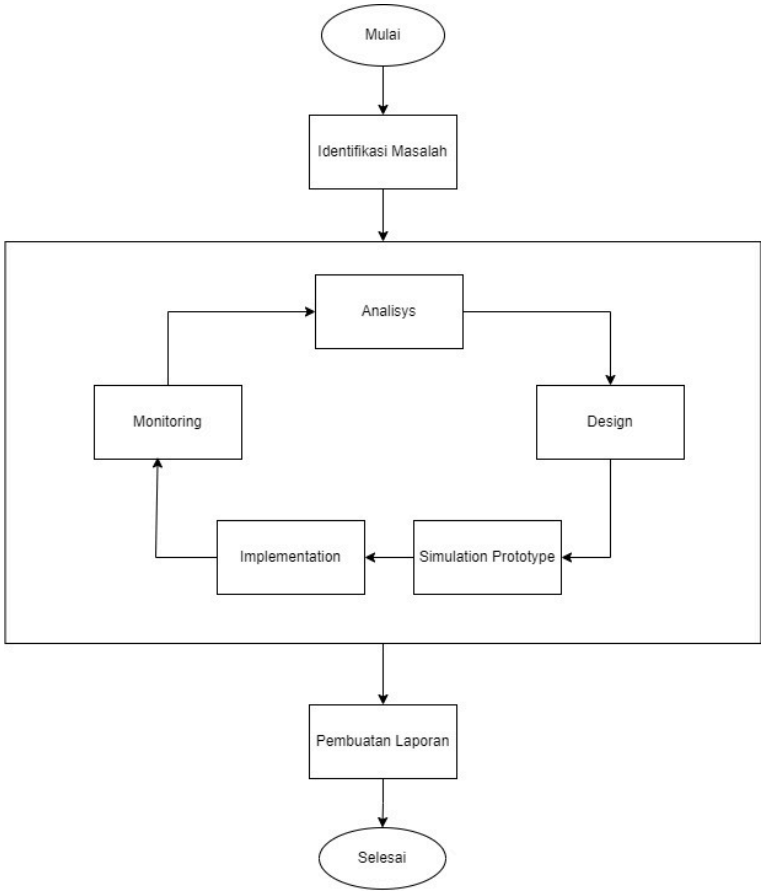
BAB III
METODE PENELITIAN

3.1. Metode Penelitian

Pada penelitian ini metodologi yang digunakan adalah *Network Development Life Cycle* (NDLC) metode tersebut terdiri dari *analysis, design, simulation prototype, implementation*, dan *monitoring* (Sanjaya & Setiyadi, 2019). Pemilihan metode NDLC pada penelitian ini karena metode NDLC merupakan salah satu metode yang efektif digunakan dalam pengembangan dan analisis insfrastruktur jaringan, dengan langkah-langkah yang spesifik sehingga dapat meningkatkan kinerja jaringan dalam menghadapi beragam gangguan yang disebabkan oleh banyak faktor. Metode NDLC seperti yang disebutkan sebelumnya metode NDLC memiliki tahapan *analysis, design, simulation prototype, implementation*, dan *monitoring.* Berikut ini merupakan alur penelitian menggunakan metode NDLC dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian Metode *Network Development Life Cycle.*

Muhamad Hisyam Nugraha Solihin, 2024
IMPLEMENTASI APLIKASI SNORT PADA SISTEM KEAMANAN SERVER DENGAN NOTIFIKASI TELEGRAM
Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

3.2. Analisis

Pada tahapan analisis penulis melakukan identifikasi terhadap kebutuhan sistem keamanan yang digunakan untuk melindungi *server* dari beragam serangan siber. Analisis yang dilakukan mencakup identifikasi jenis-jenis serangan siber yang biasanya terjadi pada *server* seperti DDoS, *SQL Injection*, XSS, dan serangan lainnya, serta kebutuhan integrasi sistem snort dengan notifikasi telegram. Kebutuhan perangkat keras dan perangkat lunak yang digunakan pada penelitian ini juga dilakukan analisis

3.2.1. Jenis dan Sumber Data

Pada penelitian ini jenis dan sumber data yang akan digunakan oleh peneliti adalah menggunakan data utama. Sumber data utama ini berasal dari implementasi snort dan notifikasi telegram yang terkirim dari hasil pengujian yang dilakukan, data yang diambil mencakup hasil pengamatan sistem, catatan pengujian, dan log yang dihasilkan selama eksperimen dilakukan. Data utama yang diambil akan memberikan informasi tentang respons sistem selama pengujian berlangsung.

3.2.2. Teknik Pengambilan Data

Pada penelitian ini ada beberapa faktor yang dilakukan atau diamati untuk mendapatkan data yang dibutuhkan. Berikut merupakan beberapa faktor yang diamati untuk pengambilan data pada penelitian ini:

1. Literatur, pada tahapan ini penulis mengumpulkan beberapa informasi mengenai implementasi snort dan penambahan notifikasi telegram yang dibuat yang berupa jurnal, skripsi, atau informasi lain pada internet untuk memberikan bayangan serta pemahaman teori yang lebih mendalam mengenai sistem yang dibuat pada penelitian ini.
2. Observasi pada tahapan ini penulis melakukan pengamatan secara langsung pada saat sistem keamanan snort dengan notifikasi telegram diimplementasikan pada ubuntu *server* versi 20.04.4. pengamatan yang dilakukan adalah untuk mengetahui response sistem dalam mendeteksi serangan yang terjadi terhadap jaringan ubuntu *server*.
3. Analisis, pada tahapan ini analisis yang dilakukan adalah mengamati log snort yang dijalankan pada ubuntu *server* yang akan menampilkan informasi serangan

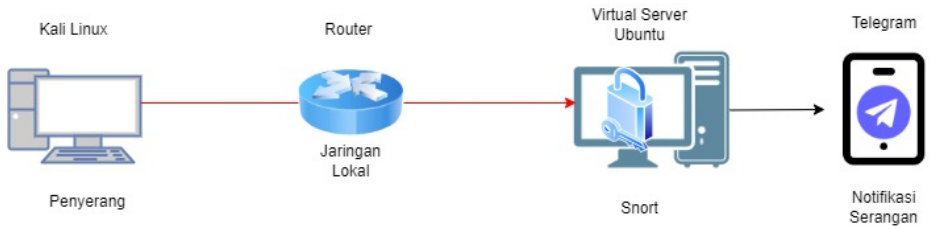
yang terdeteksi oleh *rules* snort serta seberapa cepat *rules* tersebut bereaksi terhadap serangan yang masuk.

4. Dokumentasi, pada tahapan ini hasil dari pengujian implementasi aplikasi snort pada ubuntu *server* yang meliputi dokumen teknis yang memuat informasi mengenai snort seperti panduan konfigurasi, serta petunjuk penggunaan sistem untuk memberikan pemahaman mengenai implementasi sistem snort yang dibuat.

Keempat tahapan diatas diharapkan bisa memberikan data yang beragam sehingga dapat dilakukan analisis mendalam dari hasil implementasi aplikasi snort yang dikombinasikan dengan notifikasi telegram.

3.3. Desain

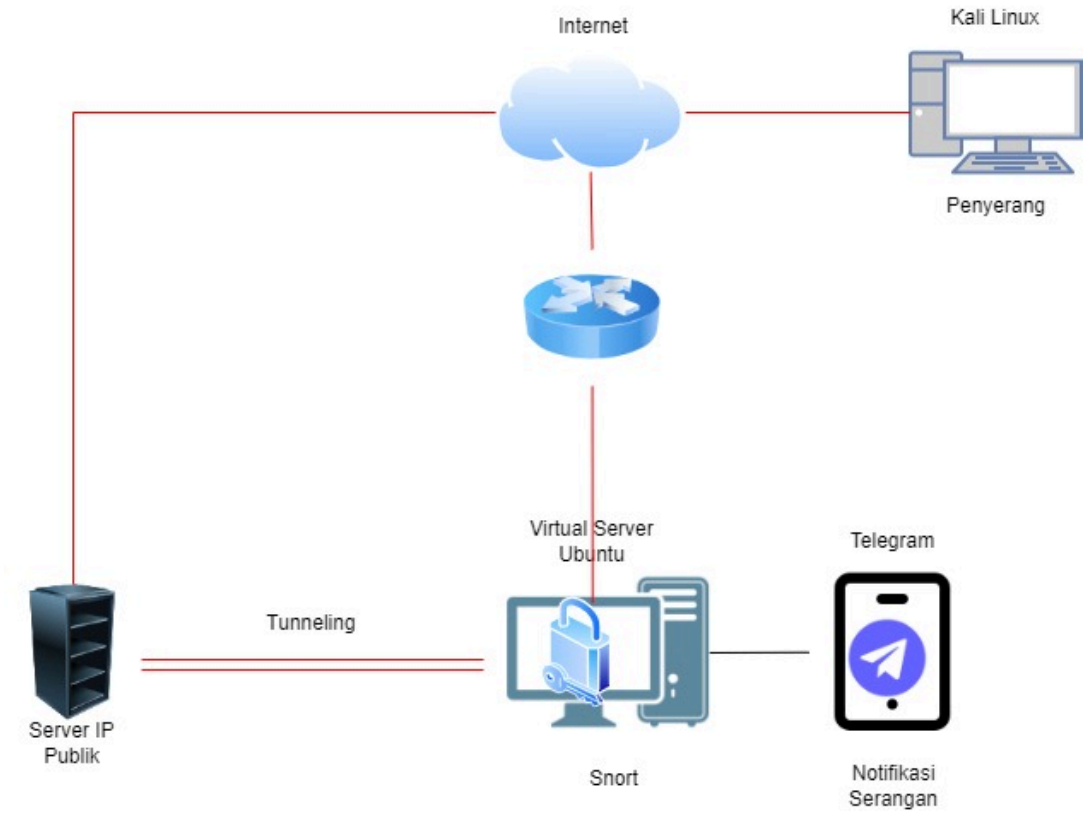
Pada tahapan desain peneliti membuat perancangan topolgi jaringan yang akan digunakan untuk implementasi aplikasi snort sebagai sistem keamanan *server*. Tahapan ini juga mencakup perancangan *rules* pada snort yang spesifik untuk mendeteksi serangan yang telah dilakukan identifikasi pada tahapan analisis seperti aturan untuk mendeteksi serangan DDoS, *SQL Injection*, dan *Port Scanning*. Tahap desain juga mencakup integrasi notifikasi Telegram, yang akan mengirimkan peringatan kepada administrator jaringan segera setelah serangan terdeteksi. Untuk lebih jelasnya berikut merupakan gambaran dari topologi jaringan yang digunakan pada penelitian ini:



Gambar 3.2 Topologi Jaringan IP Lokal

Pada gambar 3.2 diatas merupakan gambar dari perancangan topologi jaringan yang akan dibuat pada penelitian ini. Dari gambaran topologi diatas penyerang melakukan serangan dari device yang terkoneksi dalam satu jaringan dengan *server* selanjutnya penyerang melancarkan aksi serangan kepada *server* dengan tujuan untuk mengeksploitasi jaringan *server* dan melumpuhkan layanan yang tersedia pada *server*. Eksploitasi *server* ini dilakukan dengan cara mengirimkan

beberapa jenis serangan siber menggunakan *tools* bantuan yang dibutuhkan untuk mengganggu kinerja dari *server*, setelah serangan dikirimkan maka *rules* snort yang telah dibuat akan disesuaikan agar bisa mendeteksi serangan siber, setelah *rules* disesuaikan dengan jenis serangan yang ingin dideteksi maka *rules* akan terpicu, selanjutnya setelah *rules* pada snort terpicu oleh serangan yang masuk kepada *server* dan mengindikasikan ada serangan yang masuk sistem akan mengirimkan notifikasi telegram kepada administrator jaringan agar segera melakukan pengecekan terhadap *server*. Ada juga topologi untuk penyerangan yang dilakukan menggunakan ip publik sebagai berikut:



Gambar 3.3 Topologi Jaringan IP Publik

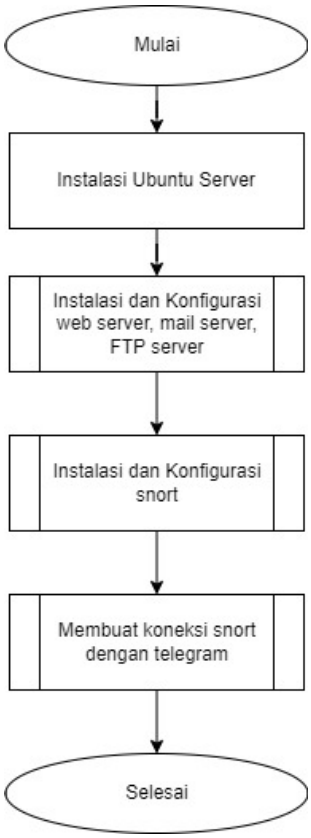
3.4. Simulasi

Pada tahapan simulasi ini penguji melakukan uji coba terhadap sistem yang telah dirancang dengan menjalankan berbagai skenario serangan dari jaringan ip lokal dan menggunakan ip publik. Simulasi yang dilakukan meliputi pengujian terhadap sembilan jenis serangan yang sering terjadi, termasuk *Slow HTTP*, serangan pada protokol POP3, *Port Scanning*, dan lain-lain. Hasil simulasi menunjukkan bagaimana aplikasi snort dapat melakukan tugasnya untuk mendeteksi setiap

serangan dan memblokir ip dari *device* penyerang dan bagaimana integrasi dengan sistem notifikasi Telegram dapat membantu administrator jaringan dalam merespons serangan siber tersebut secara cepat.

3.5. Implementasi

Pada tahapan implementasi aplikasi Snort diinstal dan dikonfigurasi pada server utama yaitu ubuntu *server* sesuai dengan spesifikasi yang telah dirancang. Implementasi akan mencakup penerapan *rules* yang telah dirancang untuk mendeteksi berbagai jenis serangan siber. Setelah itu, bot Telegram diintegrasikan untuk memastikan bahwa setiap pendeteksian serangan siber yang terjadi maka pengiriman notifikasi segera dikirimkan kepada administrator jaringan. Untuk lebih jelasnya bisa pada gambar 3.4 dibawah:



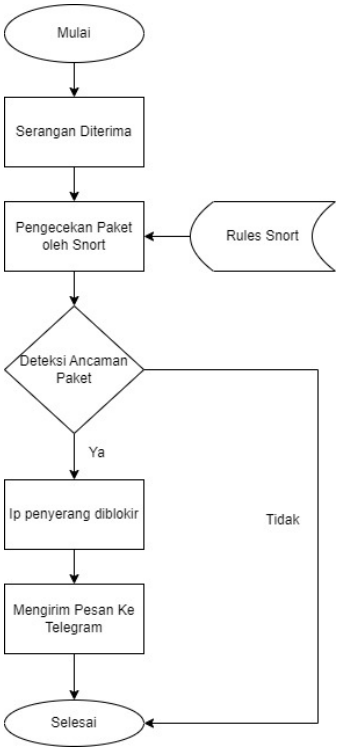
Gambar 3.4 Alur Implementasi Snort

Gambar 3.4 menjelaskan tahapan penerapan snort, lebih jelasnya tahapan yang dibuat sebagai berikut:

1. Langkah pertama yang dilakukan adalah menginstall sistem operasi ubuntu *server* pada *virtualbox* untuk diamankan oleh snort. Ubuntu *server* yang di install pada penelitian ini adalah ubuntu *server* versi 20.04.4.
2. Selanjutnya pada langkah kedua peneliti melakukan beberapa instalasi yang berhubungan dengan layanan *server* pada umumnya agar *server* bisa menyerupai *server* real dengan layanan yang tersedia. Layanan yang diinstall oleh peneliti pada penelitian ini meliputi Web *Server* untuk layanan HTTP, Mail *server* untuk layanan email, DNS *server* untuk pengelolaan domain name, serta FTP *Server* untuk menejemen transfer file.
3. Setelah layanan pada *server* terpenuhi langkah berikutnya adalah untuk pengamanan *server* tersebut. Pada penelitian yang peneliti lakukan sistem keamanannya adalah menggunakan snort. Maka peneliti melakukan instalasi dan konfigursai pada snort agar bisa berjalan dan mengamankan *server* dengan baik.
4. Setelah layanan dan sistem keamanan diinstall dan berjalan dengan baik maka langkah selanjutnya dan langkah terakhir adalah melakukan koneksi antara snort pada ubuntu *server* dengan Instagram menggunakan API telegram hal ini untuk membuat notifikasi dari log snort ke telegram bila mana terjadi serangan.

3.5.1 Cara Kerja Snort

Sebagaimana telah dijelaskan diawal pada penelitian ini peneliti menggunakan snort sebagai sistem keamanan pada *server*. Untuk penjelasan lebih jelas akan dijelaskan pada gambar dibawah bagaimana cara kerja dari sistem keamanan snort agar bisa mendeteksi serangan yang diterima oleh *server* sebagai berikut:



Gambar 3.5 Flowchart Cara Kerja Snort

Pada gambar 3.3 diatas dapat menjelaskan cara kerja dari sistem keamanan snort. Alur yang akan dilalui adalah dimulai Ketika snort dijalankan. Ketika snort dijalankan snort akan mengawasi jalannya lalulintas pada *server* dan jika terjadi anomali atau ada paket jaringan yang mencurigakan melintas kedalam jaringan *server* maka snort akan memeriksanya dengan *rules* yang sudah disesuaikan, apabila *rules* dengan paket yang melintas terdeteksi maka snort akan menganggap bahwa paket jaringan tersebut merupakan sebuah serangan dan akan memberikan notifikasi telegram kepada administrator jaringan lalu ip dari *device* penyerang akan diblokir aksesnya, hasil dari pengawasan jaringan tersebut akan tersimpan pada log snort agar bisa di identifikasi lebih lanjut oleh administrator jaringan yang mengeceknya, namun apabila lalulintas jaringan yang mengarah ke *server* itu aman dan tidak ada paket jaringan yang mencurigakan maka snort tidak akan mengirimkan *alert* apapun dan sistem dianggap normal tidak ada serangan.

3.6. Monitoring

Monitoring adalah langkah penting dalam manajemen keamanan jaringan, terutama untuk memastikan bahwa sistem deteksi intrusi seperti snort berfungsi dengan optimal. Dalam penelitian ini, proses monitoring dilakukan untuk mengawasi

lalu lintas jaringan secara langsung dan untuk memastikan bahwa snort dapat mendeteksi potensi ancaman dengan cepat dan akurat. Monitoring juga mencakup evaluasi notifikasi yang dikirimkan oleh bot Telegram kepada administrator jaringan.

3.6.1. Proses Monitoring

Monitoring dilakukan dengan memanfaatkan fitur-fitur yang ada dalam Snort untuk menganalisis lalu lintas jaringan secara terus-menerus. snort diinstall pada server utama dan dikonfigurasi untuk memantau semua lalu lintas yang masuk dan keluar dari jaringan. Proses ini melibatkan beberapa langkah penting:

1. Pengaturan Logging: Snort dikonfigurasi untuk mencatat semua kejadian yang terdeteksi, baik berupa potensi serangan maupun lalu lintas yang mencurigakan. Log ini disimpan dalam format yang mudah dianalisis, memungkinkan administrator untuk meninjau aktivitas jaringan kapan saja.
2. Integrasi dengan Telegram: Untuk mempercepat respons terhadap ancaman, sistem ini terintegrasi dengan bot Telegram. Ketika Snort mendeteksi ancaman berdasarkan *rules* yang telah ditetapkan, sebuah notifikasi otomatis dikirimkan melalui Telegram kepada administrator jaringan. Hal ini memungkinkan tindakan segera dilakukan, bahkan ketika administrator tidak berada di depan sistem monitoring.
3. Analisis Laporan Harian: Laporan harian juga dihasilkan berdasarkan data yang dikumpulkan oleh snort. Laporan ini memberikan gambaran umum mengenai aktivitas jaringan, termasuk jumlah serangan yang terdeteksi, jenis serangan, dan sumber serangan. Analisis ini membantu dalam mengidentifikasi pola atau tren yang mungkin tidak segera terlihat melalui monitoring rutin.

3.6.2. Evaluasi Kerja Monitoring

Untuk memastikan bahwa proses monitoring berjalan sesuai harapan, beberapa metrik kinerja dievaluasi:

1. Kecepatan Pendeteksian: Waktu yang diperlukan oleh Snort untuk mendeteksi ancaman setelah lalu lintas berbahaya masuk ke jaringan diukur dan dianalisis.

2. Efektivitas Notifikasi Telegram: Efektivitas integrasi dengan Telegram dievaluasi dengan mengukur respons time dari administrator setelah menerima notifikasi. Hal ini penting untuk menilai apakah notifikasi Telegram benar-benar mempercepat tindakan pencegahan dan mitigasi.
3. Stabilitas dan Performa Sistem: Selama proses monitoring, stabilitas dan performa keseluruhan dari sistem juga diawasi untuk memastikan bahwa penambahan fitur notifikasi dan logging tidak memperlambat server atau menyebabkan masalah operasional lainnya.

3.7. Skema Pengujian

Pada bagian skema pengujian serangan dalam penelitian ini berisikan langkah-langkah untuk mensimulasikan serangan siber yang bertujuan untuk mengevaluasi efektifitas *rules* yang dibuat pada snort. Pengujian yang dilakukan mencakup serangan yang umumnya dilancarkan pada keamanan jaringan dengan tujuan mengganggu layanan pada *server*. Ujicoba serangan ini dilakukan sebanyak 30 kali setiap jenis serangan, Untuk penjelasan yang lebih detail adalah sebagai berikut:

1. Pengujian dengan menggunakan DDoS. Pada pengujian serangan DDoS dilakukan dengan cara mengirimkan banyak permintaan kepada *server* yang bertujuan untuk membuat sumber *server* penuh yang akan mengakibatkan layanan terganggu atau bahkan hingga terhenti. Setelah serangan disimulasikan maka kinerja snort akan di evaluasi dalam mendeteksi serangan DDoS.
2. Pengujian dengan menggunakan paket ICMP. Pada pengujian serangan paket ICMP ini akan disimulasikan dengan mengirimkan paket ICMP berukuran besar kepada *server* tujuan, simulasi serangan akan dievaluasi apakah *snort* bisa mendeteksi serangan dengan jenis ICMP.
3. Pengujian serangan menggunakan jenis serangan *Port Scanning*. Pada pengujian ini akan disimulasikan dengan cara melakukan *scanning* kepada *server* tujuan dengan tujuan mengetahui *port* apa saja yang terbuka pada *server*. Setelah simulasi dilakukan maka akan dievaluasi hasil pengujin dengan jenis serangan *port scanning*.

4. Pengujian serangan menggunakan jenis serangan *ipspoofing*. Pada pengujian jenis serangan ini dilakukan dengan cara memantau paket jaringan yang masuk kedlam *server* yang bersasal dari ip palsu yang bisa mengancam keamanan *server*. Setelah melakukan simulasi serangan *ipspoofing* maka hasilnya akan dievaluasi apakah bisa mendeteksi serangan tersebut atau tidak.
5. Pengujian serangan menggunakan jenis serangan SNMP (*Simple Network Management Protocol*). Pada pengujian jenis serangan SNMP ini dilakukan dengan cara mengirimkan beberapa paket jaringan yang bisa mengancam keamanan *server*. Setelah simualsi pengujian dilakukan maka hasil dari pengujian tersebut akan dievaluasi apakah *snort* bisa mendeteksinya atau tidak.
6. Pengujian serangan menggunakan jenis serangan *slowhttp* . pada jenis serangan ini akan dilakukan dengan bantuan *tools slowhttptest* yang mana paket serangan akan diikirmkan kepada *server* yang bertujuan untuk menghabiskan sumberdaya pada *server* tersebut. Setelah jenis serangan ini disimulasikan maka hasilnya akan dievaluasi apakah *snort* akan mendeteksi serangan tersebut atau tidak.
7. Pengujian dengan menggunakan jenis serangan SQL *Injection*. Pada jenis serangan ini akan disimulasikan dengan mengirimkan atau menyuntikan perintah SQL yang berisikan *input* biasanya berisikan *input* untuk mencoba *username* dan *password.* Setelah melakukan simulasi serangan SQL *Injection* maka akan dilakukan evaluasi mengenai kinerja *snort* apakah bisa mendeteksi serangan SQL *Injection* atau tidak.
8. Pengujian serangan dengan menggunakan jenis serangan *cross site scripting* XSS. Pada pengujian serangan jenis ini dilakukan dengan cara mengirimkan atau menyuntikan *script* berbahaya kepada web yang dilayani oleh *web server* pada *server* tujuan. Setelah melakukan simulasi serangan XSS maka akan dilakukan evaluasi mengenai kinerja *snort* apakah bisa mendeteksi serangan XSS atau tidak.
9. Pengujian serangan dengam jenis serangan POP3. Pada pengujian untuk menguji serangan POP3 dilakukan dengan cara mengrimkan paket berbahaya, tujuan dari serangan ini adalah untuk mendapatkan isi email dari

server tujuan. Setelah melakukan simulasi serangan POP3 maka akan dilakukan evaluasi mengenai kinerja *snort* apakah bisa mendeteksi serangan POP3 atau tidak.

3.8 Spesifikasi Alat

Pada tahapan ini akan dijelaskan mengenai spesifikasi perangkat keras dan perangkat lunak yang digunakan oleh penulis dalam penelitian ini. Spesifikasi alat yang tepat sangat penting untuk memastikan bahwa sistem yang diimplementasikan, termasuk Snort sebagai sistem keamanan *server* dan integrasi dengan Telegram, dapat berjalan dengan optimal dan memberikan hasil yang akurat dalam melakukan pendeteksian serta mencegah serangan siber. Rincian detail dari setiap komponen yang terlibat, baik dari sisi server maupun perangkat yang digunakan untuk pengujian dan simulasi serangan. Berikut merupakan spesifikasi perangkat keras dan lunak yang peneliti pakai:

Tabel 3.1
Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi	Catatan
1	PC *server*	Intel Core i5-8250U CPU 1.60GHz 1.80 GHz, NVIDIA GeForce MX130 GPU, 8GB RAM	Spesifikasi pc yang dipakai ubuntu *server*
2	PC *Attacker*	Intel Core i5-8250U CPU 1.60GHz 1.80 GHz, NVIDIA GeForce MX130 GPU, 8GB RAM	Spesifikasi pc yang dipakai kali linux

Tabel 3.2
Spesifikasi Perangkat Lunak

No	Perangkat Lunak	Versi	Deskripsi
1	Sistem operasi server	Ubuntu 22.04.4 LTS	Sistem operasi ini merupakan sistem operasi yang digunakan server pada penelitian ini
2	Sistem operasi penyerang	Kali GNU/Linux Rolling 2024.2	Sistem operasi ini merupakan sistem operasi yang digunakan penyerang pada penelitian ini
3	Snort	*Version* 2.9.15.1 GRE	Aplikasi yang digunakan untuk mendeteksi dan memblokir serangan.
4	*wireshark*	*Wireshark* 3.4.3 (Git v3.4.3 *packaged as* 3.4.3-1)	Aplikasi yang digunakan untuk memantau paket jaringan.
5	*Hping3*	*version* 3.0.0-alpha-2	*Tools* untuk mengirim serangan pada penelitian ini.
6	*Web server*	Apache/2.4.52	Versi apache yang digunakan pada penelitian ini.
7	FTP Server	vsFTPd 3.0.5	Versi FTP yang digunakan pada penelitian ini.
8	*Mail Server*	*Version 3.6.4*	Versi mail server yang digunakan pada penelitian ini.