BAB I

PENDAHULUAN

1.1 Latar Belakang

Aplikasi *web* memiliki peran yang sangat penting dalam kehidupan sehari – hari di era digital saat ini. Banyak orang yang memanfaatkan untuk berbagai aktivitas, seperti berbelanja *online*. Umumnya, situs *web* menggunakan *database* di *backend* untuk menyimpan data dari pengguna. Karena *database* ini menyimpan informasi sensitif, seperti kata sandi, nomor kartu kredit dan nomor jaminan sosial disimpan dalam *database*, penyerang siber sering juga menargetkannya untuk menyerang (Kareem et al., 2021). Peningkatan nya dalam jumlah pengguna website secara signifikan membuka celah bagi pelaku serangan siber untuk mengeksploitasi kerentanan sistem, salah satunya melalui serangan *SQL Injection*.

Berdasarkan dari laporan Badan Siber dan Sandi Negara di tahun 2024, 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 *Phising* 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 (Id-SIRTII /CC, 2023).

Serangan *Structrured 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.

Dua *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 efesiensi hasil deteksi.

Ada beberapa peneliti sebelumnya, pada studi yang dilakukan oleh (Erlansari et al., 2020) "*Early Intrusion Detection System (IDS) using Snort and Telegram approach*", membangun sistem *IDS* berbasis *Snort* yang terintegrasi dengan *Telegram* untuk mendeteksi serangan jaringan, sperti *port scanning*, sekaligus memberikan notifikasi secara otomatis, *brute force*, dan *DDoS* secara *real time*. Hasilnya menunjukkan bahwa sistem diuji lewat simulasi dan terbukti efektif meningkatkan keamanan jaringan. Pada studi oleh (Fadhilah & Marzuki, 2020) dengan judul "*Performance Analysis of IDS Snort and IDS Suricata with Many-Core Processor in Virtual Machines Against Dos/DDoS Attacks*" membahas perbandingan kinerja *IDS snort* 3.0 dan *Suricata* dalam mendeteksi *SodS* dan *DDoS* di lingkungan *virtual machine* dengan berbagai jumlah *core* prosesor. Hasilnya menunjukkan bahwa *snort* 3.0, yang sudah mendukung *multi threading*, mampu mendeteksi lebih banyak paket serangan seiring peningkatan jumlah *core*, meskipun menggunakan lebih banyak *CPU* dan *memory*. Sebaliknya *Suricata* lebih efesien dalam penggunaan sumber daya, tetapi tidak menunjukkan peningkatan deteksi saat jumlah *core* ditambah. Studi lain juga dilakukan oleh (Kalabo et al., 2024) dengan judul "Analisa Performa *Intrusion Detection System (IDS) Snort* Dan *Suricata* Terhadap Serangan *TCP SYN Flood*" peneliti melakukan simulasi serangan dengan berbagai jumlah paket dan menguji akurasi deteksi, kecepatan deteksi, serta penggunaan sumber daya *RAM* dari kedua *IDS* tersebut. Hasilnya menunjukkan bahwa *Snort* lebih unggul dalam hal akurasi dan kecepatan deteksi dibandingkan *Suricata*, sementara *Suricata* lebih efesien dalam penggunaan sumber daya *RAM*.

Sebagai latar belakang dalam penelitian ini, peneliti akan menggunakan *DVWA* di *virtual machine Ubuntu* sebagai *web victim* dan penyerangan menggunakan *virtual machine kali linux* menggunakan *SQL Injection manual*. *Snort* dan *Suricata* akan diimplementasikan dalam lingkungan *web server* di *ubuntu* untuk memantau dan mendeteksi serangan yang mungkin dapat terjadi. Penelitian ini bertujuan untuk untuk mengetahui efektivitas dan tingkat akurasi masing – masing sistem untuk mendeteksi serangan, khususnya serangan *SQL Injection*.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, identifikasi masalah pada penelitian ini sebagai berikut:

1. Efektivitas deteksi serangan *SQL injection* oleh *Snort* dan *Suricata* belum secara komprehensif.
2. *Snort* dan *Suricata* memiliki perbedaan dalam arsitektur dan pendekatan teknis, namun tidak banyak penelitian yang menganalisis bagaimana perbedaan tersebut mempengaruhi performa dalam mendeteksi serangan *SQL Injection*.
3. Kebutuhan akan *IDS* yang dapat bekerja secara efisien dalam lingkungan nyata dengan lalu lintas yang kompleks belum diukur secara mendetail, khususnya terkait kemampuan dalam menangani varian – varian *SQL Injection*.

1.3 Batasan penelitian

Berdasarkan identifikasi yang telah dijelaskan, penelitian ini akan fokus pada permasalahan performa deteksi serangan *SQL Injection* oleh *Snort* dan *Suricata*. Untuk lebih memperjelas arah penelitian, maka rumusan masalah yang dapat diajukan adalah sebagai berikut:

1. Peneliti ini hanya menguji performa *Snort* dan *Suricata* dalam mendeteksi serangan *SQL injection* saja, tanpa mengkaji jenis serangan siber lainnya.
2. Peneliti menguji *IDS* dengan menggunakan *Snort* dan *Suricata* sebagai subjek utama penelitian.
3. Peneliti menggunakan lingkungan pengujian dalam jaringan lokal, sehingga hasilnya mungkin berbeda jika diterapkan pada jaringan yang lebih luas.

1.4 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, penelitian ini akan fokus pada permasalahan performa deteksi serangan *SQL Injection* oleh *Snort* dan *Suricata*. Untuk lebih memperjelas arah penelitian, maka rumusan masalah yang dapat diajukan adalah sebagai berikut:

1. Bagaimana mengimplementasikan *Snort* dan *Suricata* dalam mendeteksi serangan *SQL Injection*?
2. Bagaimana tingkat akurasi deteksi serangan *SQL Injection* menggunakan *Snort* dan *Suricata*?
3. Bagaimana efektivitas *IDS Snort* dan *Suricata* dalam mendeteksi serangan *SQL Injection*?

1.5 Tujuan Penelitian

Tujuan penelitian ini bisa dijabarkan sebagai berikut:

1. Untuk mengimplementasikan *Snort* dan *Suricata* dalam mendeteksi serangan *SQL Injection.*
2. Untuk mengetahui tingkat akurasi deteksi serangan *SQL Injection* menggunakan *Snort* dan *Suricata*.

3. Untuk mengetahui efektivitas *IDS Snort* dan *Suricata* dalam mendeteksi serangan *SQL Injection*.

1.6 Manfaat Penelitian

Manfaat penelitian dari menguji *IDS* untuk *Snort* dan *Suricata* terhadap serangan *SQL injection*.

1.6.1 Secara Teoritas

Penelitian ini untuk menambah literatur tentang analisis performa *IDS Snort* dan *Suricata* dalam mendeteksi serangan *SQL Injection*, serta memahami bagaimana perbedaan teknis antara kedua *IDS* tersebut mempengaruhi efektivitas deteksinya.

1.6.2 Secara Praktisi

Penelitian ini untuk memberikan panduan bagi praktisi keamanan siber dalam memilih *IDS* yang paling sesuai untuk digunakan dalam lingkungan produksi yang rentan terhadap serangan *SQL injection*. Dengan hasil penelitian ini, diharapkan para praktisi dapat lebih memahami keunggulan dan keterbatasan masing – masing *IDS*, sehingga dapat diterapkan pada jaringan mereka dengan lebih efektif.