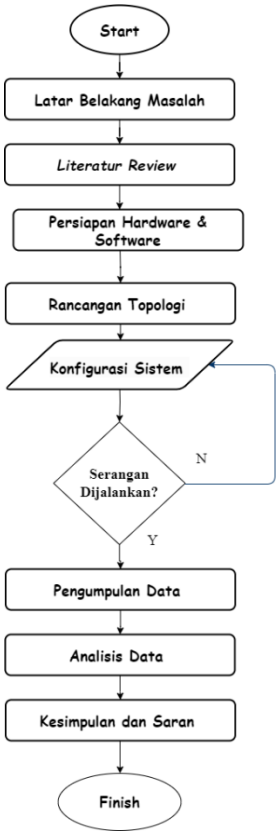
BAB 3
METODE PENELITIAN

Metodologi yang digunakan dalam penelitian ini termasuk sebuag diagram alur yang menggambarkan langkah – langkah yang dilalui selama penelitian. Selain itu, menggambarkan langkagh – langkah yang dilalui selama penelitian. Selain itu, penelitian ini juga menggunakan berbagai perangkat pendukung untuk membantu dalam proses penelitian. Topologi dalam konteks ini berperan sebagai sumber data yang akan digunakan dalam penelitian.

3.2 ALUR PENELITIAN

Penelitian ini ditujukan untuk melakukan simulasi melalui *software* VirtualBox untuk melihat kinerja dari IDS *Snort* dalam mendeteksi serangan DoS dengan kontrol SDN Floodlight pada jaringan virtual. Penelitian dilakukan dengan beberapa tahap seperti gambar 3.1 alur penelitian.



Gambar 3. 1 Alur Penelitian

Alur penelitian yang disajikan pada gambar 3.1, memiliki langkah – langkah yang terstruktur dan sistematis. Penelitian dimulai dengan tahap mulai, yang menandai awal perjalanan penelitian. Setelah itu, langkah definisi masalah membantu dalam merumuskan pernyataan masalah yang akan diselesaikan dalam penelitian. Tahap *literatur review* dilakukan untuk memahami kerangka teoritis dan penelitian terdahulu yang mendukung penelitian ini. Selanjutnya pada tahap desain penelitian, metodologi penelitian dirancang dengan cermat untuk memastikan pengumpulan data yang efisien dan analisis yang tepat. Persiapan perangkat keras dan perangkat lunak langkah yang krusial dimana semua sumber daya yang dibutuhkan untuk penelitian disiapkan. Kemudian dalam tahap rancangan topologi jaringan melibatkan perancangan infrastruktur jaringan yang akan digunakan dalam penelitian ini. Implementasi sistem adalah langkah berikutnya, dimana sistem yang diperlukan, termasuk IDS *Snort* dan elemen SDN *Floodlight*, diimplementasikan.

Pengujian sistem merupakan tahap penting, yang mencakup simulasi serangan DoS pengecekan keberhasilan simulasi. Jika simulasi berhasil, penelitian melanjutkan ke tahap pengumpulan data, dimana data relevan dikumpulkan. Namun, jika simulasi tidak berhasil, langkah kembali ke Implementasi sistem dilakukan perbaikan. setelah data terkumpul, analisis data dilakukan pada tahap analisis data untuk menghasilkan temuan yang relevan. Kesimpulan dan saran adalah tahap akhir dimana kesimpulan penelitian dibuat dan saran diberikan untuk pengembangan lebih lanjut. Kemudian pada tahap finish menandai penyelesaian seluruh penelitian, termasuk penyusunan laporan penelitian yang mencerminkan hasil dan temuan dari seluruh perjalanan penelitian ini. Dengan alur penelitian yang terstruktur ini, peneliti memiliki panduan yang jelas untuk menjalankan penelitian dengan efektif dan sistematis.

3.1 ALAT YANG DIGUNAKAN

3.1.1 Perangkat Keras

pada penelitian ini menggunakan perangkat keras yaitu laptop dengan sistem operasi ,Windows 11 dan didukung oleh prosesor Intel® Core™ i5-1035G7 dengan kecepatan 1.20 GHz. Laptop ini memiliki RAM sebesar 16 GB dan menggunakan sistem operasi 64-bit. Untuk penyimpanan, laptop ini dilengkapi

dengan *Solid State Drive* (SSD) berkapasitas 512 GB. Dengan spesifikasi ini, laptop ini diharapkan dapat memberikan performa yang baik pada penelitian ini, seperti pada Tabel 3.1.

Tabel 3. 1 Spesifikasi laptop

OS	Windows 11
Processor	Intel® Core™ i5-1035G7 CPU @ 1.20 GHz
Installed RAM	16 GB
System type	64-*bit operating system*
Storage (SSD)	512GB

3.1.2 Perangkat Lunak

Penelitian ini menggunakan beberapa perangkat lunak yang tercantum dalam Tabel 3.2, untuk mendukung simulasi dan pengujian jaringan. Pertama, VirtualBox VM versi 7.0.14 digunakan untuk simulasi jaringan. *Linux Ubuntu* 14.04 berfungsi sebagai emulator jaringan, controller, dan IDS. Mininet versi 2.3.0 digunakan sebagai emulator jaringan SDN, sementara Floodlight versi 1.2 berperan sebagai controller SDN. Terakhir, Snort versi 3.1.78.0 digunakan sebagai IDS untuk mendeteksi berbagai serangan jaringan. Kombinasi perangkat lunak ini memungkinkan saya untuk mensimulasikan, menguji, dan menganalisis jaringan secara menyeluruh dalam penelitian ini.

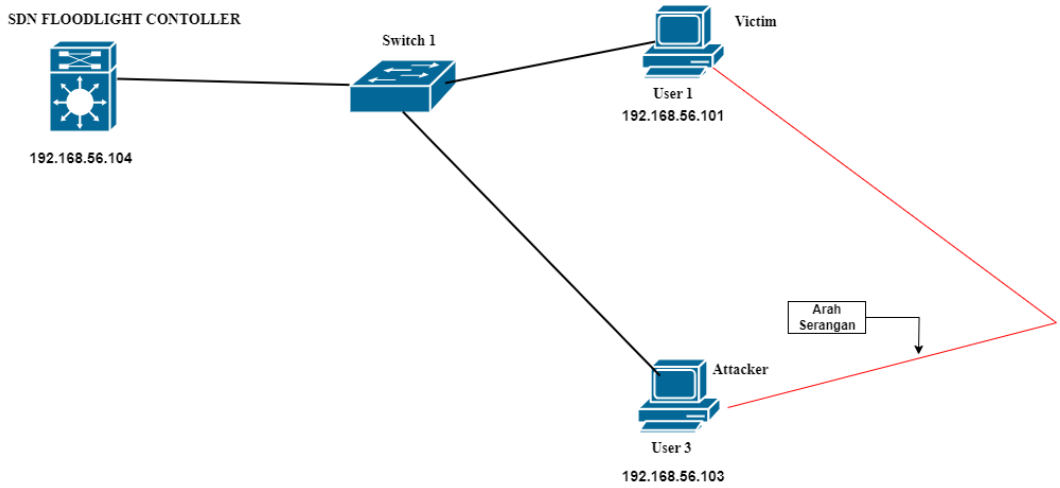
Tabel 3. 2 Perangkat Lunak

No	Software	Versi	Fungsi
1	*VirtualBox* VM	7.0.14	Simulasi Jaringan
2	Linux Ubuntu	18.04	Emulator jaringan, controller dan IDS
3	Mininet	2.3.0	Emulator jaringan SDN
4	Floodlight	1.2	Controller
5	Snort	3.1.78.0	IDS

3.3. RANCANGAN TOPOLOGI JARINGAN

Rancangan topologi jaringan pada penelitian ini yaitu jaringan berbasis SDN dengan sistem keamanan IDS. Dalam rancangan topologi jaringan ini terdapat 1 perangkat *controller* sebagai SDN *Floodlight Controller*, kemudian 1 *switch*, dan 2 *user*. Diantaranya *user* pertama sebagai penyerang (*attacker*), *user* 1 berindak sebagai korban pola serangan pada penelitian ini ditandain dengan garis berwarna merah. Selanjutnya alamat *Internet Protocol* (IP) dari masing – masing perangkat untuk perangkat *controller* dengan IP *Address* (192.168.56.104),

kemudian *user* 1 dengan IP *Address* (192.168.56.101), *user* 2 sebagai *attacker* dengan IP *Address* (192.168.56.103). setelah melakukan perancangan topologi jaringan dan menenetukan alamat IP pada masing – masing perangkat.



Gambar 3. 2 Topologi Jaringan

Selanjutnya mengkonfigurasi pada masing – masing perangkat menggunakan *software virtualbox*. Pada penelitian ada 4 virtual mesin (vm) yang digunakan, untuk perangkat *contoller* menggunakan aplikasi emulasi jaringan yaitu mininet, dan untuk serangan nya dikonfigurasi menggunakan *tolls hping3* dengan metode serangan *floodattack*, kemudian IDS *Snort* dikonfigurasi pada *user* 1 yang sebagai target serangan dalam penelitian ini, arah serangan dalam topologi ini ditandai dengan garis berwarna merah seperti pada Gambar 3.2 dan untuk tabel alamat IP seperti pada Tabel 3.3.

Tabel 3. 3 IP Address

Perangkat	*Interface*	*IP Address*	*Subnet Mask*	*Gateway*
User 1	enp0s8	192.168.56.101	255.255.255.0	192.168.56.1
Attacker	enp0s8	192.168.56.103	255.255.255.0	192.168.56.1
SDN	enp0s8	192.168.56.104	255.255.255.0	192.168.56.1

3.4 KONFIGURASI SISTEM

Pada tahap implementasi sistem merupakan langkah kunci dalam penelitian ini, dimana penulis menjalankan rancangan topologi jaringan yang telah direncanakan sebelumnya. Pada tahap ini, fokus utama penulis adalah

mengintegrasikan berbagai komponen yang diperlukan, termasuk IDS *Snort*, elemen SDN *Floodlight Controller*. Pertama – tama, penulis memulai dengan membangun lingkungan jaringan. Ini melibatkan konfigurasi perangkat lunak yang diperlukan untuk menciptakan jaringan virtual yang kompleks. Selanjutnya, penulis menginstal dan mengkonfigurasi IDS *Snort* pada user 1.IDS *Snort* akan bertanggung jawab untuk memantau lalu lintas jaringan yang melewati user tersebut dan mendeteksi tanda – tanda serangan DoS.

Selain itu, penelitian ini juga mengintegrasikan elemen SDN *Floodlight* ke dalam jaringan. *Floodlight Controller* akan mengendalikan aliran lalu lintas jaringan. Pengujian koneksi dan interaksi antara semua komponen dalam sistem adalah langkah penting selama tahap simulasi ini untuk memastikan bahwa semuanya berjalan dengan lancar. Sebelum melanjutkan ke tahap pengujian serangan DoS, penulis melakukan uji fungsionalitas IDS *Snort* untuk memastikan bahwa IDS dapat mendeteksi serangan dengan benar. Selama tahap ini, penulis juga mendokumentasikan semua langkah yang telah penulis ambil, termasuk konfigurasi IDS *Snort* dan integrasi SDN *Controller*.

Tahap simulasi sistem adalah fondasi yang kuat untuk seluruh penelitian ini, karena sistem yang telah penulis bangun menjadi pusat dari pengujian serangan SDN dan pengumpulan data. Dengan tahap ini berhasil, penulis siap untuk melanjutkan ke tahap berikutnya dengan keyakinan bahwa infrastruktur yang telah penulis bangun akan mendukung penelitian dengan efektif dan terstruktur.

3.4.1 Konfigurasi *IP Addresss* Pada Masing-Masing *User*

Penetapan alamat IP pada setiap pengguna dalam jaringan menggunakan IP *class* C dengan *Subnet mask* /24 berarti 24 *bit* pertama dari alamat IP merupakan bagian dari alamat jaringan dan 8 *bit* terakhir adalah bagian dari alamat *host*. Konfigurasi ini dilakukan dalam *directory netplan* di masing-masing Virtual Mesin (VM). Proses ini mencakup beberapa langkah penting yang memastikan setiap VM mendapatkan alamat IP yang tetap dan dapat berkomunikasi dengan perangkat lain dalam jaringan virtual. Pertama, alamat IP ditetapkan pada setiap pengguna. *User* 1 mendapatkan alamat IP 192.168.56.101, *attacker* 192.168.56.103, dan SDN

192.168.56.104. Setiap alamat IP ini unik untuk masing-masing pengguna dalam jaringan, memastikan tidak ada konflik alamat IP.

Selanjutnya, konfigurasi dilakukan dalam *directory netplan* pada setiap VM. Netplan adalah utilitas di sistem operasi berbasis ubuntu yang digunakan untuk mengelola dan mengkonfigurasi jaringan. Konfigurasi ini dilakukan pada antarmuka jaringan yang disebut enp0s8, di mana setiap VM harus memiliki file konfigurasi netplan yang sesuai untuk menetapkan alamat IP yang telah ditentukan. Misalnya, file konfigurasi "netplan /etc/netplan/01-netcfg.yaml" dapat berisi pengaturan alamat IP, *gateway*. Dalam *file* ini, konfigurasi alamat IP dan pengaturan lain yang relevan diatur sedemikian rupa untuk memastikan bahwa VM dapat berkomunikasi dengan jaringan lain secara efektif.

Selain itu, *Dynamic Host Configuration Protocol* (DHCP) diaktifkan pada setiap VM. Aktivasi DHCP ini sangat penting karena memastikan bahwa setiap alamat IP yang diberikan kepada VM tidak berubah setiap kali VM di-*restart* atau terhubung kembali ke jaringan. DHCP juga mengelola distribusi alamat IP sehingga tidak ada dua perangkat yang memiliki alamat IP yang sama dalam jaringan. Hal ini sangat penting untuk menjaga konsistensi dan integritas jaringan, menghindari konflik alamat IP yang bisa menyebabkan gangguan komunikasi antar perangkat dalam jaringan.

Setelah konfigurasi selesai, antarmuka jaringan enp0s8 diaktifkan dengan menjalankan perintah "sudo netplan apply", yang menerapkan semua konfigurasi jaringan yang telah ditetapkan dalam file netplan. Dengan menjalankan perintah ini, semua pengaturan jaringan yang telah dibuat akan diimplementasikan dan antarmuka jaringan akan mulai beroperasi sesuai dengan konfigurasi yang telah ditetapkan. Langkah ini memastikan bahwa setiap VM terhubung ke jaringan dengan pengaturan IP yang benar dan dapat berfungsi sebagaimana mestinya.

Terakhir, penting untuk memverifikasi bahwa setiap VM telah menerima alamat IP yang benar dan dapat berkomunikasi dengan perangkat lain dalam jaringan. Verifikasi ini bisa dilakukan dengan menggunakan perintah "ping" untuk memastikan bahwa setiap VM memiliki alamat IP yang sesuai dan bisa berkomunikasi dengan VM lain dalam jaringan. Secara keseluruhan, proses ini

melibatkan penetapan alamat IP, konfigurasi dalam directory netplan, aktivasi DHCP, dan pengaktifan antarmuka jaringan enp0s8 pada setiap VM, memastikan setiap VM dapat berfungsi dengan optimal dalam jaringan yang terkonfigurasi dengan baik, rincian konfigiurasinya seperti di bawah ini.

```
Konfigurasi IP User 1 (Victim)
cd /etc/netplan
nano 00-installer-config.yaml
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses:
        - 192.168.56.101/24

Konfigurasi User 2 (Attacker)
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses:
        - 192.168.56.103/24
Konfigurasi user 3 (SDN)
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses:
        - 192.168.56.104/24
```

3.4.2 Konfigurasi SDN *Floodlight Controller* Dan *Mininet*

Konfigurasi SDN *Floodlight* dan *Mininet* melibatkan beberapa tahapan penting. Pertama, *Docker* perlu diinstal sebagai lingkungan untuk menjalankan *Floodlight*. Setelah *Docker* terinstal, *Floodlight* diunduh dan dijalankan dalam *Docker*, yang memungkinkan pengelolaan lalu lintas jaringan secara efisien dan terisolasi. Selanjutnya, antarmuka *Floodlight* dapat diakses melalui *browser* untuk memantau dan mengelola aktivitas jaringan. Untuk SDN dan *mininet* ini di konfigurasi pada *user* 4 di VM, Rincian konfigurasinya disajikan di bawah ini.

```
sudo apt install docker.io
docker pull pierrecdn/floodlight
docker start floodlight_new
```

Setelah *Floodlight* berjalan dengan baik, *Mininet* di *install* untuk mensimulasikan jaringan virtual yang akan berinteraksi dengan *Floodlight* sebagai *controller*. *Mininet* kemudian dijalankan dengan konfigurasi tertentu, menghubungkannya ke *Floodlight* dan mengatur topologi jaringan serta protokol komunikasi yang digunakan. Proses ini memungkinkan pengujian dan pengembangan skenario jaringan SDN secara fleksibel dan terkontrol, rincian konfigurasinya disajikan di bawah ini.

```
sudo apt install mininet
mn --controller=remote,ip=192.168.56.104,port=6653 --switch ovsk,protocols=OpenFlow13 --topo
single,3 --mac --ip 192.168.56.101/24 --ip 192.168.56.102/24 --ip 192.168.56.103/24
```

3.4.3 Konfigurasi IDS *Snort*

Snort adalah sebuah sistem deteksi intrusi (IDS) *open-source* yang digunakan untuk mendeteksi dan menganalisis lalu lintas jaringan yang mencurigakan. Untuk menginstal *Snort* di sistem berbasis *Ubuntu*, langkah pertama yang harus dilakukan adalah membuka terminal dan menjalankan perintah "*sudo apt install snort*". Perintah ini akan mengunduh dan menginstal *Snort*. Selain itu, untuk memonitor *log* sistem secara efektif, penting juga untuk menginstal "*sysstat*" yang menyediakan alat untuk mengumpulkan dan melaporkan statistik kinerja sistem. Perintah yang digunakan adalah sudo apt install "*sysstat -y*", di mana opsi "*-y*" digunakan untuk menyetujui otomatis semua *prompt* selama instalasi. Dengan menginstal *Snort* dan *sysstat*, administrator jaringan dapat memonitor aktivitas jaringan secara *real-time* dan mendapatkan laporan kinerja sistem yang komprehensif, sehingga mempermudah dalam mendeteksi dan merespons ancaman keamanan. IDS *Snort* dapat dikonfigurasi untuk mendeteksi pola serangan yang sudah dikenal seperti port *scanning*, *buffer overflow*, dan berbagai jenis serangan lainnya. Dengan *sysstat*, administrator dapat melihat penggunaan CPU, memori, dan I/O disk secara detail, rincian konfigurasinya disajikan di bawah ini.

```
sudo apt install snort
sudo apt install sysstat -y
```

3.4.3 Konfigurasi *Web Server*

Web server adalah komponen penting dalam infrastruktur internet yang memungkinkan pengiriman konten *web* kepada pengguna. Salah satu *web server* yang populer dan sering digunakan adalah *Apache*. Untuk menginstal *Apache* di sistem operasi berbasis *Ubuntu*, Setelah instalasi, *Apache* secara *default* akan mendengarkan pada *port* 80 untuk menerima permintaan HTTP. Halaman *web default Apache* dapat diakses dengan membuka *browser* dan mengetikkan alamat IP. *Web server* di *install* pada *user* 1, di dalam VM *Virtualbox*, rincian konfigurasi disajikan di bawah ini.

```
sudo apt install apache2
```

3.4.4 Konfigurasi *Tolls*

Install tolls, bahwa Perintah *sudo apt update* dan *sudo apt install hping3* digunakan untuk *hping3* di sistem operasi berbasis *ubuntu. Sudo apt update* berfungsi untuk memperbarui daftar paket di sistem *ubuntu*, sementara *sudo apt install hping3* digunakan untuk menginstal *tolls hping3*. *hping3* adalah *tolls* yang serbaguna yang dapat digunakan untuk menguji jaringan dan mengirim paket ke tujuan tertentu untuk mengukur respons jaringan, tetapi perlu diingat *tolls* ini untuk melakukan serangan DoS, rincian konfigurasinya disajikan di bawah ini.

```
sudo apt install hping3 -y
```

3.5 PENGUJIAN SISTEM

3.5.1 Skenarion Pengujian

Pengujian sistem ini akan berfokus pada menguji seberapa baik sistem dapat mendeteksi dari serangan DoS, khususnya serangan *flood attack* yang ditujukan ke *user* 1 sebagai target yang di dalam jaringan yang telah disiapkan. Pengujian akan melibatkan pengiriman serangan dengan menggunakan *tolls hping*3. pengujian ini juga menggunakan 10 skrenario dengan jumlah paket dan ukuran paket yang berbeda, seperti pada tabel 3.4 Selain itu, untuk mengetahui kinerja dari IDS *Snort* pada penelitian ini menggunakan beberapa parameter diantaranya waktu respon, akurasi deteksi dan beban kerja jaringan dijelaskan pada tabel 4.5 metrik pengukuran. Setiap skenario 30 data yang di ambil. Dimana nantinya akan dicatat

untuk mendapatkan rata rata nya dan di analisis, dan mengevalusi kinerja IDS *Snort*. Penelitian ini diharapkan dapat memberikan wawasan mendalam tentang efektivitas IDS Snort dalam mendeteksi dan merespons serangan flood attack, serta dampaknya terhadap kinerja jaringan secara keseluruhan. Hasil dari pengujian ini diharapkan dapat memberikan panduan praktis untuk meningkatkan keamanan jaringan dengan menggunakan IDS Snort.

3.5.2 Konfigurasi Dan Parameter Serangan

Pengujian ini bertujuan untuk menguji ketahanan sistem terhadap serangan DoS. Penulis akan menjalankan serangan DoS terhadap *user* 1 menggunakan serangan *flood attack* dengan parameter yang bervariasi. Selain itu, pengujian ini juga akan memeriksa sejauh mana IDS *Snort* dapat mendeteksi serangan-serangan tersebut. Konfigurasi umum untuk pengujian yang akan dilakukan terfokus pada infrastruktur jaringan yang sudah disiapkan. Pengujian ini akan melibatkan simulasi serangan DoS, khususnya serangan *flood attack*. Serangan ini akan dilancarkan menggunakan *hping3*. Tujuan dari pengujian ini adalah untuk mengevaluasi konektivitas IDS *Snort* dalam mendeteksi dan merespons serangan *flood Attack* pada jaringan yang telah disiapkan, seperti pada tabel 3.4, dan tabel 3.5.

Tabel 3. 4 Perintah hping3

Parameter	Keterangan
-c	jumlah paket yang akan dikirimkan sebagai bagian dari serangan.
-d	ukuran data dari setiap paket yang akan dikirimkan.
-S	Aktifkan flag SYN untuk menginisiasi koneksi TCP.
-w 64	Ukuran window TCP yang diatur
-p	port target yang akan diserang
-c	Kirimkan paket secepat mungkin tanpa menunggu respons.

Tabel 3. 5 Parameter Pengujian

Pengujian ke-	Jumlah Paket	Ukuran Data (*Byte*)
1	100	64
2	500	128

Pengujian ke-	Jumlah Paket	Ukuran Data (*Byte*)
3	1000	256
4	2000	512
5	5000	1024
6	10000	2048
7	15000	4096
8	20000	8192
9	30000	16384
10	50000	32768

Serangan DoS menggunakan metode serangan *flood attack* adalah taktik yang dimanfaatkan oleh penyerang untuk mengganggu ketersediaan layanan pada suatu jaringan atau sistem komputer dengan membanjiri target dengan lalu lintas data yang tidak perlu. Serangan ini dilakukan dengan cara mengirimkan jumlah besar paket data secara simultan ke target, membebani sumber daya jaringan atau sistem yang membuatnya tidak mampu menangani lalu lintas yang berlebihan tersebut. Metode ini bisa mengakibatkan layanan yang menjadi target menjadi tidak responsif atau bahkan mengalami kegagalan total. Serangan *Flood Attack* biasanya dilakukan menggunakan protokol *Transmission Control Protocol* (TCP) atau *User Datagram Protocol* (UDP), *Domain Name System* (DNS), *Internet Control Message Protocol* (ICMP), *Hypertext Transfer Protocol* (HTTP), dan *Simple Mail Transfer Protocol* (SMTP).Pada percobaan ini sebagai target adalah user 1, rincian konfigurasinya disajikan di bawah ini.

```
hping3 -c 100 -d 64 -S -w 64 -p 80 --flood 192.168.56.101
```

Perintah "hping3 -c 100 -d 64 -S -w 64 -p 80 --flood 192.168.56.101" digunakan untuk mengirimkan paket-paket TCP/IP dengan spesifikasi tertentu menggunakan *hping3*, sebuah *tools* untuk pengujian jaringan dan keamanan. Dalam perintah ini, "-c 100" menentukan jumlah paket yang dikirim adalah 100, sedangkan "-d 64" mengatur ukuran payload data dalam paket menjadi 64 *byte*. "Flag -S" menandakan bahwa paket *Synchronize* (SYN) akan dikirim untuk memulai koneksi TCP, dan "-w 64" mengatur ukuran window TCP menjadi 64 *byte*. *Port* tujuan yang ditetapkan adalah port 80, yang biasanya digunakan untuk HTTP, dengan parameter -p 80. Opsi "--flood" menginstruksikan hping3 untuk mengirim paket-paket

tersebut secepat mungkin tanpa menunggu balasan, yang dikenal sebagai teknik *flooding*.

Alamat IP tujuan dari paket-paket ini adalah 192.168.56.101. Secara keseluruhan, perintah ini digunakan untuk mengirim 100 paket SYN ke port 80 pada IP 192.168.56.101 dengan cara yang sangat cepat, sering digunakan untuk menguji ketahanan server terhadap serangan SYN *flood*, sebuah jenis serangan DoS yang berusaha membuat server tidak responsif dengan membanjirinya dengan permintaan koneksi. SYN *flood*. SYN *flood* adalah jenis serangan DoS yang berusaha membuat server tidak responsif dengan membanjirinya dengan permintaan koneksi. Dengan mengirimkan sejumlah besar paket SYN dalam waktu singkat, penyerang dapat menyebabkan server kehabisan sumber daya untuk menangani permintaan sah lainnya.

3.5.3 Mengukur Kinerja IDS

Dalam mengukur efektivitas IDS *Snort* dari serangan DoS, tiga parameter matriks utama yang perlu diukur adalah waktu respons, akurasi deteksi, dan beban kerja. Pertama, waktu respons mengacu pada kecepatan sistem dalam mendeteksi dan merespons ancaman. Ini diukur dengan mensimulasikan serangan dan mencatat waktu yang dibutuhkan *Snort* untuk mengidentifikasi ancaman tersebut dan mengambil tindakan, seperti mengeluarkan peringatan. Kedua, akurasi deteksi mengevaluasi seberapa efektif IDS *Snort* dalam membedakan antara lalu lintas normal dan berbahaya. Untuk mengukurnya, lalu lintas jaringan yang beragam diuji, dan respons Snort terhadap setiap jenis lalu lintas tersebut dicatat. Terakhir, beban kerrja berkaitan dengan dampak tambahan yang ditimbulkan oleh IDS *Snort* terhadap jaringan, termasuk konsumsi *bandwidth*, penggunaan CPU, dan memori. Beban kerja ini diukur dengan membandingkan kinerja jaringan (kecepatan, latensi, dan penggunaan sumber daya) saat IDS *Snort* aktif dan non-aktif, seperti pada tabel 3.6.

Tabel 3. 6 Metrik Pengukuran

Parameter	Keterangan	Satuan
Waktu Respon	Waktu yang dibutuhkan IDS Snort untuk mendeteksi dan merespons ancaman setelah simulasi serangan.	detik (s) atau milidetik (ms)

Parameter	Keterangan	Satuan
Akurasi Deteksi	Tingkat ketepatan IDS Snort dalam membedakan lalu lintas normal dan serangan.	persentase (%)
Beban kerja Jaringan	Dampak tambahan yang ditimbulkan IDS Snort terhadap kinerja jaringan, termasuk konsumsi bandwidth, penggunaan CPU, dan penggunaan memori.	% untuk CPU dan memori, Mbps untuk bandwidth

3.6 PENGUMPULAN DATA

Pada tahap pengumpulan data, berbagai informasi dan parameter penting harus dikumpulkan untuk mendukung simulasi IDS berbasis *Snort* dalam mendeteksi serangan DoS pada jaringan virtual menggunakan SDN dengan *Floodlight Controller*. Proses pengumpulan data ini bertujuan untuk menyediakan informasi yang diperlukan untuk konfigurasi dan evaluasi sistem yang dibangun.

3.6.1 SDN Floodlight Controller

Pengumpulan Data SDN dengan *Floodlight Controller* sangat penting dalam konteks simulasi IDS untuk mendeteksi serangan DoS dalam jaringan virtual. *Floodlight Controller* adalah salah satu perangkat lunak yang paling umum digunakan untuk mengelola jaringan SDN. Proses ini dimulai dengan mengumpulkan informasi tentang konfigurasi *Floodlight Controller*, seperti alamat IP, dan kebijakan kontrol jaringan yang diterapkan. Selanjutnya, data tentang aliran lalu lintas yang dikendalikan oleh *Floodlight Controller* juga sangat relevan. Ini termasuk aturan aliran (*flow rules*) yang telah ditetapkan, statistik lalu lintas yang terkait dengan aturan ini, dan keadaan switch yang diatur oleh *Floodlight Controller*. Informasi ini memberikan pemahaman yang mendalam tentang bagaimana jaringan virtual berperilaku dalam mode SDN dan bagaimana respon terhadap perubahan topologi atau serangan potensial.

3.6.2 IDS Snort

Pengumpulan Data IDS *Snort* adalah tahapan krusial dalam menyusun simulasi IDS untuk mendeteksi serangan DoS dalam jaringan virtual. IDS *Snort* merupakan salah satu perangkat lunak yang paling umum digunakan untuk deteksi

intrusi dalam jaringan. Proses ini dimulai dengan mengumpulkan informasi terkait konfigurasi *Snort*, seperti aturan deteksi yang diaktifkan, *preprocessor* yang digunakan, dan konfigurasi *output log*. Selanjutnya, dilakukan pengujian aturan deteksi untuk memastikan efektivitasnya dalam mengidentifikasi serangan DoS, dengan mencatat data log serangan yang berhasil di deteksi. Informasi yang terkumpul dari pengujian ini membantu dalam penyesuaian dan penyempurnaan aturan deteksi untuk meningkatkan ketepatan dan kecepatan respons IDS terhadap serangan. Dengan memahami dan menganalisis data yang diperoleh dari IDS *Snort*, penelitian dapat mengoptimalkan kinerja sistem deteksi mereka dan memastikan bahwa serangan DoS dapat dideteksi dengan akurat dan efisien dalam lingkungan jaringan virtual yang dikendalikan oleh SDN *Floodlight Controller*.

3.6.3 Serangan DoS

Pengumpulan Data Serangan DoS merupakan tahapan penting dalam simulasi IDS untuk mendeteksi serangan DoS dalam jaringan virtual. Langkah awal dalam proses ini adalah mengidentifikasi jenis-jenis serangan DoS yang akan disimulasikan, seperti *flood attack*. Selanjutnya, diperlukan data log paket yang dikirimlah oleh attacker. Informasi ini membantu dalam menyesuaikan konfigurasi IDS untuk mengenali pola-pola serangan yang spesifik. Selain itu, pengumpulan data mencakup pemetaan pola tanda-tanda khas serangan DoS yang akan dideteksi oleh IDS *Snort*, seperti pola paket yang tidak biasa atau lonjakan frekuensi paket dalam waktu singkat. Data tentang lalu lintas normal dalam jaringan juga sangat penting untuk pembandingan, karena membantu dalam membedakan perilaku normal dan anomali yang mungkin menandakan serangan. Dengan memahami karakteristik dan pola-pola serangan DoS serta lalu lintas normal dalam jaringan, peneliti dapat mengoptimalkan konfigurasi IDS untuk mendeteksi dan merespons serangan DoS dengan lebih efektif dalam lingkungan jaringan virtual yang dikontrol oleh SDN.

3.6.4 Evaluasi

Pengumpulan Data Evaluasi merupakan tahap penting dalam proses simulasi IDS untuk mendeteksi serangan DoS dalam jaringan virtual. Di sini,

fokusnya adalah pada pengumpulan informasi terkait dengan metrik kinerja sistem, seperti waktu deteksi serangan, akurasi deteksi, dan beban kerja yang dihasilkan oleh IDS. Data ini membantu untuk mengevaluasi seberapa efektif IDS dalam mendeteksi serangan DoS dan seberapa cepat IDS dapat merespon terhadap serangan tersebut. Selain itu, analisis hasil dari data yang terkumpul juga dilakukan untuk memahami performa keseluruhan sistem. Analisis ini mencakup efektivitas deteksi serangan, serta kemampuan IDS untuk beradaptasi terhadap variasi lalu lintas dan taktik serangan yang muncul. Data evaluasi juga memberikan wawasan tentang bagaimana IDS merespon terhadap perubahan dalam topologi jaringan atau kondisi lingkungan lainnya. Dengan memahami dan menganalisis data evaluasi secara menyeluruh, penelitian ini dapat mengidentifikasi kelemahan dan area yang perlu diperbaiki dalam sistem, sehingga dapat meningkatkan keandalan dan kinerja IDS dalam mendeteksi serangan DoS di lingkungan jaringan virtual yang dikelola menggunakan SDN *Floodlight Controller*.