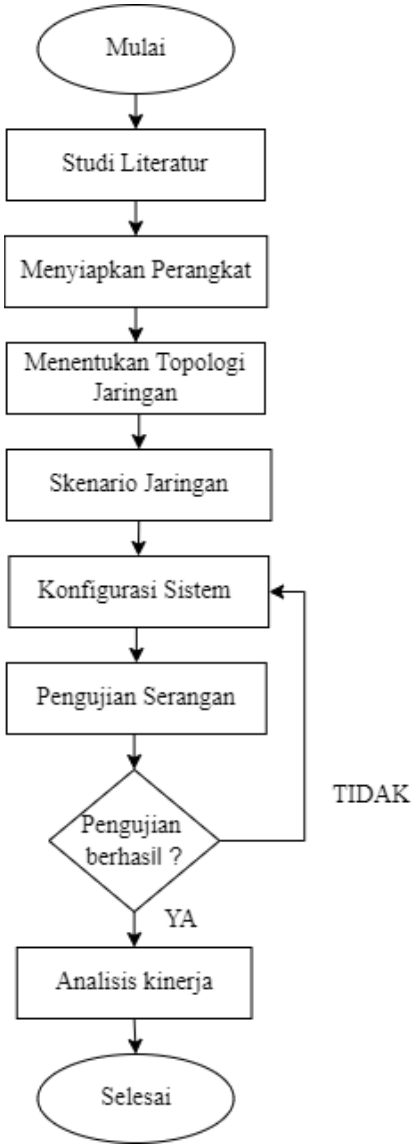
BAB III

METODOLOGI PENELITIAN

3.1 TAHAPAN PENELITIAN

Pada tahapan penelitian ini, penulis menjabarkan kegiatan yang akan dilaksanakan penulis selama proses penelitian berlangsung, dimana tahapan ini digunakan sebagai acuan untuk mencapai tujuan dari hasil penelitian. Adapun tahapan penelitian ditunjukkan pada Gambar 3.1



Gambar 3.1 Tahapan Penelitian

3.1.1 STUDI LITERATUR

Pada tahap ini, penulis melakukan studi literatur dengan tujuan untuk mendukung penelitian yang penulis lakukan. Selanjutnya, penulis juga mendalami apa yang sedang diteliti melalui teori-teori yang diperoleh dari buku, jurnal, *website*, dan penelitian sejenis. Dari tahapan ini juga kemudian dapat ditentukan kebutuhan apa saja yang akan dibutuhkan oleh penulis dalam perancangan sistem.

3.1.2 MENYIAPKAN PERANGKAT

Tahapan selanjutnya yaitu menyiapkan perangkat yang kemudian akan digunakan dalam melakukan konfigurasi sistem hingga sampai ke tahap pengujian sistem. Adapun perangkat yang dibutuhkan antara lain perangkat keras (*hardware*) dan perangkat lunak *(software*), antara lain sebagai berikut:

3.1.2.1 Perangkat Keras (*Hardware*)

1) *Personal Computer* (PC) AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx (with SSE4.2)
2) Kabel *straightover*

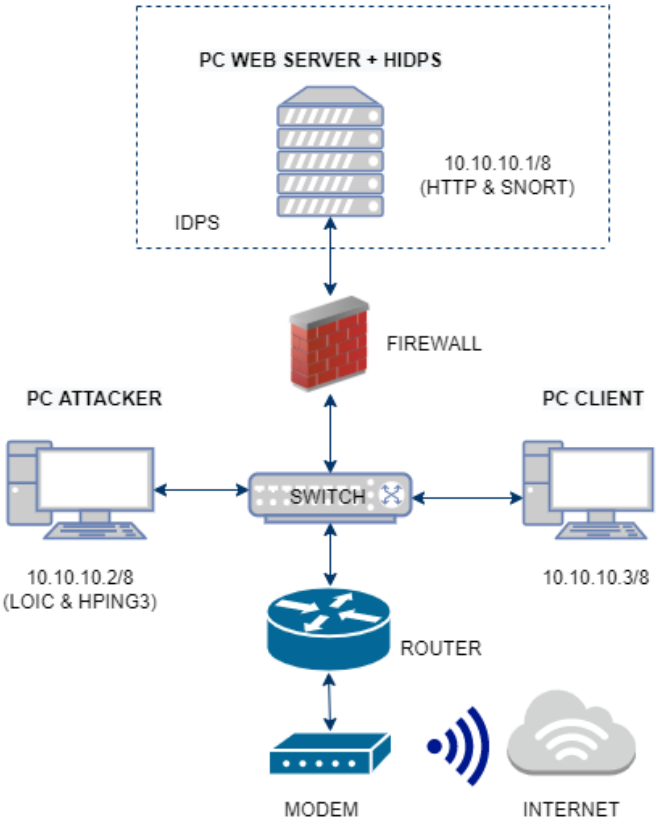
3.1.2.2 Perangkat Lunak (*Software*)

1) Linux Ubuntu 18.04 sebagai sistem operasi yang digunakan untuk membangun server IDPS.
2) Snort sebagai *tool* untuk mendeteksi dan memblokir serangan dari penyerang.
3) LOIC sebagai tool untuk melakukan serangan UDP *Flooding* ke server IDPS.
4) Hping3 sebagai tool untuk melakukan serangan SYN *Flooding* ke server IDPS.
5) Apache2 sebagai sistem untuk membangun web server.
6) *Afpacket* sebagai sistem yang melakukan pencegahan dengan memblokir serangan yang masuk.
7) Wireshark untuk melakukan *capture* data yang melewati suatu jaringan.

3.1.3 TOPOLOGI JARINGAN

Pada tahapan ini, komponen yang dibutuhkan pada tahapan sebelumnya akan dikonfigurasikan dengan membentuk topologi fisik berdasarkan sistem yang akan dijalankan. Pada penelitian ini, penulis mengimplementasikan sistem dengan jaringan *Local Area Network* (LAN).

Pada topologi jaringan yang digunakan, penulis membangun 2 *client* yaitu *client* 1 yang akan melakukan penyerangan ke web server, kemudian *client* 2 berperan sebagai web *client* yang dapat melakukan akses ke Web Server. Dimana, pada PC web server dibangun menggunakan apache2 untuk memberikan layanan HTTP kepada *client* serta dikonfigurasi dengan *rule* snort untuk menjalankan fungsi dari IDS dan IPS. Sehingga, snort akan melakukan pencegahan dan melakukan pemblokiran terhadap serangan yang ditujukan ke PC web server. Adapun perancangan topologi jaringan dalam penelitian ini yaitu:

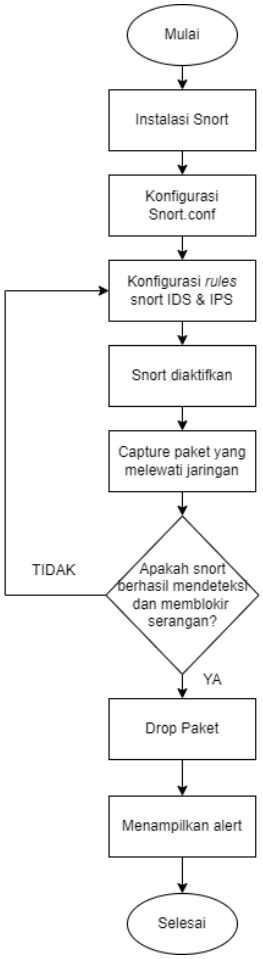


Gambar 3.2 Topologi Jaringan

Berdasarkan Gambar 3.2 diatas, dapat dilihat topologi jaringan yang digunakan dalam implementasi sistem IDPS ini. Jaringan ini dilengkapi dengan *Switch* yang mengatur lalu lintas data *internal* dengan alamat IP kelas A yaitu 10.10.10.1/8, serta *firewall* yang dimana terdapat sensor *inline* dan IDPS yang nantinya akan diimplementasikan untuk meningkatkan kemampuan snort yang dijalankan pada web server, kemudian router dan modem yang menghubungkan *internal network* dengan internet.

3.1.4 SKENARIO JARINGAN

Pada skenario jaringan penelitian ini, penulis melakukan konfigurasi rule pada snort serta *tool* pendukung lain yang akan digunakan. Dimana, *rule* akan menjalankan fungsi *Intrusion Detection System* (IDS) yang berfungsi untuk mendeteksi serangan masuk pada jaringan sesuai dengan *rule* yang telah diatur pada konfigurasi IDS. Selanjutnya, penulis akan melakukan konfigurasi *rule* pada snort yang akan menjalankan fungsi *Intrusion Prevention System* (IPS) untuk memblokir serangan yang ingin masuk ke dalam jaringan dengan mengaktifkan *afpacket* sebagai keamanan jaringan. Sehingga, rule yang telah dikonfigurasi pada snort dapat mendeteksi serangan apa saja yang akan diberi *alert* atau yang akan ditolak oleh sistem. Adapun alur skenario jaringan IDPS dapat dilihat pada diagram Gambar 3.3.



Gambar 3.3 Alur Skenario IDPS

Dapat dilihat pada Gambar 3.3, Dilakukan penginstallan paket-paket yang dibutuhkan dalam mendukung kinerja Snort agar penginstallan snort nantinya tidak terdapat kesalahan. Selanjutnya, melakukan konfigurasi *rules* pada snort.conf dan *rules* lainnya untuk konfigurasi sistem IDPS. Ketika penyerangan dilakukan dari PC penyerang ke PC server¸ maka *packet capture* pada Snort akan meng-*capture* sebuah paket penyerangan pada jaringan. Setelah Snort berhasil meng-*capture* paket dengan baik maka serangan yang masuk akan terdeteksi oleh *packet engine*. Selanjutnya apabila paket telah terdeteksi, IDS akan menjatuhkan paket dan *afpacket* akan memblokir paket yang dikirim melalui penyerang. Kemudian, IPS akan mengirimkan peringatan yang disimpan di dalam log, lalu peringatan tersebut akan muncul pada PC web server.

3.1.5 PERANCANGAN SISTEM

Pada tahap perancangan sistem IDPS ini, dilakukan konfigurasi snort dan *attacker* (penyerang) Tahapan instalasi Snort pada Ubuntu adalah sebagai berikut :

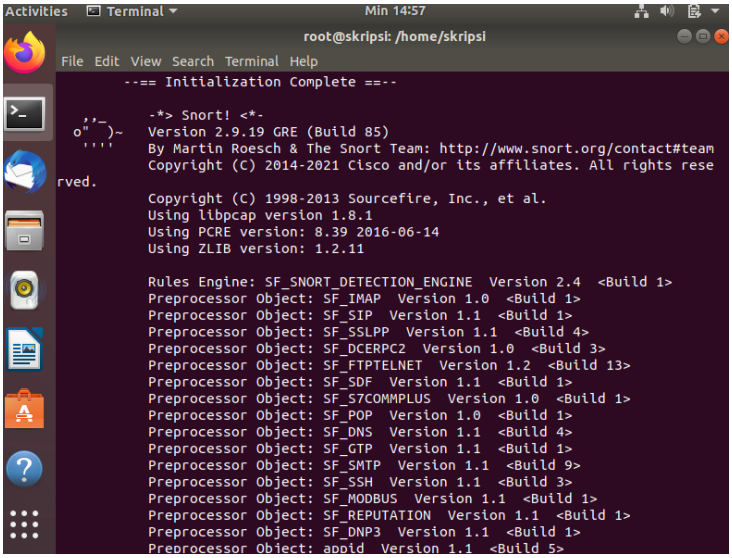
3.1.5.1 Konfigurasi Snort

1) Melakukan penginstallan *tool* Snort pada Ubuntu 18.04

Pada *tool* Snort, terdapat paket-paket yang dibutuhkan untuk menjalankan fungsi *Intrusion Detection Prevention System* (IDPS). Paket-paket tersebut dapat di *install* melalui terminal ubuntu 18.04. *Script* yang dijalankan untuk menginstall paket-paket tersebut adalah :

```
sudo apt install -y gcc libpcre3-dev zlib1g-dev
libluajit- 5.1-dev \ libpcap-dev openssl libssl-dev
libnghttp2-dev libdumbnet-dev \ bison flex libdnet
autoconf libtool
```

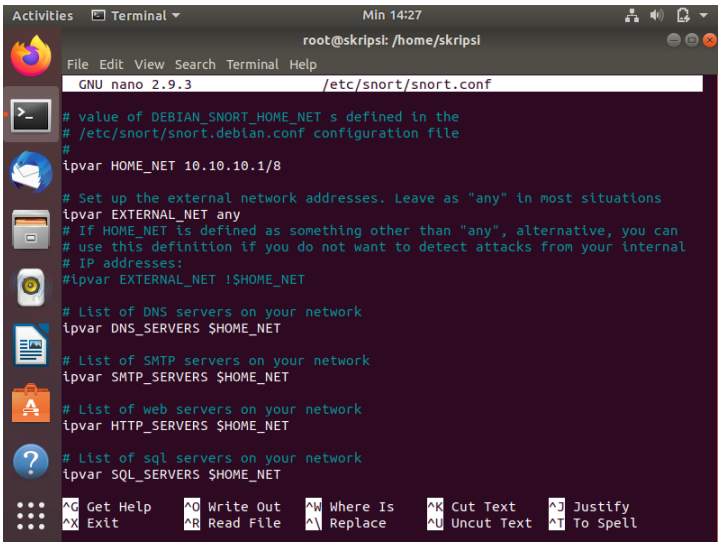
instalasi snort dapat dilakukan dapat dengan perintah apt-get install atau mengunduh program pada situs (www.snort.org). Setelah snort berhasil diinstal, maka lakukan pengecekan konfigurasi yang telah dilakukan seperti yang ditunjukkan pada gambar 3.4



Gambar 3.4 Penginstalan snort *complete*

2) Melakukan konfigurasi Snort.conf

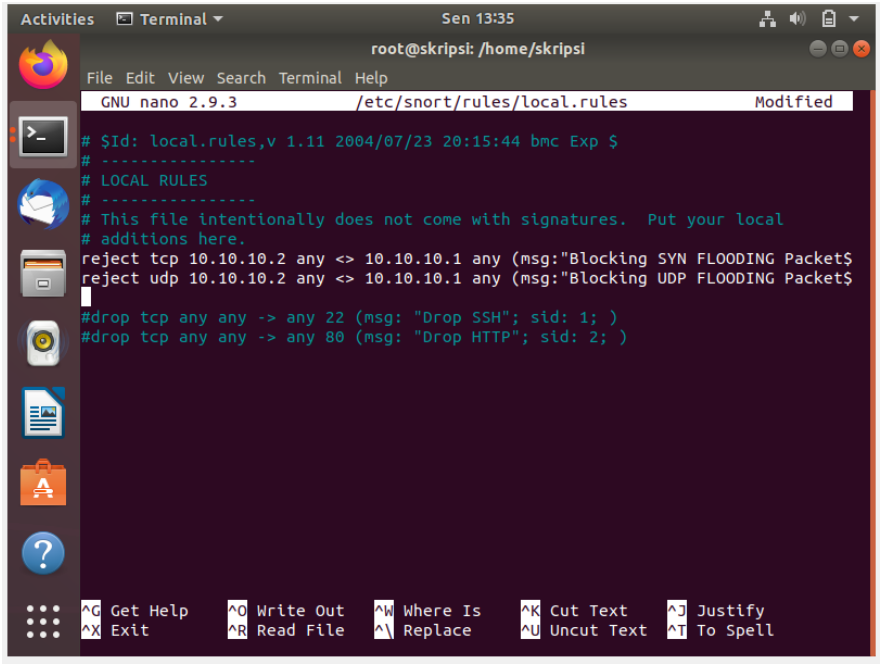
File konfigurasi Snort terletak di file snort.conf dalam direktori /etc/snort/snort.conf. Pengaturan utama yang dilakukan adalah menyimpan *rules* dalam direktori dan kemudian mengkonfigurasi jaringan. Pada snort.conf, dilakukan konfigurasi alamat jaringan dengan memberikan alamat IP host yaitu 10.10.10.1/8 dengan nama address group yaitu HOME_NET. Dalam file Snort.conf, dapat membuat *rules* atau perintah yang berfungsi untuk mengkonfigurasi penetapan alamat jaringan, menentukan antarmuka mana yang akan digunakan, dan menjalankan fungsi IDPS. Konfigurasi alamat IP dapat dilihat pada Gambar 3.5



Gambar 3.5 Konfigurasi alamat IP

3) Melakukan konfigurasi *file rules*

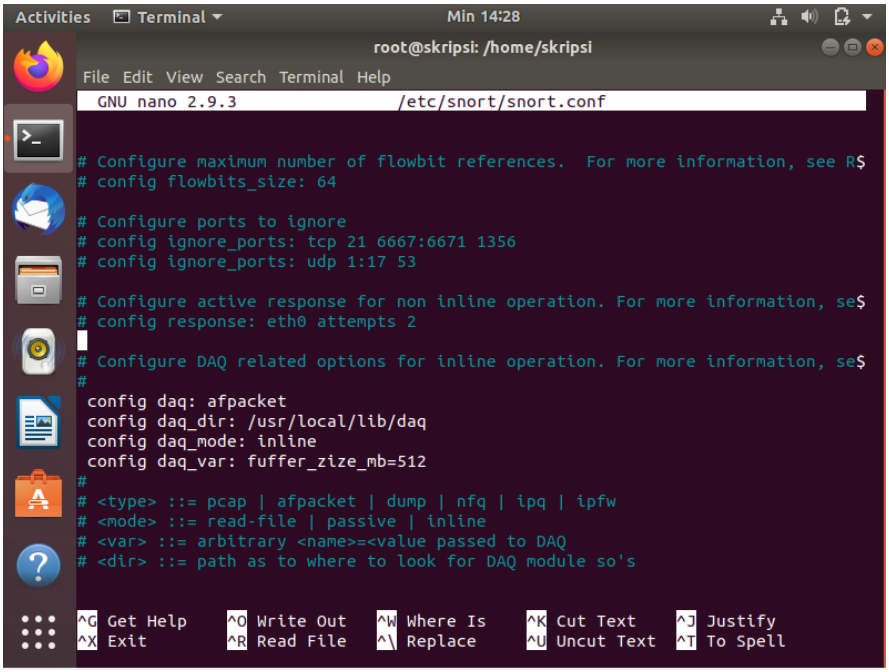
Pada tahap ini, untuk menyimpan rules yang telah dikonfigurasi maka kana disimpan dalam direktori yaitu RULE_PATH. Selanjutnya, dilakukan pengaktifan file local.rules, dimana file ini dapat menambah rules agar snort dapat memberi peringatan saat mendeteksi serangan. Dalam penelitian ini, penulis menggunakan rules untuk mendeteksi dan memblokir serangan UDP *flooding* dan serangan *SYN flooding* yang masuk kedalam jaringan yang ditunjukkan pada Gambar 3.6



Gambar 3.6 *Rules* yang digunakan

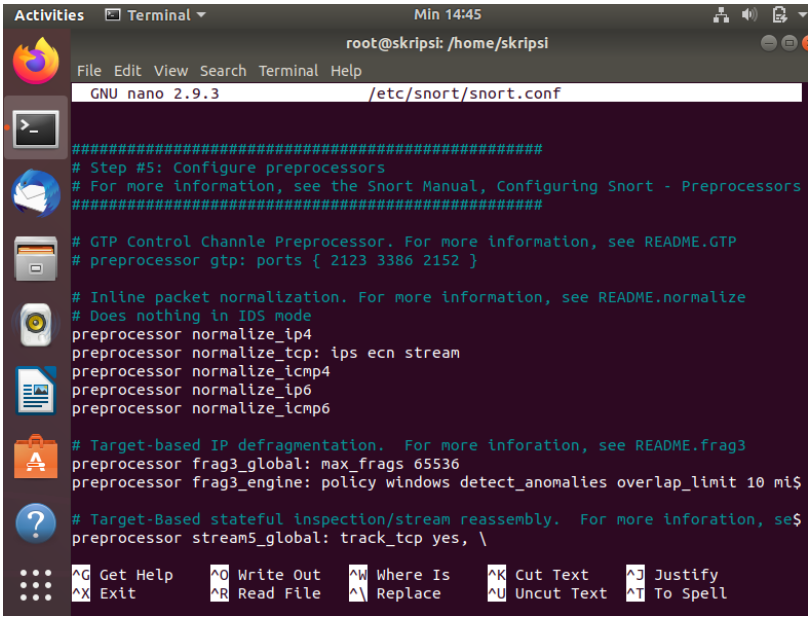
4) Konfigurasi *rules Intrusion Prevention System* (IPS)

Pada tahap ini akan dilakukan konfigurasi system IPS, karena kinerja snort yaitu menjalankan fungsi IDS atau dapat mendeteksi serangan atau aktifitas mencurigakan dalam jaringan. IDS sendiri dapat di*upgrade* ke sistem pencegahan intrusi "IPS". Ini berarti tidak hanya mendeteksi, tetapi juga menerapkan *rule* tertentu untuk mencegah serangan pada server. Selanjutnya, penulis melakukan konfigurasi IPS dengan mengaktifkan fungsi IPS yaitu menjalankan mode *inline* dengan *data aquisition* (daq). Dalam penelitian ini, penulis menggunakan skema penangkapan paket yaitu *afpacke*t. Dimana, *afpacket* menggunakan skema *forward* atau meneruskan paket dari *interface* satu ke *interface* lainnya sehingga dalam konfigurasi IDPS ini menggunakan dua *interface*. Konfigurasi *afpacket* pada Snort dapat dilihat pada Gambar 3.7.



Gambar 3.7 Konfigurasi *afpacket*

Kemudian, lakukan konfigurasi untuk mengaktifkan mode IPS seperti Gambar 3.8 dibawah ini.



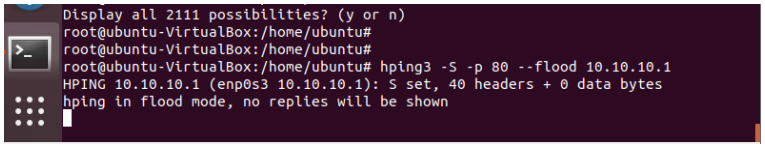
Gambar 3.8 Mengaktifkan mode IPS

3.1.5.2 Konfigurasi Penyerang (*Attacker*)

Pada penelitian ini, penulis melakukan pengujian serangan untuk mengetahui kinerja snort dalam menjalankan fungsi IDPS. Uji coba serangan

yang penulis gunakan adalah SYN *flooding* dan UDP *flooding.* Serangan SYN *flooding* dengan menggunakan *tool* Hping3 dan serangan UDP *flooding* menggunakan *tool* LOIC. Untuk konfigurasi penyerang, menggunakan alamat IP 10.10.10.2/8

Konfigurasi pada hping3 dilakukan pada OS ubuntu 18.04, kemudian memasukkan alamat IP target 10.10.10.1 dan memilih aktifitas *scan* yang diinginkan. Uji coba serangan SYN *flooding* menggunakan Hping3 dapat dilihat pada Gambar 3.11



Gambar 3.9 SYN *flooding* menggunakan Hping3

Berdasarkan Gambar 3.10, uji coba serangan dikirim menuju port 80 yang merupakan HTTP yaitu web server yang dibangun oleh penulis.

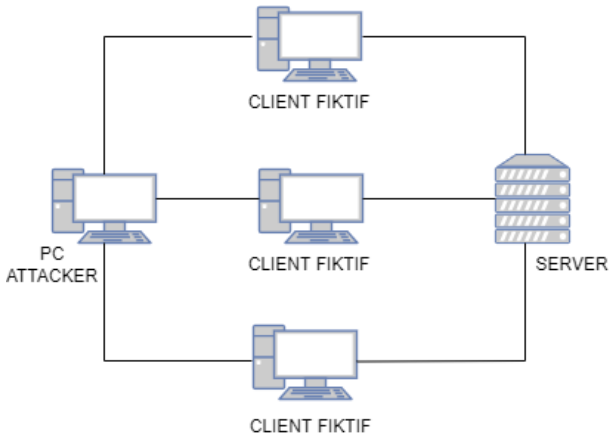
Selanjutnya, untuk menjalankan serangan UDP Flooding menggunakan *tool* LOIC. Lalu memasukkan alamat IP target dan memilih protokol UDP untuk metode serangan flooding yang akan digunakan dengan port tujuan yaitu 80. Tampilan LOIC ketika melakukan uji coba serangan dapat dilihat pada Gambar 3.11.

Gambar 3.10 UDP flooding menggunakan LOIC

Setelah penyerang dikonfigurasi dan dapat melakukan serangan, sistem Snort dapat menentukan apakah *rule* konfigurasi yang dijalankan dapat mendeteksi dan memblokir serangan SYN dan UDP *flooding* yang masuk.

3.1.6 ALUR SERANGAN

Pada serangan ini, paket dikirimkan ke server secara berlebihan sehingga menyebabkan server kebanjiran paket dan server gagal berfungsi dengan baik. Adapun alur serangan yang terjadi dapat dilihat pada Gambar 3.11 berikut.



Gambar 3.11 Alur serangan

Dalam melakukan penyerangan SYN flooding, *attacker* melakukan sinkronisasi terlebih dahulu yaitu koneksi TCP ke server, agar server dan pc *attacker* dapat saling terkoneksi. Selanjutnya, *attacker* mengirimkan kode SYN untuk meminta koneksi ke server dan server akan mengenali *request* attacker, kemudian mengirimkan SYN-ACK ke PC *attacker*. Akan tetapi, attacker merespon kembali dengan mengirimkan SYN ke semua *port* dalam server secara terus-menerus. Attacker membuat SYN yang kelihatan *valid* dengan IP *address* yang palsu sehingga server tidak mengakhiri koneksi tersebut. Akibatnya, koneksi yang terjalin tetap berjalan sehingga membuat server menjadi sibuk untuk merespon *request attacker* secara terus-menerus. Hal ini tentunya membuat *client* sulit terkoneksi dengan server.

Dalam penyerangan UDP flooding, attacker akan membanjiri server dengan mengirimkan UDP palsu ke host dengan menggunakan port secara acak. Tentunya hal ini mengakibatkan host mengecek secara terus-menerus aplikasi yang berjalan pada port tersebut. Sehingga server akan melakukan *reply* pesan "D*estination Unreachable*" dan menyebabkan server kehabisan *resource*.

3.1.7 ANALISIS HASIL

Dalam penelitian ini, penulis menggunakan parameter *Quality of Service* (QoS) dalam menganalisa hasil performa pada Snort dengan menggunakan perangkat lunak *network analyzer* yaitu wireshark pada setiap skenario pengujian. Adapun parameter QoS yang akan diukur yaitu *throughput*, *delay*, *Jitter*, dan *packet loss*.

1. *Throughput*

   Pengukuran *throughput* berdasarkan standarisasi *Telecommunications and Internet Protocol Over Networks* (TIPHON). Dimana dalam pengukuran nya terdapat beberapa kategori untuk menentukan kualitas jaringan tersebut.

Tabel 3.3 Standar Kualitas *Throughput*

Kategori *Throughput*	Nilai *Throughput* (bps)	Indeks
Sangat Baik	100	4
Baik	75	3
Cukup Baik	50	2
Buruk	<25	1

(Sumber : TIPHON 1999-2006) [33]

2. *Delay*

   Pengukuran *delay* berdasarkan standarisasi *Telecommunications and Internet Protocol Over Networks* (TIPHON). Dimana dalam pengukuran nya terdapat beberapa kategori untuk menentukan kualitas jaringan tersebut.

Tabel 3.4 Standar Kualitas *Delay*

Kategori *Latency*	Nilai *Delay* (ms)	Indeks
Sangat Baik	<150 ms	4
Baik	150 ms s/d 300 ms	3

Cukup Baik	300 s/d 450 ms	2
Buruk	>450 ms	1

(Sumber : TIPHON 1999-2006) [33]

3. *Jitter*

Pengukuran *Jitter* berdasarkan standarisasi *Telecommunications and Internet Protocol Over Networks* (TIPHON). Dimana dalam pengukuran nya terdapat beberapa kategori untuk menentukan kualitas jaringan tersebut.

Tabel 3.5 Standar Kualitas *Jitter*

Kategori *Jitter*	*Peak Jitter* (ms)	Indeks
Sangat Baik	0 ms	4
Baik	1 s/d 75 ms	3
Cukup Baik	76 s/d 125 ms	2
Buruk	>225 ms	1

(Sumber : TIPHON 1999-2006) [33]

4. *Packet Loss*

Pengukuran *Jitter* berdasarkan standarisasi *Telecommunications and Internet Protocol Over Networks* (TIPHON). Dimana dalam pengukuran nya terdapat beberapa kategori untuk menentukan kualitas jaringan tersebut.

Tabel 3.6 Standar Kualitas *Packet Loss*

Kategori *Degredasi*	Nilai *Packet Loss* (%)	Indeks
Sangat Baik	0 – 2%	4
Baik	3 - 14%	3
Cukup Baik	15 - 24%	2
Buruk	>25%	1

(Sumber : TIPHON 1999-2006) [33]