SIMULASI PENCEGAHAN SERANGAN *DENIAL OF SERVICE* DENGAN MENGGUNAKAN TEKNOLOGI *INTRUSION PREVENTION SYSTEM* BERBASIS SNORT TERHADAP KEAMANAN JARINGAN *WIRELESS* LAN

Moses Marganda Lumban Tobing
Program Studi D3 Teknik Telekomunikasi
Universitas Telkom Kampus Jakarta
Jakarta, Indonesia
mosestobing@student.telkomuniversity.ac.id

Nurwan Reza Fachrur Rozi, S.T., M.T.
Pembimbing 1 D3 Teknik Telekomunikasi Universitas Telkom Kampus Jakarta Jakarta, Indonesia
nurwan@telkomuniversity.ac.id

Muhammad Iqbal, S.T., M.T.
Pembimbing 2 D3 Teknologi Telekomunikasi Universitas Telkom Bandung, Indonesia
miqbal.staff.telkomuniversity.ac.id

Jumlah pengguna internet di Indonesia terus meningkat, mencapai 221 juta pada triwulan pertama 2024 (APJII, 2024). Peningkatan ini seiring dengan meningkatnya serangan siber, dengan 279,84 juta serangan tercatat pada 2023 (BSSN). Serangan Denial of Service (DoS) adalah salah satu jenis serangan yang merusak, menghabiskan sumber daya sistem sehingga tidak dapat berfungsi dengan benar. Penelitian ini bertujuan mencegah serangan Denial of Service (DoS) pada jaringan WLAN menggunakan IPS berbasis Snort. Metode yang dilakukan pada penelitian ini meliputi, studi literatur, perencanaan, uji dan simulasi, dan analisis hasil. Hasilpada penelitian ini menunjukkan Snort efektif mencegah serangan DoS dan mengurangi dampaknya terhadap performa jaringan. Penelitian ini mengevaluasi kemampuan Snort dalam identifikasi dan mitigasi serangan serta berkontribusi pada pemahaman efektivitas Snort IPS dalam skenario nyata dan pengembangan sistem keamanan jaringan.

Kata Kunci: IPS, Snort, WLAN, Denial of Service (DoS), TCP Flood, UDP Flood, Rules, Alert, hping3, Dropped Packet

I. PENDAHULUAN

A. Latar Belakang

Internet kini menjadi bagian vital dari kehidupan sehari-hari masyarakat global. Berdasarkan data APJII, jumlah pengguna internet di Indonesia pada tahun 2023 mencapai 215 juta pengguna. Pada triwulan pertama tahun 2024, jumlah ini meningkat menjadi 221 juta pengguna, bertambah 6 juta pengguna dalam waktu kurang dari tiga bulan [1].

Dengan meningkatnya jumlah pengguna internet, tingkat kejahatan siber juga meningkat. Menurut laporan BSSN melalui Data Indonesia, pada tahun 2023 terjadi 279,84 juta serangan siber di Indonesia, yang berarti ada lebih dari satu serangan per pengguna [2]. Jenis serangan termasuk Generic Trojan RAT *Activity, Advanced Persistent Threat (APT), malware,* dan *Denial of Service* (DoS).

Di dalam sejarah perkembangan internet di dunia, terdapat beberapa kejahatan siber yang menggemparkan dunia yang salah satunya adalah serangan DDoS [3]. Seperti yang terjadi pada situs CNN tahun 2014 dan GitHub tahun 2018, yang mengakibatkan gangguan layanan dan kerusakan reputasi. Di Indonesia, serangan DDoS pada KasKus tahun 2017 menyebabkan situs tidak dapat diakses, mengganggu komunitas online. Laporan dari Vida menunjukkan peningkatan 40% serangan DDoS di Indonesia pada tahun 2023 dan prediksi peningkatan pada tahun 2024.

Jaringan komputer menggunakan media gelombang elektromagnetik untuk transmisi data dikenal sebagai WLAN atau Wi-Fi. Wi-Fi digunakan untuk akses internet dan jaringan tanpa kabel di organisasi, namun memiliki celah keamanan seperti LAN berkabel. Pihak yang tidak bertanggung jawab dapat menyusup ke jaringan WLAN dengan identitas IP *Address* atau *port* sasaran yang terbuka.

Snort IPS adalah perangkat lunak yang mendeteksi dan menangani serangan jaringan termasuk DoS, port scan, dan worm. Snort menganalisis paket data berdasarkan aturan yang ditentukan pengguna. Fleksibilitas dan kompatibilitasnya dengan berbagai sistem operasi membuat Snort populer di kalangan profesional keamanan siber untuk mendeteksi dan mencegah intrusi jaringan.

Penelitian ini bertujuan untuk mengetahui tingkat kerentanan keamanan jaringan WLAN dan menggunakan Snort untuk mendeteksi serta mencegah serangan. Melalui simulasi, penelitian ini akan mengidentifikasi dan menganalisis keamanan jaringan dengan Snort.

B. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijabarkan, maka rumusan masalah yang ditetapkan dalam penelitian Proyek Akhir ini adalah:

1. Bagaimana cara penerapan IPS Snort terhadap keamanan jaringan *Wireless* LAN sehingga dapat bekerja sebagai *monitoring* keamanan jaringan?

2. Bagaimana cara melakukan pencegahan terhadap keamanan jaringan *Wireless* LAN?
3. Bagaimana kelebihan dari penerapan IPS Snort terhadap keamanan jaringan *Wireless* LAN?

C. Batasan Masalah

Batasan masalah dalam penelitian Proyek Akhir ini yaitu:

1. Penggunaan *software* Snort untuk melakukan deteksi sekaligus serangan jaringan internet.
2. Metode yang digunakan dalam penelitian *penetration testing* untuk melakukan pemindaian dan simulasi serangan terhadap jaringan.
3. Jenis serangan yang akan disimulasikan adalah *Denial of Service* berupa protokol TCP SYN *Flood*, UDP *Flood* dan kombinasi antara TCP SYN *Flood* dan UDP *Flood*.
4. Percobaan pemindaian dan simulasi serangan akan dilakukan 10 kali dalam waktu 60 detik dan 120 detik.
5. Sumber jaringan internet berasal dari hotspot seluler *smartphone*.
6. *Rules* yang dibuat hanya untuk mengatasi serangan TCP SYN *Flood* dan UDP *Flood*.

D. Tujuan Penelitian

Tujuan dalam penelitian Proyek Akhir ini yaitu:

1. Untuk mengetahui tentang cara penerapan IPS Snort terhadap keamanan jaringan *Wireless* LAN sehingga dapat bekerja sebagai *monitoring* keamanan jaringan.
2. Untuk mengetahui tentang cara melakukan pendeteksian sekaligus pencegahan terhadap keamanan jaringan *Wireless* LAN.
3. Untuk mengetahui kelebihan dari penerapan IPS Snort terhadap keamanan jaringan *Wireless* LAN.

II. KAJIAN TEORI

A. Tinjauan Pustaka

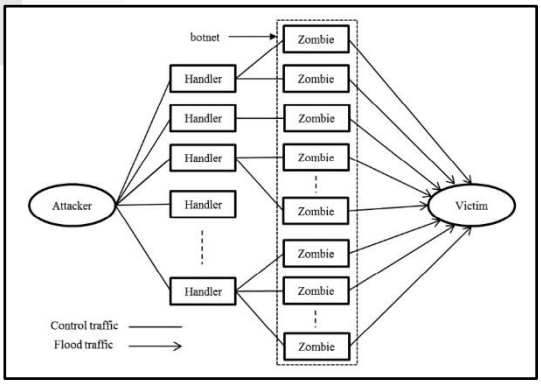
Penelitian yang berjudul "Implementasi Dan Analisis Pertahanan Dari Serangan DoS Dan DDoS Pada *Virtual Server* Dengan Menggunakan HIPS Snort" oleh Rizky Aditya pada tahun 2020 berfokus pada penggunaan HIPS Snort untuk melindungi *virtual server* dari serangan DoS dan DDoS, khususnya TCP SYN *Flood*. Hasil penelitian ini menunjukkan bahwa HIPS Snort dengan aturan yang telah dibuat dapat menahan serangan dalam waktu kurang lebih 60 detik dengan rata-rata paket yang di-drop mencapai 96,89% untuk satu *attacker* dan 98,04% untuk dua *attacker*. Percobaan dilakukan sebanyak 30 kali dengan hasil rata-rata paket yang di-drop sebesar 1.236.309,4 untuk satu *attacker* dan 1.432.763,4 untuk dua *attacker*. Selain itu, HIPS Snort terbukti lebih efektif dibandingkan *Firewall* dalam hal persentase paket *drop* yang mendekati angka 100%.

Penelitian yang berjudul "Perancangan Dan Analisis Sistem Keamanan Jaringan Komputer Menggunakan Snort" oleh Winrou Wesley Purba dan Rissal Efendi pada tahun 2020 membahas penggunaan NIDS Snort dalam sistem keamanan jaringan komputer di PT Promanufacture. Hasil penelitian menunjukkan bahwa penggunaan NIDS Snort dapat meminimalisir serangan DDoS, terlihat dari beberapa parameter keberhasilan seperti performansi CPU yang menurun dari 40% 3,09 GHz menjadi 1% 2,59 GHz, penggunaan *memory* yang berkurang dari 46% menjadi 42%, *utilization* yang turun dari 40% menjadi 1%, *speed processing* yang menurun dari 3,09 GHz menjadi 2,59 GHz, jumlah *thread* yang meningkat dari 1949 menjadi 2040, dan *handles* yang naik dari 70.948 menjadi 72.092.

Peneltian yang berjudul "*Analysis of Digital Forensics in the Implementation of Intrusion Detection using Snort*" oleh S. A. Mogaji, O.A. Ayeni, dan V. A. Olutayo pada tahun 2022 berfokus terhadap penggunaan IDS Snort bersama dengan Honeypot dan Wireshark untuk analisis forensik serangan jaringan. Penelitian ini berhasil menunjukkan bahwa Honeypot yang diinstall mampu mengelabui *attacker* seolah-olah merupakan server asli. Ketika *attacker* melakukan FTP *attack*, aksi tersebut berhasil direkam oleh Snort dan data tersebut diekspor ke *library* tcpdump untuk dianalisis lebih lanjut menggunakan Wireshark. Analisis ini memberikan informasi terkait waktu serangan, IP *address* sumber (*attacker*), dan IP *address* tujuan (*victim*).

B. Denial of Service

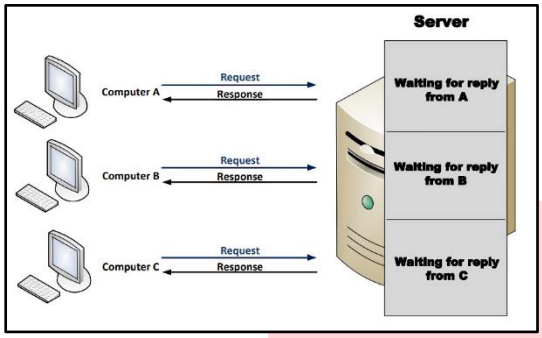
Denial of Service (DoS) adalah serangan siber yang membuat layanan jaringan tidak tersedia bagi pengguna sah dengan membanjiri target dengan lalu lintas berlebihan atau data yang merusak sistem. Serangan ini menyebabkan layanan menjadi lambat atau tidak responsif. DoS merupakan salah satu bentuk serangan yang paling banyak dipublikasikan dan sulit diatasi, tetapi mudah dilakukan dan berpotensi menyebabkan kerusakan signifikan. Serangan ini bertujuan mencegah penggunaan layanan oleh pihak yang berwenang dengan mengonsumsi sumber daya sistem, mengakibatkan gangguan komunikasi dan kerugian waktu serta uang.



Gambar 1 Skema DDoS Attack [4]

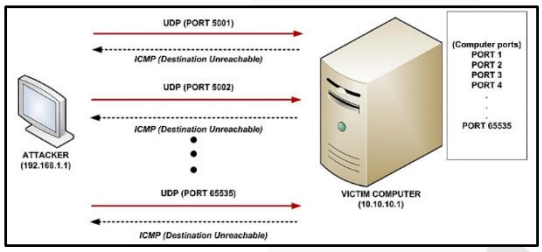
C. TCP SYN & UDP *Flood Attack*

TCP SYN *Flood Attack* adalah serangan DoS dan DDoS yang membanjiri server dengan permintaan TCP SYN tanpa menyelesaikan proses *handshake*. Ini menyebabkan perangkat korban kewalahan sehingga gagal mengirimkan ACK yang diharapkan karena alamat IP sumber yang palsu dan server tidak menerima paket konfirmasi. Akhirnya, permintaan menumpuk dan memenuhi memori.



Gambar 2 Skema TCP SYN Flood [5]

UDP *Flood Attack* adalah serangan DoS dan DDoS di mana penyerang mengirim banyak paket UDP ke *port* acak pada komputer korban. Ini membuat komputer korban kewalahan dan dapat memperlambat atau membuat sistem *crash*. Serangan TCP dan UDP Flood membanjiri korban dengan datagram untuk membebani sistem, tetapi TCP *Flood* mengonsumsi sumber daya CPU, sementara UDP Flood menghabiskan *bandwidth* koneksi, menyebabkan degradasi lebih parah pada lalu lintas jaringan sah.



Gambar 3 Skema UDP Flood [5]

D. Penetration Testing

Penetration testing (pentest) adalah simulasi serangan untuk mengidentifikasi dan memperbaiki kelemahan keamanan dalam sistem komputer, jaringan, atau aplikasi. *Penetration testing* juga dapat menentukan seberapa sulit seorang *attacker* untuk mampu menembus pertahanan sistem keamanan suatu organisasi/perusahaan dimana dilakukan dengan melakukan simulasi pengguna yang tidak sah untuk menyerang sistem yang menjadi sasaran seorang *attacker* [6]. Berikut adalah metodologi pentest:

1. **Perencanaan dan Pengintaian (*Reconnaissance*):** Mengumpulkan informasi tentang target seperti alamat IP dan domain.
2. **Pemindaian (*Scanning*):** Menggunakan alat otomatis untuk memetakan jaringan, mengidentifikasi layanan, port terbuka, dan kelemahan.
3. **Eksploitasi (*Exploitation*):** Memanfaatkan kelemahan yang ditemukan untuk mendapatkan akses tidak sah.
4. **Pemeliharaan Akses (*Post-Exploitation*):** Menjaga akses ke sistem untuk pengujian lebih lanjut dan pengumpulan informasi tambahan.
5. **Analisis dan Pelaporan (*Analysis and Reporting*):** Menyusun laporan yang merinci temuan, metode, dampak, dan rekomendasi perbaikan.

E. *Intrusion Prevention System* (IPS)

Intrusion Prevention System (IPS) adalah teknologi keamanan jaringan yang memantau dan mencegah ancaman atau serangan berbahaya dengan mendeteksi aktivitas mencurigakan dan mengambil tindakan mitigasi. IPS bertugas untuk memonitor paket-paket data (*data packets*) jaringan dari adanya aktivitas mencurigakan dan mencoba melakukan aksi tertentu yang disesuaikan dengan menggunakan aturan tertentu (Xinyau Zhang, 2007) [7].

a. *Network-based Intrusion Prevention System* (NIPS)

Network-based Intrusion Prevention System (NIPS) adalah salah satu kategori IPS yang merupakan sebuah sistem keamanan jaringan yang yang tidak hanya memonitor dan mendeteksi aktivitas mencurigakan di jaringan, tetapi juga mengambil tindakan untuk mencegah atau menghentikan serangan yang terdeteksi. NIPS biasanya ditempatkan di titik strategis dalam jaringan untuk dapat memantau dan mempengaruhi lalu lintas jaringan secara *real-time.*

b. *Host-based Intrusion Prevention System* (HIPS)

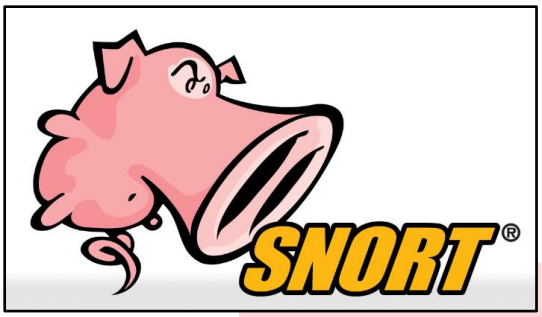
Host-based Intrusion Prevention System (HIPS) merupakan sebuah sistem teknologi keamanan yang dipasang pada perangkat individu, seperti server atau *workstation*, untuk memantau dan mencegah ancaman atau aktivitas berbahaya oleh pihak-pihak yang tidak bertanggung jawab. HIPS berfungsi untuk melindungi *host* dari serangan yang mungkin tidak terdeteksi oleh sistem keamanan jaringan lainnya.

F. Snort

Snort adalah sistem IDS/IPS *open source* yang dikembangkan oleh Sourcefire dan sekarang dimiliki oleh Cisco. Snort dapat dijalankan di berbagai *platform* seperti Linux, Windows, dan MacOS X. Snort melakukan analisis lalu lintas jaringan secara *real-time* untuk mendeteksi berbagai serangan dan aktivitas mencurigakan. Berikut adalah komponen utama Snort:

1. ***Packet Decoder*:** Menerima dan menguraikan paket data dari jaringan, menganalisis *header* untuk tahap analisis berikutnya.
2. ***Preprocessor*:** Mempersiapkan dan memodifikasi paket sebelum dianalisis oleh *detection engine*, termasuk *reassembly* paket, normalisasi protokol, dan deteksi anomali.
3. ***Detection Engine*:** Inti dari Snort yang mencocokkan paket dengan aturan yang ditetapkan untuk mendeteksi serangan dan aktivitas mencurigakan.

4. ***Logging and Alert System*:** Menangani penyimpanan dan pengelolaan *alert* yang dihasilkan, termasuk penulisan *log*, pengiriman *alert*, dan integrasi dengan sistem lain.
5. ***Output Modules*:** Menghasilkan dan memproses *alert* setelah mendeteksi aktivitas mencurigakan, menentukan cara dan tujuan pengiriman *alert*.



Gambar 4 Logo Software Snort [8]

G. Anomaly-based Detection

Anomaly-based detection adalah metode *yang* mengidentifikasi aktivitas mencurigakan dengan mendeteksi penyimpangan dari pola perilaku normal. Berbeda dengan metode berbasis *signature* yang mengandalkan pola serangan yang dikenal, *anomaly-based detection* memantau dan menganalisis anomali dalam aktivitas untuk mendeteksi serangan atau perilaku tidak normal. *Anomaly-based detection* dapat dimanfaatkan untuk mengenali berbagai serangan dari luar yang tidak sah untuk mendeteksi perilaku yang tidak normal dari pengguna (*user*) [9]. Metode ini menawarkan pendekatan dinamis dan adaptif, memungkinkan sistem mengenali dan merespons ancaman yang mungkin tidak terdeteksi oleh metode konvensional.

H. Alerts

Alerts di dalam Snort adalah komponen penting yang memberi tahu administrator tentang aktivitas yang mungkin merupakan ancaman. *Alerts* disimpan dalam log atau dikirim ke sistem *monitoring* untuk analisis lebih lanjut. Setiap *alert* memberikan informasi detail tentang kejadian yang terdeteksi, termasuk jenis serangan, sumber dan tujuan lalu lintas, serta konteks lainnya. *Alerts* dikategorikan menjadi beberapa tingkat, seperti sedang, berbahaya, dan paling berbahaya, yang membantu mengidentifikasi jenis serangan yang sering terjadi pada jaringan.

I. Rules

Baik sebagai sistem pendeteksian maupun sebagai sistem pencegahan, intrusi Snort bergantung pada keberadaan Snort *rules* untuk melindungi jaringan [10]. *Snort rules* tersebut dibagai menjadi dua bagian utama, yaitu:

a. Rule Header

Merupakan *rules* yang diberikan untuk mendefinisikan tindakan yang akan diambil pada lalu lintas yang cocok, protokol yang digunakan sumber paket, IP *Address* sumber, *port* sumber, arah lalu lintas jaringan yang diinginkan, IP *Address* tujuan dan *port* tujuan dari paket data yang datang.

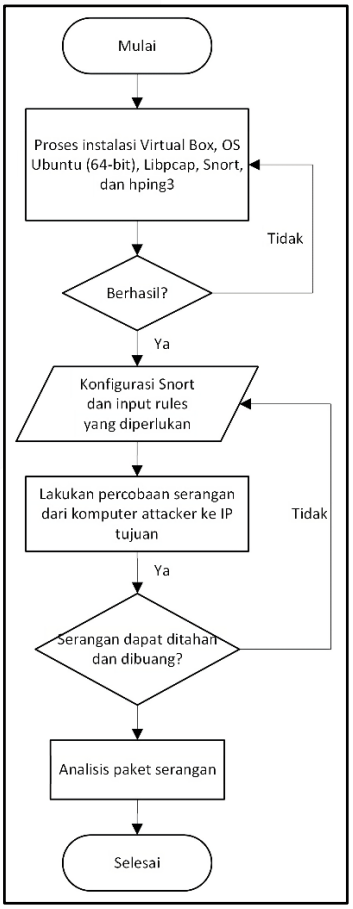
1. ***Rule Action*:** Tindakan yang akan diambil pada paket yang cocok. Ada lima aksi dasar:
   1) *Alert*: Memberi peringatan saat paket mencurigakan terdeteksi.
   2) *Drop*: Membuang paket setelah memberi peringatan.
   3) *Block*: Memblokir paket mencurigakan dan semua paket berikutnya dalam aliran.
   4) *Log*: Mencatat paket setelah memberi peringatan.
   5) *Pass*: Mengabaikan paket mencurigakan.
2. ***Protocol*:** Jenis protokol yang digunakan (IP, ICMP, TCP, UDP).
3. ***Source Address*:** Alamat IP sumber paket.
4. ***Source Port*:** Port sumber paket.
5. ***Direction*:** Arah lalu lintas (ditandai dengan "->").
6. ***Destination Address*:** Alamat IP tujuan paket.
7. ***Destination Port*:** *Port* tujuan paket.

b. Rule Option

Menentukan kriteria spesifik untuk memutuskan apakah paket akan diloloskan atau dihentikan. Ditulis dalam tanda kurung setelah *rule header* dan menentukan *output alert* yang diinginkan.

III. METODE

A. Flowchart



Gambar 5 Flowchart Pengerjaan Penelitian

Proyek akhir dimulai dengan instalasi beberapa alat yang diperlukan, termasuk Virtual Box untuk membuat PC *virtual*, Ubuntu (64-bit) sebagai sistem operasi, serta Snort, Libpcap, dan hping3. Jika instalasi gagal, ini menunjukkan adanya kesalahan atau kekurangan dalam langkah-langkah instalasi, sehingga proses harus diulang dari awal.

Setelah instalasi berhasil, langkah berikutnya adalah konfigurasi Snort. Ini melibatkan *input rules* yang bertujuan untuk mendeteksi dan melindungi server dari berbagai jenis serangan. Setelah Snort dikonfigurasi, percobaan dilakukan dengan meluncurkan serangan seperti TCP SYN *Flood*, UDP *Flood*, dan kombinasi antara TCP *Flood* dan UDP *Flood*.

Jika serangan tidak dapat ditahan dengan efektif atau paket data tidak dibuang dengan benar, kemungkinan ada kesalahan dalam konfigurasi yang memerlukan penyesuaian ulang. Namun, jika Snort berhasil menahan serangan dan membuang paket berbahaya secara efektif, langkah selanjutnya adalah menganalisis paket yang diterima untuk menentukan tindakan pencegahan tambahan. Proses ini akan membantu mencegah serangan di masa depan dan menyelesaikan simulasi.

B. Skenario Pengujian

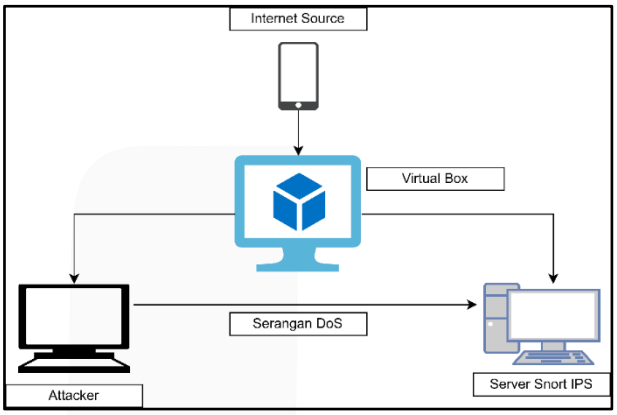
Dalam penelitian kali ini, peneliti akan menggunakan metode NDLC (*Network Design Life Cycle*). Tahapan-tahapan pada metode ini meliputi *analysis, design, simulation prototyping, implementation, monitoring* dan *management* [11]. Dalam perancangan arsitektur jaringan untuk proyek ini, topologi sederhana digunakan untuk mempermudah penelitian dan memberikan gambaran jelas mengenai lintas jaringan. Proyek dimulai dengan membangun topologi jaringan menggunakan satu PC utama yang menjalankan Oracle Virtual Box, dengan sumber jaringan internet dari *smartphone*. Dalam Virtual Box, terdapat dua VM, yaitu satu sebagai server dan satu sebagai *attacker*. IP *Address* didistribusikan menggunakan DHCP untuk memberikan alamat secara otomatis kepada setiap client.

Pada proses konfigurasi Snort, *rules* diatur untuk mendeteksi dan menangani serangan, termasuk TCP *Flood*, UDP *Flood*, dan kombinasi TCP SYN *Flood* dengan UDP *Flood*. Setelah Snort dikonfigurasi, simulasi dilakukan dengan meluncurkan serangan yang disebutkan. Paket data yang dikirimkan oleh *attacker* disimpan oleh libpcap dan diproses oleh Snort. Paket tersebut akan di-*decode* dan didefragmentasi, kemudian dianalisis oleh *detection engine* untuk memberikan *output* berupa *alert* dan *log*.

Parameter pengujian meliputi jenis serangan, jumlah paket yang dibuang, rata-rata paket dibuang, persentase paket dibuang, serta rata-rata paket diterima dan dianalisis. Analisis hasil pengujian bertujuan untuk mengevaluasi efektivitas Snort dalam menahan serangan dan mengidentifikasi tanda-tanda serangan DDoS seperti penggunaan *bandwidth* yang tinggi, respons server yang buruk, dan peningkatan beban CPU tanpa permintaan yang jelas. Keberhasilan Snort bergantung pada penerapan rules yang tepat dan kompatibilitas perangkat yang digunakan.

Tabel 1 Parameter Pengujian 1

No	Parameter	Keterangan
1.	*Attack Type*	Jenis serangan yang masuk ke dalam *server*
2.	*Total Paket Dropped*	Banyaknya paket yang berhasil dibuang oleh Snort
3.	Rata-Rata Paket *Dropped*	Rata-rata paket yang dibuang oleh Snort
4.	Persentase Rata-Rata Paket *Dropped*	Persentase rata-rata paket yang dibuang oleh Snort
5.	Rata-Rata Paket *Received*	Rata-rata paket yang diterima oleh Snort
6.	Rata-Rata Paket *Analyzed*	Rata-rata paket yang dianalisis oleh Snort untuk keputusan tindakan selanjutnya
7.	Persentase Rata-Rata Paket *Analyzed*	Persentase rata-rata paket yang dianalisis oleh Snort untuk keputusan tindakan selanjutnya



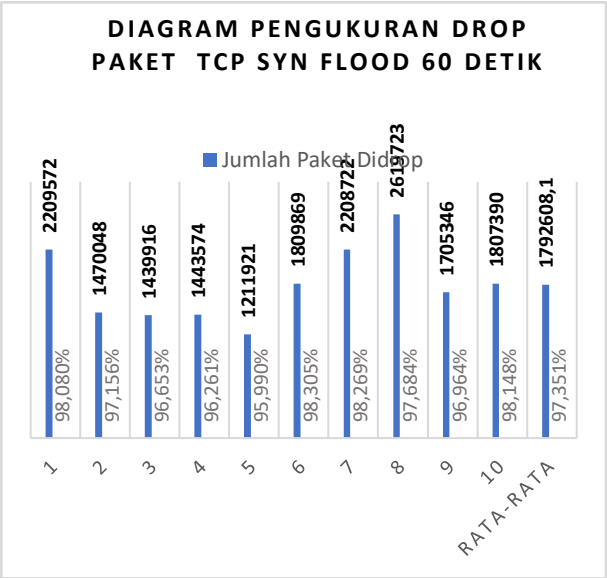
Gambar 6 Topologi Penelitian Proyek Akhir

IV. HASIL DAN PEMBAHASAN

A. Pengukuran TCP SYN *Flood* dengan Waktu 60 Detik & 120 Detik

1. Pengukuran dengan Waktu 60 Detik

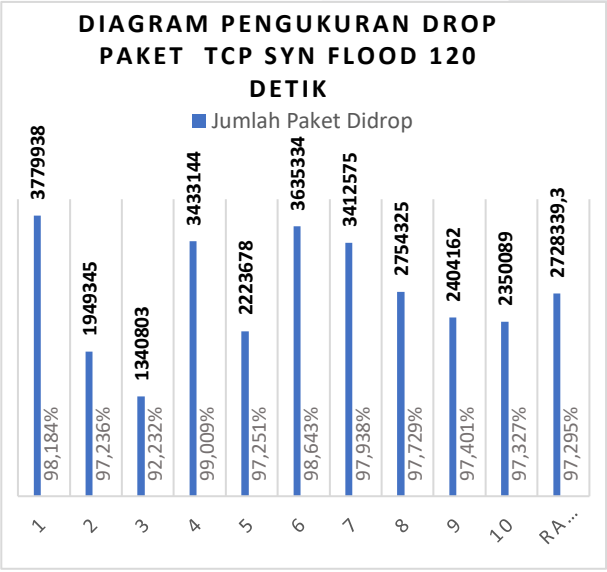
Pada pengujian selama 60 detik, Snort berhasil membuang 17.926.081 paket, yang merupakan 97,351% dari total paket yang diterima. Dalam hal ini, Snort memproses rata-rata 352 paket per detik dari total 43.249 paket yang diterima, dengan 2.209.572 paket atau 98,080% dari paket yang datang berhasil dibuang. Meskipun terlihat ada penyimpangan antara jumlah paket yang di-*drop* dan jumlah paket yang diterima, hal ini disebabkan oleh beban serangan yang sangat tinggi dari *attacker*, yang menyebabkan duplikasi lalu lintas dan membuat Snort harus men-*drop* paket secara berlebihan. Estimasi total paket yang sebenarnya datang adalah sekitar 2.252.827 paket, di mana Snort hanya menerima sekitar 1,919% dari total paket tersebut untuk diproses lebih lanjut.



Gambar 7 Diagram Packet Dropped SYN Flood 60 Detik

2. Pengukuran dengan Waktu 120 Detik

Dalam pengujian dengan durasi 120 detik, Snort berhasil membuang 27.283.393 paket, dengan rata-rata pembuangan mencapai 97,295% dari total paket yang diterima. Pada pengujian ini, Snort membuang 3.778.938 paket atau 98,184% dari total paket yang datang, yang menunjukkan bahwa meskipun durasi serangan lebih lama, efektivitas Snort dalam membuang paket berbahaya tetap tinggi. Selisih persentase pembuangan antara 60 detik dan 120 detik sangat kecil, yaitu 0,056%, menunjukkan konsistensi Snort dalam menahan serangan TCP SYN Flood, terlepas dari durasi serangan. Snort berhasil memproses rata-rata 46.254,3 paket dalam waktu 60 detik dan 65.003,2 paket dalam waktu 120 detik, dengan rata-rata paket yang dianalisis masing-masing sebanyak 15.299,2 paket dan 33.023,8 paket. Hasil ini menegaskan bahwa Snort efektif dalam mencegah serangan dan menjaga performa perangkat tetap baik dengan membuang sebagian besar paket berbahaya yang masuk.

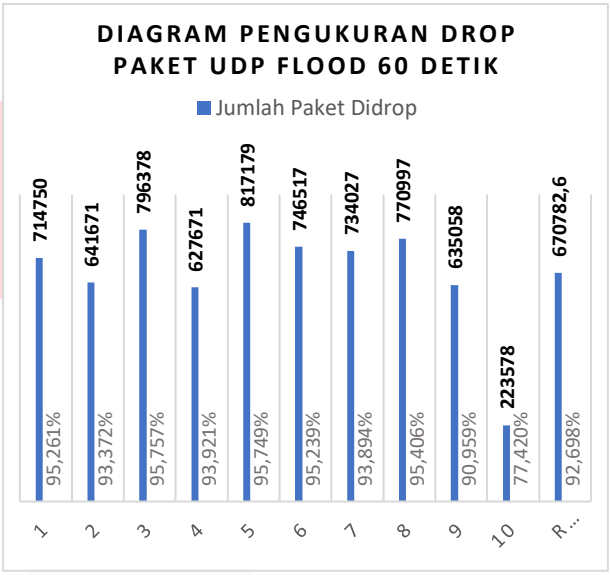


Gambar 8 Diagram Packet Dropped SYN Flood 120 Detik

B. Pengukuran UDP *Flood* dengan Waktu 60 Detik & 120 Detik

1. Pengukuran dengan Waktu 60 Detik

Pada pengujian selama 60 detik, Snort berhasil membuang 6.707.826 paket, yang merupakan 92,698% dari total paket yang diterima. Dalam hal ini, Snort memproses dan menganalisis rata-rata 44.380,3 paket, dengan 714.750 paket yang dibuang, yang setara dengan 95,261% dari total paket yang datang.

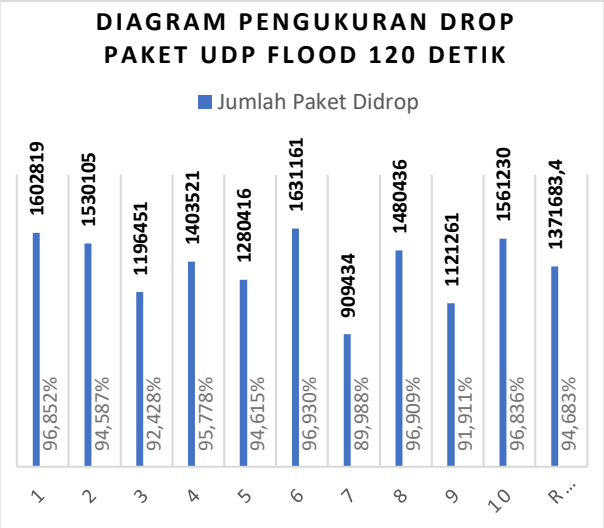


Gambar 9 Diagram Packet Dropped UDP Flood 60 Detik

2. Pengukuran dengan Waktu 120 Detik

Pada pengujian dengan durasi 120 detik, Snort membuang 13.716.834 paket, atau 94,683% dari total paket yang diterima. Snort berhasil membuang 1.602.819 paket, yang setara dengan 96,852% dari total paket yang datang selama pengujian ini. Rata-rata paket yang diterima untuk diproses dan dianalisis adalah 72.218,3 paket, dengan 53.286,8 paket yang dianalisis.

Hasil dari kedua pengujian menunjukkan bahwa Snort efektif dalam menanggulangi serangan UDP Flood, dengan selisih rata-rata kemampuan pembuangan paket antara 60 detik dan 120 detik hanya sebesar 1,985%. Snort mampu menjaga performa dengan membuang sebagian besar paket berbahaya dan tetap memproses paket yang valid untuk analisis lebih lanjut, yang penting untuk meningkatkan keamanan jaringan dengan mencegah ancaman sebelum mencapai target.

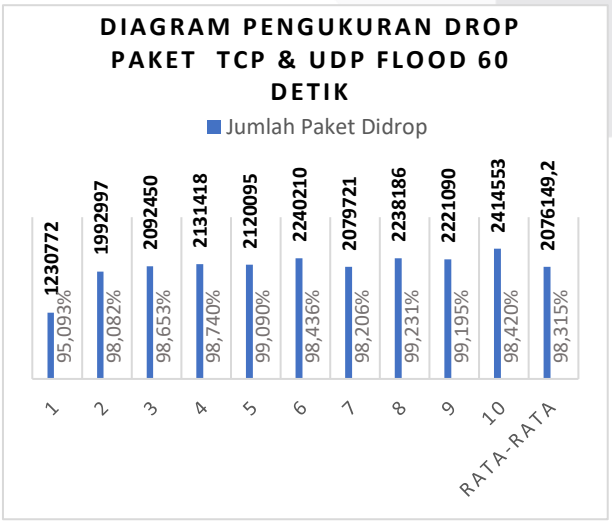


Gambar 10 Diagram Packet Dropped UDP Flood 120 Detik

C. Pengukuran Kombinasi TCP SYN & UDP *Flood* dengan Waktu 60 Detik & 120 Detik

1. Pengukuran dengan Waktu 60 Detik

Pada pengujian serangan kombinasi TCP SYN *Flood* dan UDP *Flood* dengan durasi 60 detik, Snort menunjukkan kinerja yang sangat baik dalam mengelola serangan tersebut. Dalam percobaan ini, Snort berhasil membuang total 20.761.492 paket, dengan rata-rata pembuangan paket mencapai 2.076.149,2 paket. Ini setara dengan 98,315% dari total paket yang diterima, menandakan efektivitas Snort dalam menghadapi beban serangan yang tinggi. Hasil analisis juga menunjukkan bahwa Snort berhasil membuang 1.230.772 paket dari total paket yang datang, yang setara dengan 95,093% dari paket yang masuk. Selama periode ini, rata-rata paket yang diterima untuk diproses dan dianalisis adalah 32.543,6 paket, dengan 21.093,7 paket di antaranya berhasil dianalisis. Ini menunjukkan bahwa meskipun Snort membuang sebagian besar paket berbahaya, sistem tetap dapat memproses dan menganalisis sejumlah paket untuk menjaga keamanan jaringan.

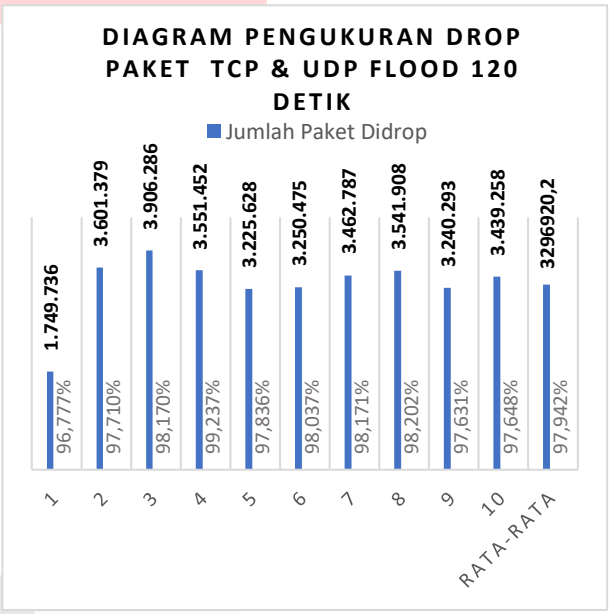


Gambar 11 Kombinasi SYN & UDP Flood Attack 60 Detik

2. Pengukuran dengan Waktu 120 Detik

Dalam pengujian serangan kombinasi TCP SYN *Flood* dan UDP *Flood* dengan durasi 120 detik, Snort tetap mempertahankan kinerja yang impresif. Total paket yang berhasil dibuang mencapai 32.969.202, dengan rata-rata pembuangan paket sebesar 3.296.920,2 paket, atau 97,942% dari total paket yang diterima. Selama pengujian ini, Snort juga membuang 3.601.379 paket dari total paket yang datang, setara dengan 97,710% dari paket yang masuk. Rata-rata paket yang diterima untuk diproses dan dianalisis selama 120 detik adalah 67.108,8 paket, dengan 33.862 paket di antaranya berhasil dianalisis. Ini menunjukkan bahwa meskipun jumlah paket berbahaya yang dibuang meningkat seiring dengan waktu, Snort masih dapat memproses dan menganalisis paket yang tersisa secara efisien, mempertahankan kemampuan sistem untuk melindungi jaringan dari serangan yang berkepanjangan.

Gambar 12 Diagram Packet Dropped Kombinasi SYN & UDP Flood 120 Detik



D. Perbandingan Kinerja Server

1. Kinerja Server Sebelum Serangan Dapat Ditahan

Sebelum Snort diaktifkan untuk menahan serangan, performa server mengalami penurunan yang signifikan. Seperti yang terlihat pada Gambar 13, CPU1 server bekerja pada kapasitas yang sangat tinggi, mencapai 95,7%, menunjukkan beban kerja yang sangat berat akibat banjiran permintaan dari *attacker*. Selain itu, penggunaan memori fisik meningkat menjadi 3,0 GiB atau 62,5% dari total kapasitas sebagai upaya untuk menangani serangan tersebut. Aktivitas jaringan juga meningkat drastis, dengan kecepatan *downlink* mencapai 601,7 KiB/s dan *uplink* 468,3 KiB/s.

Gambar 13 Kinerja Server Sebelum Serangan Dapat Ditahan

2. Kinerja Server Setelah Serangan Dapat Ditahan

Setelah Snort diaktifkan dan berhasil menahan serangan, performa server menunjukkan pemulihan yang jelas. Gambar 14 menunjukkan bahwa CPU1 kini bekerja pada tingkat yang lebih stabil, dengan penggunaan turun menjadi 25,8%. Penggunaan memori juga berkurang menjadi 2,8 GiB atau 57,7%. Aktivitas jaringan kembali normal, dengan kecepatan *downlink* rata-rata sebesar 128 *bytes*/s dan *uplink* yang kembali ke 0 bytes/s. Hal ini menandakan bahwa Snort berhasil membuang banjiran paket berbahaya dan mengembalikan server ke kondisi kerja yang lebih baik dan stabil.

Gambar 14 Kinerja Server Setelah Serangan Dapat Ditahan

V. KESIMPULAN

Berdasarkan perancangan sistem dan analisis hasil dari proyek akhir mengenai efisiensi sistem pertahanan IPS Snort, dapat disimpulkan bahwa Snort terbukti sangat efektif dalam mencegah serangan *Denial of Service*, baik dalam bentuk TCP SYN *Flood*, UDP *Flood*, maupun kombinasi keduanya, dengan persentase paket yang berhasil di-drop mencapai lebih dari 90%. Durasi dan kompleksitas serangan memengaruhi jumlah paket yang dapat di-drop, seperti terlihat dari kemampuan Snort meng-drop 98,315% paket selama 60 detik dan 97,942% paket selama 120 detik pada serangan kombinasi. Implementasi Snort dalam mode *Intrusion Prevention System* (IPS) juga menunjukkan efektivitas tinggi dalam menahan serangan DoS, dengan penurunan signifikan dalam penggunaan CPU dari 95,7% menjadi 25,8%, serta penggunaan memori fisik yang turun dari 62,5% menjadi 57,7%. Aktivitas jaringan yang stabil kembali setelah penanganan serangan menandakan bahwa Snort berhasil memfilter lalu lintas berbahaya dan mengembalikan kinerja server ke kondisi yang lebih efisien.

REFERENSI

[1] Asosiasi Penyelenggara Jasa Internet Indonesia, “APJII Jumlah Pengguna Internet Indonesia Tembus 221 Juta Orang,” 07 February 2024. [Online]. Available: https://apjii.or.id/berita/d/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang.

[2] M. A. Rizaty, "Data Jumlah Serangan Siber ke Indonesia Hingga 2023," Data Indonesia, 09 January 2024. [Online]. Available: https://dataindonesia.id/internet/detail/data-jumlah-serangan-siber-ke-indonesia-hingga-2023. [Accessed 22 February 2024].

[3] Lintasarta Cloudeka, "5 Contoh Serangan DDoS yang Menggemparkan dalam Sejarah IT yang Wajib Anda Ketahui!," Lintasarta, 30 October 2023. [Online]. Available: https://www.cloudeka.id/id/berita/web-sec/contoh-serangan-ddos/.

[4] B. A. Khalaf, S. A. Mostafa, A. Mustapha, M. A. Mohammed and W. M. Abduallah, "Comprehensive Review of Artificial Intelligence and Statistical Approaches in Distributed Denial of Service Attack and Defense Methods," *SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE AND COGNITIVE COMPUTING FOR COMMUNICATION AND NETWORK*, vol. 7, p. 4, 2019.

[5] K. Treseangrat, PERFORMANCE ANALYSIS OF DEFENSE MECHANISMS AGAINST UDP FLOOD ATTACKS, Research Bank, 2014.

[6] S. Shah and B. Mehtre, "An Overview of Vulnerability Assessment and Penetration Testing Techniques," *J Comput Virol Hack Tech*, vol. 11, no. February 2015, pp. 27-49, 2015.

[7] J. Gondohanindijo, "IPS (Intrusion Prevention System) Untuk Mencegah Tindak Penyusupan/Intrusi," *Majalah Ilmiah INFORMATIKA*, vol. 3, no. 3, pp. 1-22, 2012.

[8] Snort.org, "Snort," Snort Team, 2024. [Online]. Available: https://www.snort.org/. [Accessed March 2024].

[9] M. R. R. Islami and Fariadi, "DETEKSI DINI SERANGAN PADA WEBSITE MENGGUNAKAN METODE ANOMALI BASED," *JIKO (Jurnal Informatika dan Komputer)*, vol. 5, no. 17 Novermber 2022, pp. 224-229, 2022.

[10] Cisco Talos Detection Response Team, "Snort 3 Rule Writing Guide," [Online]. Available: https://docs.snort.org/rules/. [Accessed July 2024].

[11] T. Sanjaya and D. Setiyadi, "Network Development Life Cycle (NDLC) Dalam Perancangan Jaringan Komputer Pada Rumah Shalom Mahanaim," *JURNAL MAHASISWA BINA INSANI*, vol. 4, no. Agustus 2019, pp. 1-10, 2019.