.*
- Hu, Q., Yu, S. Y., & Asghar, M. R. (2020). Analysing performance issues of open-source intrusion detection systems in high-speed networks. *Journal of Information Security and Applications*, 51, 102426. <https://doi.org/10.1016/j.jisa.2019.102426>
- Jemal, I., Cheikhrouhou, O., Hamam, H., & Mahfoudhi, A. (2021). SQL Injection Attack Detection and Prevention Techniques Using Deep Learning. *Journal of Physics: Conference Series*, 1757(1). <https://doi.org/10.1088/1742-6596/1757/1/012055>
- Kareem, F. Q., Ameen, S. Y., Salih, A. A., Ahmed, D. M., Kak, S. F., Yasin, H. M., Ibrahim, I. M., Ahmed, A. M., Rashid, Z. N., & Omar, N. (2021). SQL Injection Attacks Prevention System Technology: Review. *Asian Journal of Research in Computer Science*, July, 13–32. <https://doi.org/10.9734/ajrcos/2021/v10i330242>
- Safana Hyder Abbas, Wedad Abdul Khuder Naser, & Amal Abbas Kadhim. (2023). Subject review: Intrusion Detection System (IDS) and Intrusion Prevention System (IPS). *Global Journal of Engineering and Technology Advances*, 14(2), 155–158. <https://doi.org/10.30574/gjeta.2023.14.2.0031>

	Biodata, Penulis pertama, Fabian Ramot Argenta Pasaribu, merupakan mahasiswa Prodi Sistem Infomasi Universitas Putera Batam. Mahasiswa aktif dalam bidang informatika
	Biodata, Penulis kedua Andi Maslan, merupakan Dosen Prodi Sistem Informasi Universitas Putera Batam.