BAB 3
METODE PENELITIAN

Metodologi Penelitian berisi uraian diagram alur penelitian. Diagram alur penelitian menjelaskan mengenai tahap-tahap penelitian. Dalam penelitian ini diperlukan juga alat pendukung untuk menunjang penelitian. Penelitian ini juga membutuhkan topologi yang berfungsi sebagai objek pengambilan data pada proses penelitian.

3.1 ALAT YANG DIGUNAKAN

3.1.1 PERANGKAT KERAS

Hardware yang dibutuhkan dalam penelitian ini untuk menjalankan kebutuhan *software* dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kebutuhan *Hardware*

No	Nama	Spesifikasi	Keterangan
1	PC	Processor (CPU)	11th Gen Intel(R) Core (TM) i7-11700F @ 2.50GHz (16 CPUs), ~2.5GHz
		Memory	16 GB RAM
		Penyimpanan	HDD 1 TB

3.1.2 PERANGKAT LUNAK

Software yang dibutuhkan dalam penelitian ini untuk konfigurasi pengujian dapat dilihat pada Tabel 3.2.

Tabel 3.2 Kebutuhan *Software*

No	Software	Keterangan
1	Snort	Software IDS
2	pfSense	Software Firewall
3	VMware Workstation	Software Virtual Machine
4	EVE-NG	Software Virtualisasi jaringan
5	Hping3	Software Serangan

No	Software	Keterangan
6	Apache2	Software Web Server
7	Wireshark	Software Packet Capture
8	Orange	Software Data Mining

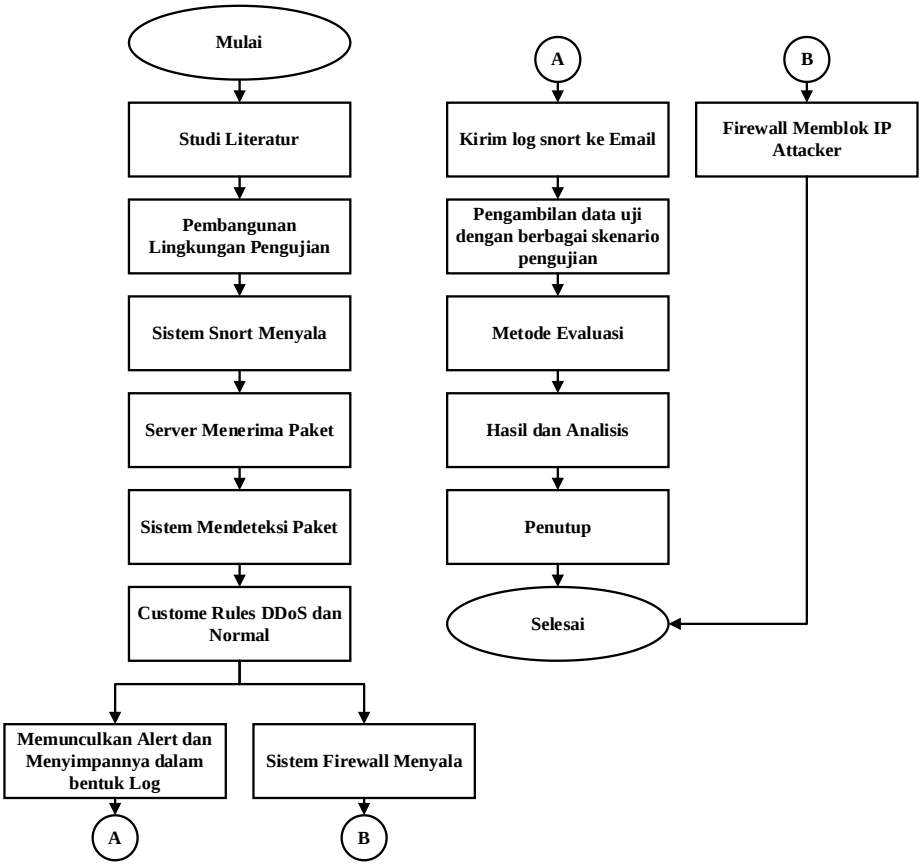
Konfigurasi *software* dalam penelitian ini dapat dilihat pada Tabel 3.3.

Tabel 3.3 Konfigurasi *Software*

No	Software	Letak Konfigurasi
1	Snort	dikonfigurasi pada pfSense karena termasuk IDS packages dari pfSense
2	pfSense	dikonfigurasi pada Interface Web melalui IP pfSense
3	VMware Workstation	dikonfigurasikan pada PC Admin dengan OS Windows
4	EVE-NG	dikonfigurasi pada Vmware dengan OS Ubuntu
5	Hping3	dikonfigurasikan pada PC Attacker dengan OS Ubuntu
6	Apache2	dikonfigurasikan pada Server dengan OS Ubuntu
7	Wireshark	dikonfigurasikan pada PC Admin dengan OS Windows
8	Orange	dikonfigurasikan pada PC Admin dengan OS Windows

3.2 ALUR PENELITIAN

Pada alur penelitian berisi tahapan dalam menyelesaikan penelitian yang berjudul Deteksi dan Pencegahan Serangan DDoS Pada *Server* Menggunakan Snort dengan Notifikasi *Email*. Alur penelitian juga berfungsi sebagai panduan alur pengerjaan dalam penelitian yang ditunjukan agar penelitian berjalan dengan sesuai harapan. Tahapan tersebut berupa langkah-langkah yang akan dilakukan dalam penelitian secara sistematis dan spesifik yang pada akhirnya berguna sebagai sistem pengumpulan data dari setiap serangan. Berikut merupakan gambaran metodologi berupa *flowchart* yang ditunjukan pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

Gambar 3.1 menunjukan diagram alur perancangan sistem dalam penelitian ini. Langkah pertama yaitu Studi literatur, digunakan sebagai pedoman pengetahuan dasar dalam melakukan analisis, perancangan, implementasi dan pengujian dalam tahap-tahap penelitian. Dasar teori yang dibutuhkan sebagai pendukung penelitian ini adalah Jaringan Komputer, IDS, Snort, DDoS, *Firewall* dan *Email Notification*.

Langkah kedua yaitu Pembangunan lingkungan pengujian, yaitu segala komponen-komponen, topologi sistem dan perancangan sistem coba dirancang pada tahap ini. Komponen yang digunakan meliputi *Virtual Machine*, Snort, *Firewall*, *Email*, *Orange*. Selanjutnya dari alur sistem itu sendiri yaitu Snort akan dijalankan yang nantinya *Server* akan menerima paket dan Snort akan mendeteksi paket menggunakan *custome rules*nya, termasuk kategori skenario DDoS dan Normal. Kemudian dari Snort akan memunculkan *alert* kedua skenario tersebut dan disimpan dalam bentuk *log*, bersandingan dengan *Firewall* yaitu *pfSense* yang ikut menyala untuk melakukan pemblokiran IP terhadap skenario DDoS. Setelah Snort

menghasilkan *alert* kedua skenario dan disimpan dalam bentuk *log*, selanjutnya menerapkan sistem notifikasi dengan mengirimkan pesan *log* Snort kepada *Admin* melalui fitur *email* yang disediakan oleh *pfSense*.

Langkah ketiga yaitu Pengambilan data uji dengan berbagai skenario pengujian, di mana pengujian akan dilakukan dengan 2 skenario yang akan dijalankan pada *environment* yang telah dirancang. Skenario tersebut antara lain, yaitu melancarkan skenario DDoS yaitu TCP *Flood* dan UDP *Flood* serta melancarkan skenario Normal yaitu TCP Normal dan UDP Normal.

Setelah melancarkan kedua skenario tersebut, pengambilan data uji dilakukan dengan pengambilan data *log* Snort berisi paket skenario DDoS dan Normal dari pesan *email* secara *periodic* yang nantinya akan dilakukan klasifikasi *log* Snort sebagai metode evaluasi kinerja Snort parameter akurasi dan efektivitas deteksi. Klasifikasi *log* Snort menggunakan *software Orange*, di mana langkah pertama adalah menyiapkan dataset *log* Snort melalui pengelompokkan atribut, atribut *feature* sebagai pendukung target dan atribut *label* sebagai kelas target itu sendiri, artinya *alert* pada *log* Snort termasuk dalam kategori DDoS atau Normal. Lalu dari data pengelompokkan atribut diklasifikasikan menggunakan algoritma *decision tree* beserta visualisasinya melalui *tree viewer*, kemudian mengevaluasi hasil klasifikasi dan memvisualisasikanya dalam tabel *confusion matrix*.

Selain klasifikasi, dilakukan juga pengukuran *detection latency* sebagai metode evaluasi kinerja Snort parameter kecepatan deteksi. Pengambilan data *detection latency* diperoleh dari waktu paket skenario DDoS dan Normal terdeteksi Snort dikurangi waktu mulai paket skenario DDoS dan Normal masuk atau ter*capture* oleh *pfSense*. Lalu dilakukan juga pengukuran CPU *Usage* sebagai metode evaluasi kinerja Snort parameter penggunaan sumber daya. Pengambilan data CPU *Usage* diperoleh dari kondisi awal skenario DDoS dikirimkan dan kondisi akhir skenario Normal dikirimkan.

Langkah keempat yaitu Hasil dan Analisis, yaitu membahas nilai kinerja Snort yang diperoleh. Nilai kinerja Snort yang pertama adalah dari nilai *confusion matrix* dan pengukuran metrik sebarannya. Nilai kinerja Snort yang kedua adalah dari nilai standar deviasi. Jadi nilai *confusion matrix* merujuk pada hasil klasifikasi *log* Snort untuk parameter akurasi dan efektivitas deteksi, sedangkan nilai standar

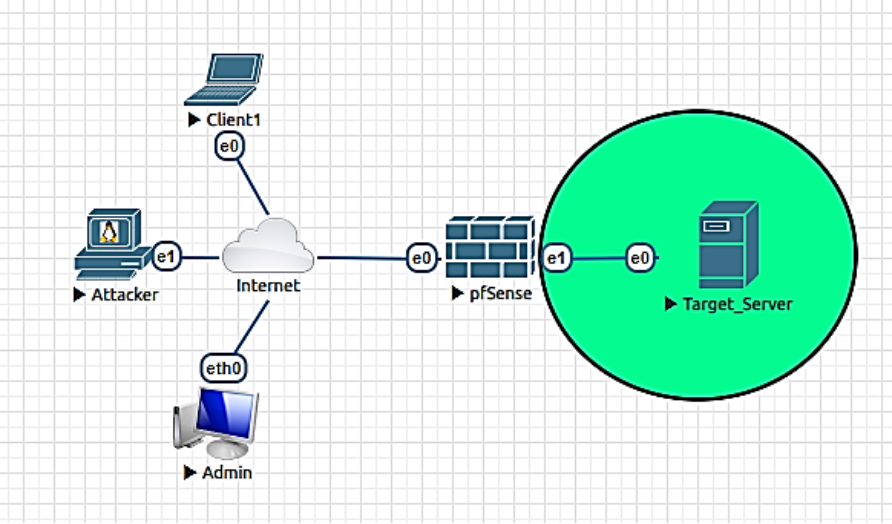
deviasi merujuk pada pengukuran parameter kecepatan deteksi dan penggunaan sumber daya melalui metode *detection latency* dan CPU *Usage*.

Terakhir yaitu langkah kelima atau Penutup, berisi kesimpulan yang memberikan jawaban atas pertanyaan yang diajukan sebelumnya. Keseluruhan jawaban hanya terfokus pada ruang lingkup pertanyaan dan jumlah jawaban disesuaikan dengan jumlah rumusan masalah yang diajukan. Bagian ini juga berisi saran untuk penelitian kedepannya.

3.3 PERANCANGAN TOPOLOGI

3.3.1 TOPOLOGI DETEKSI DAN PENCEGAHAN

Penelitian ini menggunakan topologi jaringan untuk melakukan pengujian deteksi dan pencegahan serangan DDoS pada *server* menggunakan Snort dengan notifikasi *Email*. Topologi jaringan dapat dilihat pada Gambar 3.2, menunjukkan topologi Router-to-LAN, di mana dalam topologi ini *pfSense* bertindak sebagai *router* sekaligus *firewall* yang menghubungkan LAN dengan WAN atau internet. *pfSense* memiliki dua antarmuka jaringan, yaitu antarmuka WAN yang terhubung ke luar jaringan dan antarmuka LAN yang terhubung ke dalam atau lokal jaringan. Jadi topologi ini terdiri dari jaringan WAN yang diwakili oleh internet yang menghubungkan tiga perangkat, yaitu PC *Client1*, PC *Attacker*, dan *Admin*, dengan menggunakan *pfSense* sebagai perangkat *router* dan *firewall*. Target *Server* juga terhubung langsung ke *pfSense* dan berada dalam jaringan LAN.



Gambar 3.2 Topologi Jaringan Router-to-LAN

Kelima perangkat ini digunakan untuk melakukan pengujian deteksi dan pencegahan pada *server* menggunakan Snort dengan notifikasi *email*, yang nantinya dapat memperoleh hasil kinerja deteksi dari klasifikasi *log* Snort melalui skenario dan parameter. PC *Client1* berfungsi untuk mengirimkan paket normal dan PC *Attacker* berfungsi untuk mengirimkan paket serangan (DDoS). Admin berfungsi sebagai penerima notifikasi *alert* dari *pfSense* melalui *Email*. Kemudian *pfSense* berfungsi sebagai *firewall* sekaligus di mana Snort diinstall, karena termasuk fitur dan *additional packages pfSense*.

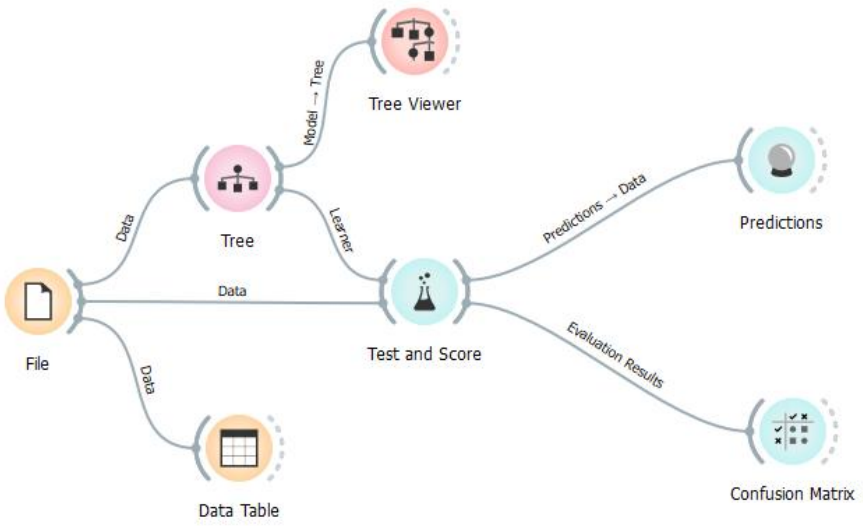
Tabel 3.4 Alamat IP Perangkat Jaringan

No	*Device*	*Interface*	IP *Address*
1	PC Client1	eth0	10.211.92.54/23
2	PC Attacker	eth1	10.211.92.55/23
3	Admin	eth0	10.212.39.37
4	pfSense	vtnet0	10.211.92.53/22
		vtnet1	192.168.200.1/24
5	Target Server	eth0	192.168.200.10/24
6	EVE-NG	eth0	10.211.92.52

Tabel 3.4 berisi mengenai identitas berupa alamat ip dari masing-masing perangkat PC *Client1*, PC *Attacker*, *Admin*, *pfSense*, dan Target *Server* serta ip dari sistem virtualisasi jaringan atau EVE-NG.

3.3.2 TOPOLOGI KLASIFIKASI

Penelitian ini juga menggunakan topologi klasifikasi untuk melakukan klasifisikasi *log* Snort. Klasifikasinya akan berfokus dalam membedakan lalu lintas DDoS dan Normal dari atribut yang digunakannya. Topologi dapat dilihat pada gambar 3.3, merupakan topologi pada *Software Orange*.



Gambar 3.3 Topologi Klasifikasi *Log* Snort pada *Orange*

Pada topologi klasifikasi ini, tersusun beberapa *widget orange*, di mana yang pertama ada *File*, *Data Table*, *Tree*, *Tree Viewer*, *Test and Score*, *Predictions*, dan *Confusion Matrix*. Jadi dari susunan *widget* tersebut menunjukkan langkah dalam pengklasifikasian *log* Snort menggunakan *Software Orange*. Seperti pada Tabel 3.5 menjelaskan setiap susunan *widget orange* sebelumnya.

Tabel 3.5 Deskripsi *Widget* pada *Orange*

No	Widget	Deskripsi
1	File	Membaca data nilai atribut dari file input.
2	Data Table	Menampilkan data nilai atribut dalam spreadsheet
3	Tree	Algoritma pohon keputusan
4	Tree Viewer	Visualisasi pohon klasifikasi
5	Test and Score	Menguji algoritma pembelajaran pada data
6	Predictions	Menampilkan prediksi model pada data
7	Confusion Matrix	Menunjukkan proporsi antara kelas yang diprediksi dan yang sebenarnya

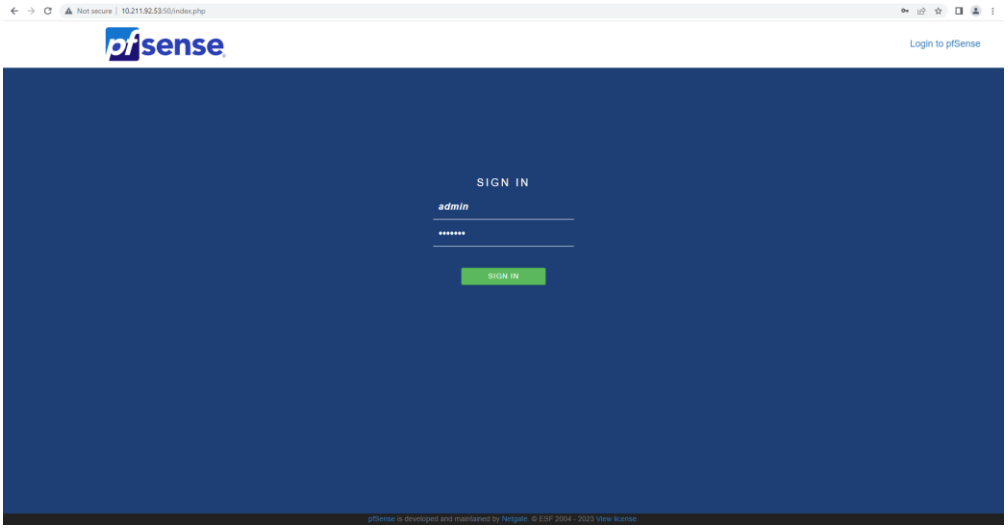
3.4 KONFIGURASI SOFTWARE

3.4.1 KONFIGURASI PFSENSE

PfSense adalah sebuah perangkat lunak *open-source* yang berfungsi sebagai *firewall* dan *router*. Konfigurasi *pfSense* memungkinkan pengguna untuk mengatur dan mengelola fitur-fitur yang ada dalam perangkat lunak tersebut, sehingga dapat memenuhi kebutuhan jaringan yang spesifik. Contonya konfigurasi *firewall* pada *pfSense*, di mana berfungsi untuk mengatur aturan-aturan yang mengontrol lalu lintas jaringan yang masuk dan keluar dari jaringan, seperti mengakses antarmuka *web*, konfigurasi antarmuka WAN dan LAN, membuat dan mengonfigurasi *firewall rules*, mengatur *port forwarding* dan *Virtual* IPs serta mengatur fitur tambahan yang lainnya.

A. Konfigurasi akses antarmuka *web*

Konfigurasinya yaitu dengan membuka *browser web* dan mengakses alamat IP *pfSense* yaitu 10.211.92.53/50. Masukkan *username* dan *password* secara *default* yaitu "*admin*" dan "*pfsense*". Seperti pada gambar 3.4 merupakan *interface log in pfSense*.

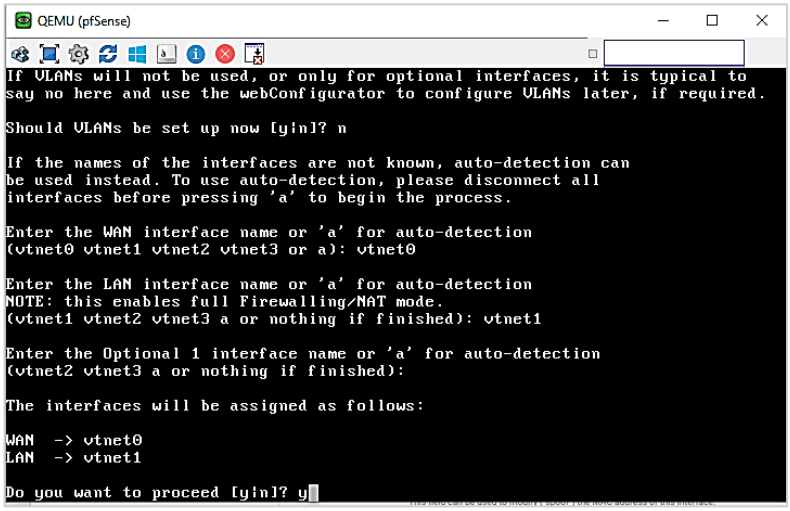


Gambar 3.4 Akses *Login pfSense*

B. Konfigurasi Antarmuka WAN dan LAN

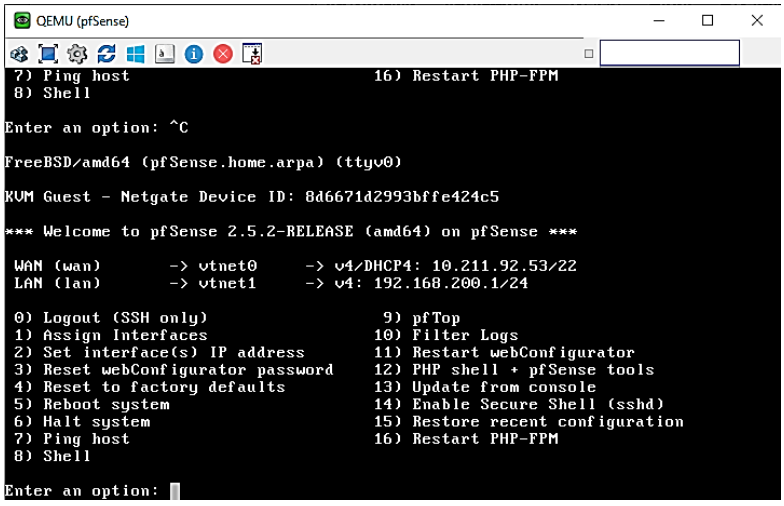
Konfigurasi antarmuka WAN pada *pfSense* berfungsi untuk menghubungkan jaringan lokal dengan *internet*, memungkinkan perangkat dalam jaringan untuk terhubung ke layanan *online* dan berkomunikasi dengan sumber daya di *internet*. Kemudian melalui antarmuka WAN, *pfSense* menerima alamat IP publik dari ISP untuk mengidentifikasi jaringan lokal di

internet. Pada gambar 3.5 merupakan konfigurasi antarmuka WAN dengan melakukan *assign interface* pada menu *FreeBSD pfSense* dan memasukkan *vtnet0* untuk WAN (DHCP) serta *vtnet1* untuk LAN.



Gambar 3.5 *Assign Interface* WAN dan LAN

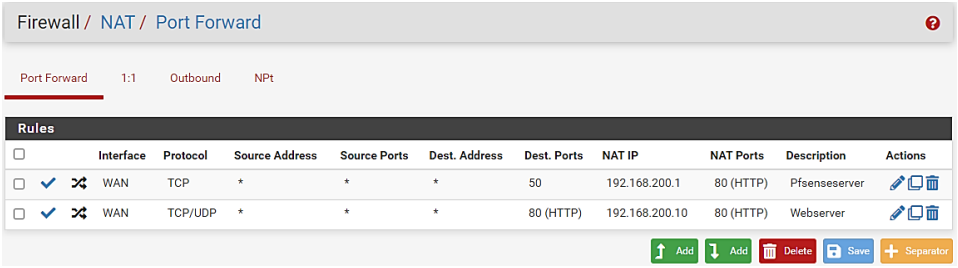
Kemudian dengan melakukan *assign interface* WAN dan LAN, maka muncul IP dari masing-masing *interface* seperti pada gambar 3.5, di mana *vtnet0* sebagai WAN dengan IP 10.212.92.53/22 dan *vtnet1* sebagai LAN dengan IP 192.168.200.1/24. Seperti gambar 3.6 merupakan tampilan menu *pfSense* pada sistem *FreeBSD*.



Gambar 3.6 *Interface* WAN dan LAN *pfSense*

C. Konfigurasi NAT *Port Forwarding*

Konfigurasi NAT *port forwarding* pada *pfSense* memungkinkan akses dari *internet* ke layanan atau aplikasi yang dihosting di jaringan internal. Dengan menetapkan aturan *port forwarding*, lalu lintas yang masuk ke alamat IP publik akan diarahkan ke alamat IP dan *port* yang ditentukan dari *server* internal. Ini memungkinkan *hosting* layanan seperti *web server*, *server email*, dan lainnya. Pengguna dapat mengalihkan *port* eksternal ke *port* internal yang berbeda, meningkatkan keamanan dengan mengontrol lalu lintas yang diterima, dan memungkinkan akses publik ke layanan yang di*hosting* di jaringan internal. Seperti pada gambar 3.7 merupakan tampilan dari NAT *Port Forward* pada *Firewall pfSense.*



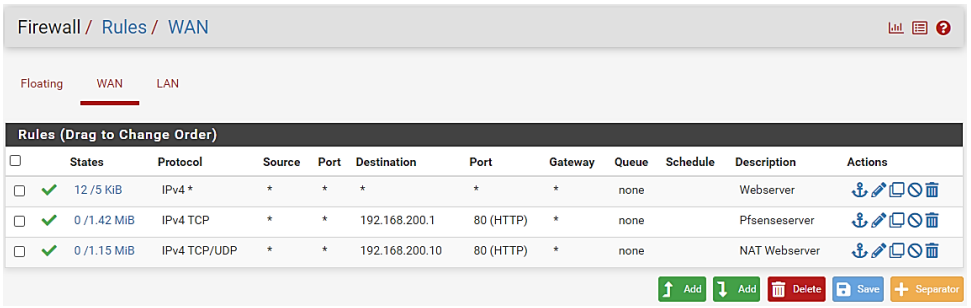
Gambar 3.7 NAT *Port Forward pfSense*

Dari gambar 3.7 menjelaskan bahwa terdapat 2 konfigurasi *port forwarding* pada *pfSense* yaitu *pfSense* dan *webserver*. Alasan dilakukan *port forwarding*, karena *pfSense* memiliki IP LAN, maka diterapkan *port forwarding* agar *pfSense* dapat diakses secara publik melalui alamat IP publik. Kemudian pada *web server* juga dilakukan *port forwarding*. Pada dasarnya *web server* berasal dari *server* yang memiliki IP LAN, maka diterapkan juga *port forwarding* agar *web server* dapat diakses secara publik melalui alamat IP public. Kesimpulannya adalah dari konfigurasi tersebut memungkinkan untuk mengakses *server web* menggunakan *port* yang berbeda secara publik tanpa mengganggu konfigurasi *server* internal.

D. Konfigurasi *Firewall Rules*

Konfigurasi *Firewall Rules* memungkinkan untuk menerapkan kebijakan keamanan jaringan yang ketat, di mana dapat mengatur aturan untuk memblokir atau mengizinkan lalu lintas berdasarkan alamat IP, *port*, protokol, dan kriteria

lainnya. Konfigurasi ini membantu melindungi jaringan internal dari serangan dan ancaman yang mungkin berasal dari luar.

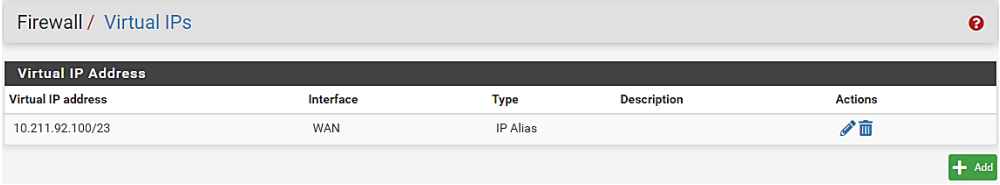


Gambar 3.8 *Firewall Rules* dari *pfSense*

Dari gambar 3.8 menjelaskan bahwa ketiga *firewall rules* diterapkan dengan *action* diizinkan (*Pass*) pada lalu lintas yang masuk melalui jaringan WAN menuju jaringan LAN dengan protocol TCP/UDP.

E. Konfigurasi *Virtual* IPs

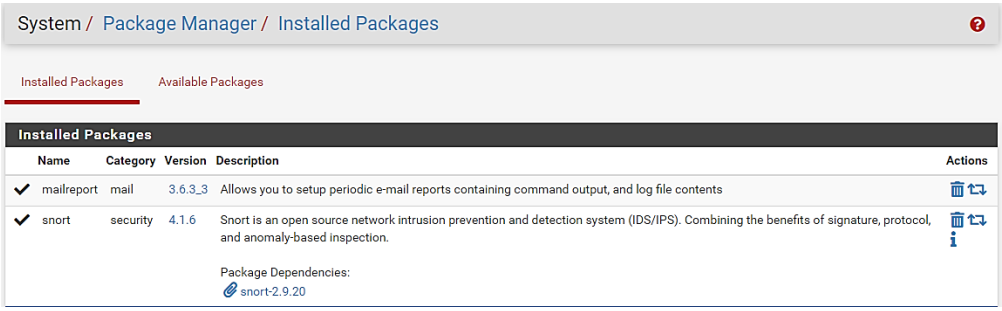
Konfigurasi *Virtual* IPs pada *pfSense* memungkinkan untuk meng-*host* beberapa layanan atau aplikasi di *server* yang sama menggunakan alamat IP yang berbeda. Dengan menetapkan beberapa alamat IP *virtual* pada antarmuka yang sama, dapat mengarahkan lalu lintas ke layanan atau aplikasi yang spesifik berdasarkan alamat IP tujuan. Selain itu, penggunaan *Virtual* IPs juga memungkinkan akses publik ke layanan internal dengan mengkonfigurasi aturan *port forwarding*. Dengan mengarahkan lalu lintas dari alamat IP publik ke alamat IP *virtual*, dapat memberikan akses eksternal ke layanan tersebut tanpa perlu mengungkapkan alamat IP sebenarnya dari *server* internal. Seperti pada gambar 3.9 merupakan konfigurasi *Virtual* IPs pada *server* dengan alamat IP yang berbeda yaitu 10.211.92.100/23.



Gambar 3.9 *Virtual* IPs *pfSense*

F. Konfigurasi *Package Manager*

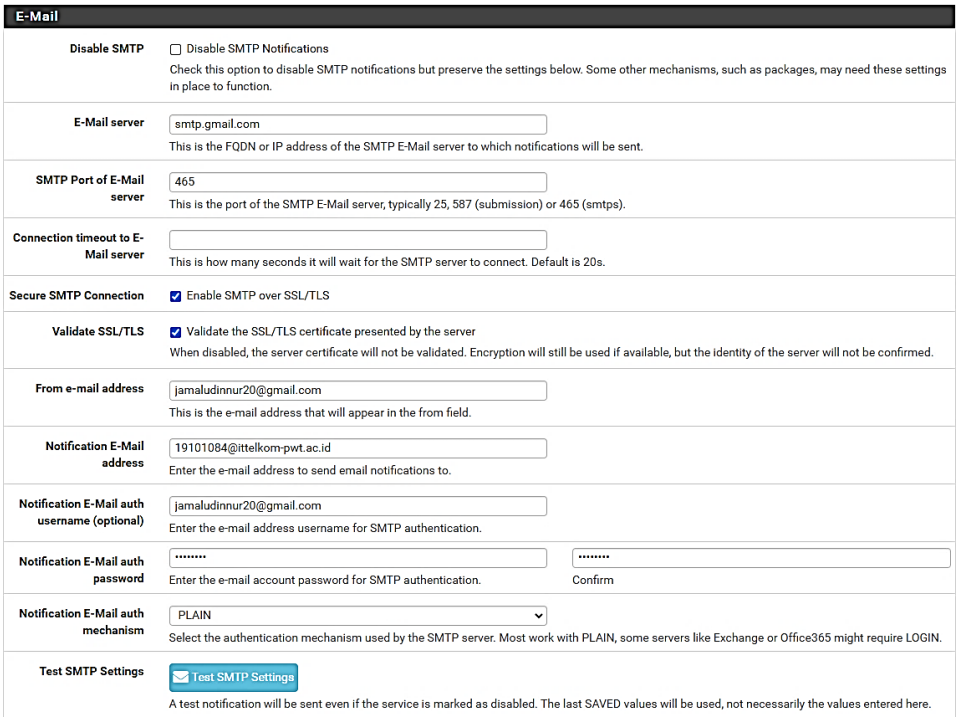
Konfigurasi *Package Manager* pada *pfSense* memiliki fungsi utama untuk mengelola dan menginstal paket tambahan atau ekstensi yang dapat meningkatkan fungsionalitas dan kemampuan *pfSense*. Seperti pada gambar 3.10 menunjukkan instalasi *Mailreport* sebagai sistem notifikasi secara *periodic* dan Snort sebagai sistem IDS.



Gambar 3.10 Installasi paket tambahan pada *pfSense*

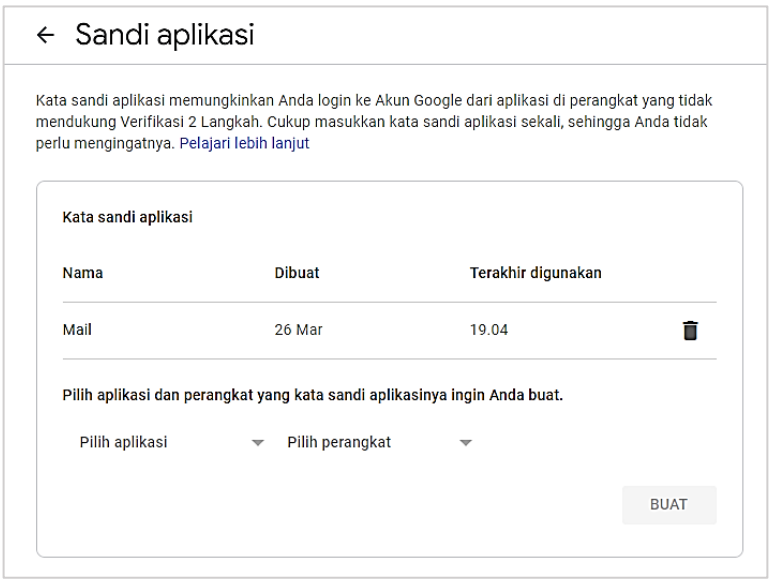
G. Konfigurasi Notifikasi *Email*

Konfigurasi notifikasi *email* pada *pfSense* memiliki fungsi utama untuk memberikan pemberitahuan atau notifikasi melalui *email* tentang kejadian penting atau masalah yang terjadi di jaringan seperti serangan DDoS.



Gambar 3.11 Konfigurasi Notifikasi *Email* pada *pfSense*

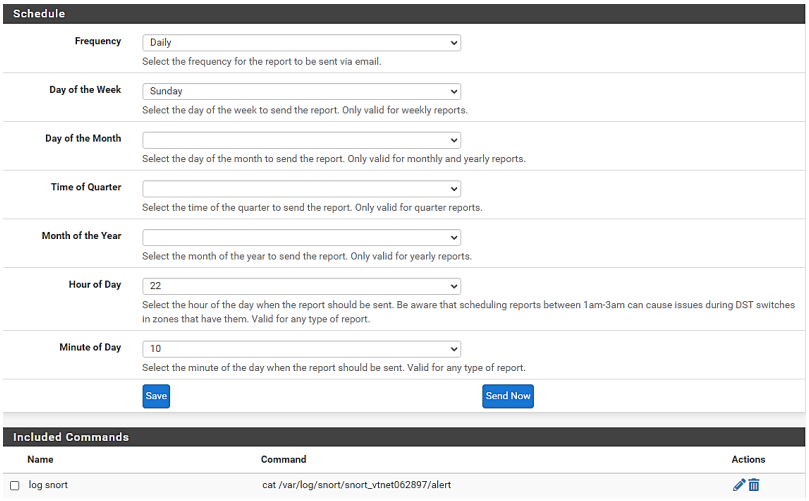
Dari gambar 3.11 menunjukkan terkait konfigurasi notifikasi *email* pada *pfSense*. Konfigurasinya meliputi *E-Mail server* yaitu *smtp.gmail.com*, SMTP *Port*nya adalah 465, *Enable Secure* SMTP *Connection*, *Enable validate* SSL/TLS, dengan *From e-mail address* jamaludinnur20@gmail.com sebagai pengirim notifikasi, dan *Notification E-Mail address* 19101084@ittelkom-pwt.ac.id sebagai penerima (*admin*) notifikasi. *Notification E-Mail auth password* yaitu sandi aplikasi dari akun *email* untuk autentikasi SMTP. Seperti pada gambar 3.12 yang menunjukkan konfigurasi sandi aplikasi *email*.



Gambar 3.12 Konfigurasi Sandi Aplikasi pada Akun *Email*

H. Konfigurasi *Mailreport*

Konfigurasi *Mailreport* pada *pfSense* berfungsi untuk mengatur pengiriman laporan melalui *email* secara otomatis. Fitur ini memungkinkan untuk mengonfigurasi pengiriman laporan sistem, laporan *firewall*, atau laporan Snort ke alamat email yang ditentukan. Dengan menggunakan *Mailreport*, dapat mengatur frekuensi pengiriman laporan, seperti laporan harian, mingguan, atau bulanan.

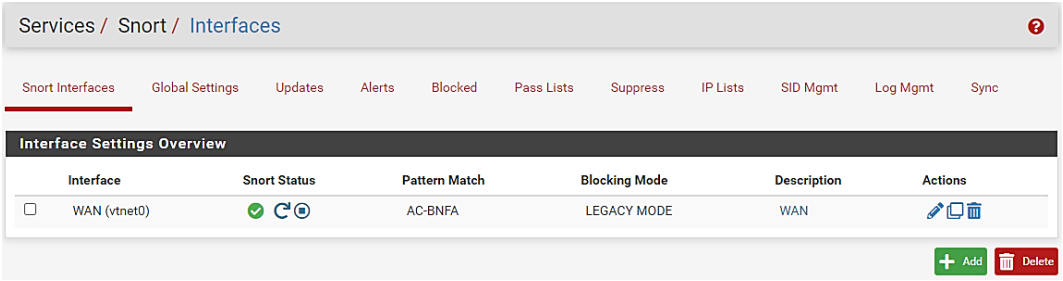


Gambar 3.13 Konfigurasi *Emailreport* dan *command*nya

Seperti pada gambar 3.13 menunjukkan konfigurasi *Mailreport* atau *Emailreport* pada *pfSense* dengan frekuensi pengiriman laporan yaitu harian pada pukul 22.10 dan *include commands* dengan perintah *cat /var/log/snort/snort_vtent062897/alert* untuk mengirimkan *content* secara spesifik kepada admin yaitu pesan *log* Snort.

3.4.2 KONFIGURASI SNORT

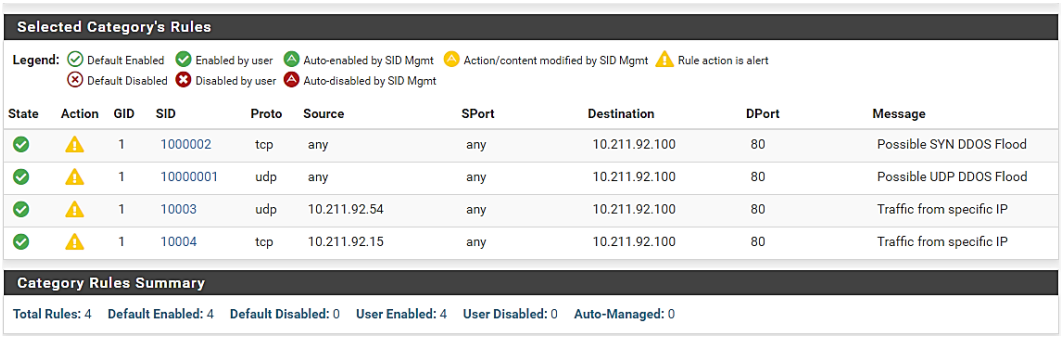
Konfigurasi Snort pada *pfSense* berfungsi sebagai *Intrusion Detection System* (IDS) yang dapat mendeteksi serangan jaringan dan mencegahnya. Dengan mengaktifkan Snort, dapat memantau dan menganalisis lalu lintas jaringan secara *real-time*, mendeteksi serangan seperti DDoS. Snort memberikan pemberitahuan dan *log* yang rinci tentang serangan yang terdeteksi, memungkinkan untuk merespons dengan cepat. Selain itu, Snort juga dapat melindungi jaringan dengan mengkonfigurasi tindakan respons terhadap serangan, seperti memblokir lalu lintas dari sumber yang mencurigakan. Integrasi yang baik dengan *pfSense* memudahkan konfigurasi dan pengelolaan Snort. Dengan Snort, dapat meningkatkan keamanan jaringan dan melindungi sumber daya jaringan yang penting seperti melindungi *server* dari serangan yang melalui jaringan WAN.



Gambar 3.14 Konfigurasi Snort

Dari gambar 3.14 menunjukkan Snort menggunakan *Legacy mode* sebagai *mode blocking* yang berfungsi untuk mendukung penggunaan versi Snort yang lebih lama atau aturan Snort yang tidak sepenuhnya kompatibel dengan versi Snort yang lebih baru yang didukung oleh *pfSense*. Jadi, *legacy mode* memungkinkan pemeliharaan ketersediaan layanan dan menjaga kontinuitas operasional dalam lingkungan *pfSense*.

Disisi lain Snort juga memiliki *custome rules* yang dibuat oleh pengguna, fungsinya sendiri yaitu untuk mendeteksi serangan yang spesifik dan unik yang tidak tercakup oleh aturan bawaan Snort. Aturan ini dapat dikustomisasi sesuai dengan kebutuhan spesifik jaringan, memungkinkan pengguna untuk mengadaptasi aturan Snort sesuai dengan lingkungan jaringan, mengurangi kesalahan positif, merespons serangan baru, dan menerapkan kebijakan keamanan yang khusus. Dengan *custom rules*, pengguna memiliki fleksibilitas dan kontrol lebih besar dalam deteksi serangan dan perlindungan jaringan, memungkinkan mereka untuk mengidentifikasi dan memonitor serangan yang lebih spesifik serta menerapkan langkah-langkah keamanan tambahan yang sesuai dengan kebijakan internal jaringan. Seperti pada gambar 3.15 menunjukkan *custome rules* Snort dengan status *active*.



Gambar 3.15 Tampilan *Custome Rules* Snort

Dari gambar 3.15 sebelumnya hanya menunjukkan 4 *custome rules active* yang akan digunakan, selanjutnya akan dijelaskan lebih spesifik bentuk dari keempat *rules* tersebut.

Rules pertama, untuk mendeteksi serangan DDoS SYN TCP *Flood* dapat dipergunakan perintah berikut:

```
drop tcp any any -> 10.211.92.100 80 (flags: S;
msg:"Possible SYN DDOS Flood"; flow: stateless;
threshold: type both, track by_dst, count 1000,
seconds 1; sid:1000002;rev:1;)
```

Rules pertama di atas merupakan *rules* Snort untuk mendeteksi serangan DDoS SYN TCP *Flood*, di mana aturan Snort ini menginstruksikan sistem untuk menjatuhkan (menghapus) paket TCP dengan flag SYN yang dikirim dari alamat sumber apa pun ke alamat tujuan 10.211.92.100 dengan *port* 80. Pesan yang akan dicetak jika aturan terpenuhi adalah "*Possible SYN DDOS Flood*" (Potensi Serangan Banjir SYN DDoS). Aturan ini diterapkan secara *stateless*, yang berarti tidak ada pelacakan status koneksi. Namun, aturan ini memiliki ambang batas pelaporan yang ditetapkan sebagai 1000 paket dalam 1 detik dengan alamat tujuan yang sama. Jika ambang batas tersebut tercapai, aturan ini akan memicu tindakan menjatuhkan paket-paket tersebut. ID keamanan (SID) aturan ini ditetapkan sebagai 1000002, dan ini adalah versi pertama dari aturan tersebut. Dengan demikian, aturan Snort ini bertujuan untuk mendeteksi dan mencegah potensi serangan banjir SYN DDoS dengan menjatuhkan paket-paket yang memenuhi kriteria yang ditetapkan. Jadi jika paket yang melanggar aturan tersebut maka akan ditolak secara langsung oleh *firewall*.

Rules kedua, untuk mendeteksi serangan DDoS UDP *Flood* dapat dipergunakan perintah berikut:

```
drop udp any any -> 10.211.92.100 80 (msg:"Possible
UDP DDOS Flood"; flow: stateless; threshold: type
both, track by_dst, count 1000, seconds 1;
sid:10000001;rev:1;)
```

Rules kedua di atas merupakan *rules* Snort untuk mendeteksi serangan DDoS UDP *Flood*, di mana aturan Snort ini menginstruksikan sistem untuk menjatuhkan (menghapus) paket UDP yang dikirim dari alamat sumber apa pun ke alamat tujuan 10.211.92.100 dengan *port* 80. Pesan yang akan dicetak jika aturan terpenuhi adalah "*Possible UDP DDOS Flood*" (Potensi Serangan Banjir UDP DDoS). Aturan ini diterapkan secara stateless, yang berarti tidak ada pelacakan status koneksi. Namun, aturan ini memiliki ambang batas pelaporan yang ditetapkan sebagai 1000 paket dalam 1 detik dengan alamat tujuan yang sama. Jika ambang batas tersebut tercapai, aturan ini akan memicu tindakan menjatuhkan paket-paket tersebut. ID keamanan (SID) aturan ini ditetapkan sebagai 10000001, dan ini adalah versi pertama dari aturan tersebut. Dengan demikian, aturan Snort ini bertujuan untuk mendeteksi dan mencegah potensi serangan banjir UDP DDoS dengan menjatuhkan paket-paket yang memenuhi kriteria yang ditetapkan. Jadi jika paket yang melanggar aturan tersebut maka akan ditolak secara langsung oleh *firewall*.

Rules ketiga, untuk mendeteksi lalu lintas paket TCP Normal dapat dipergunakan perintah berikut:

```
alert tcp 10.211.92.54 any -> 10.211.92.100 80
(flags: S; msg: "Traffic from specific IP";
sid:10004;)
```

Rules ketiga di atas merupakan *rules* Snort untuk mendeteksi paket TCP Normal, di mana aturan Snort ini bertujuan untuk memberikan peringatan ketika ada lalu lintas TCP Normal dari alamat IP spesifik 10.211.92.54 menuju alamat tujuan 10.211.92.100 dengan *port* 80. Aturan ini memeriksa paket-paket dengan flag SYN (*synchronize*) yang diatur. Jika aturan terpenuhi, maka akan dicetak pesan "*Traffic from specific IP*" sebagai peringatan. ID keamanan (SID) aturan ini ditetapkan sebagai 10004. Dengan demikian, aturan Snort ini berfungsi untuk memonitor dan memberikan peringatan terhadap lalu lintas TCP Normal yang berasal dari alamat IP spesifik tersebut.

Rules keempat, untuk mendeteksi lalu lintas paket UDP Normal dapat dipergunakan perintah berikut:

```
alert udp 10.211.92.54 any -> 10.211.92.100 80 (msg:
"Traffic from specific IP"; sid:10003;)
```

Rules keempat di atas merupakan *rules* Snort untuk mendeteksi paket UDP Normal, di mana aturan Snort ini berfungsi untuk memberikan peringatan saat ada lalu lintas UDP Normal yang berasal dari alamat IP spesifik 10.211.92.54 menuju alamat tujuan 10.211.92.100 dengan *port* 80. Ketika aturan terpenuhi, akan dicetak pesan "*Traffic from specific IP*" sebagai peringatan. ID keamanan (SID) aturan ini ditetapkan sebagai 10003. Dengan demikian, aturan Snort ini digunakan untuk memonitor dan memberikan peringatan terhadap lalu lintas UDP Normal yang berasal dari alamat IP spesifik tersebut.

3.5 PENGUJIAN TOPOLOGI

Pengujian topologi merupakan proses selanjutnya dalam melakukan pengujian IDS dan *Firewall* melalui uji *test* serangan DDoS menggunakan *software* Hping3. Selain itu, pengujian dari kedua sistem keamanan tersebut juga dilakukan dengan uji *test* lalu lintas normal menggunakan *software* yang sama. Proses pengujian dilakukan dengan menjalankan *script* perintah serangan DDoS dan lalu lintas Normal. Di mana *script* pada hping3 dapat berisi jenis paket yang dikirimkan, *port* tujuan, IP tujuan, dan lain-lain.

Script Hping3 pertama, untuk menjalankan serangan DDoS SYN TCP *Flood* dapat menggunakan perintah berikut:

```
sudo hping3 -S --faster --rand-source -p 80
10.211.92.100
```

Script Hping3 pertama menunjukkan perintah untuk melakukan uji *test* serangan DDoS dengan jenis paket TCP. Jadi dalam perintah ini, opsi -*S* digunakan untuk mengatur *flag* SYN pada paket TCP yang dikirimkan, dengan tujuan membuat koneksi TCP dengan target. Opsi –*faster* akan fokus pada pengiriman dan penerimaan paket dengan waktu yang lebih singkat. Jadi opsi ini memungkinkan untuk melakukan pengujian jaringan dengan kecepatan yang lebih tinggi dan mengoptimalkan waktu yang diperlukan untuk memeriksa konektivitas atau *respons* dari *host* atau jaringan yang dituju. Opsi --*rand-source* digunakan untuk

menggunakan alamat sumber acak pada paket serangan, dengan maksud menyembunyikan identitas penyerang. *Port* tujuan serangan ditentukan dengan opsi *-p* 80, yang mengindikasikan bahwa serangan ditujukan ke *port* 80, yang umumnya digunakan untuk layanan HTTP. Dengan menjalankan perintah ini, maka akan membanjiri target dengan lalu lintas yang besar dan mengakibatkan gangguan atau penurunan kinerja pada sistem atau layanan yang berjalan pada *port* 80.

Script Hping3 kedua, untuk menjalankan serangan DDoS UDP *Flood* dapat menggunakan perintah berikut:

```
sudo hping3 --udp --faster --rand-source -p 80
10.211.92.100
```

Script Hping3 kedua menunjukkan perintah untuk melakukan uji test serangan DDoS dengan jenis paket UDP. Jadi dalam perintah ini, yang membedakan dari perintah serangan sebelumnya adalah pada jenis paket yang dikirimkan yaitu dengan menggunakan perintah opsi *--udp* untuk menentukan bahwa serangan akan menggunakan protokol UDP, yang merupakan protokol tanpa koneksi yang sering digunakan untuk mengirimkan datagram dalam jaringan.

Script Hping3 ketiga, untuk menjalankan lalu lintas paket TCP Normal dapat menggunakan perintah berikut:

```
sudo hping3 -S -p 80 10.211.92.100
```

Script Hping3 ketiga menunjukkan perintah yang digunakan untuk mengirimkan lalu lintas atau paket normal menggunakan protokol TCP. Jadi dalam perintah ini, opsi *-S* menentukan penggunaan protokol TCP untuk mengirimkan paket. Opsi 10.211.92.54 merupakan alamat sumber paket, yang menunjukkan bahwa paket akan dikirimkan dari alamat IP tersebut. Opsi *-p* 80 dan 10.211.92.100 untuk menentukan *port* tujuan serta alamat tujuan paket, dalam hal ini adalah *port* 80 yang sering digunakan untuk layanan HTTP.

Script Hping3 keempat, untuk menjalankan lalu lintas paket UDP Normal dapat menggunakan perintah berikut:

```
sudo hping3 -2 -p 80 10.211.92.100
```

Script Hping3 keempat menunjukkan perintah yang digunakan untuk mengirimkan lalu lintas paket normal menggunakan protokol UDP. Jadi dalam

perintah ini, yang membedakan dari perintah lalu lintas normal sebelumnya adalah pada jenis paket yang dikirimkan yaitu dengan menggunakan perintah opsi *-2* untuk menentukan bahwa pengiriman lalu lintas normal akan menggunakan protokol UDP.

3.6 SKENARIO PENGUJIAN DAN PENGAMBILAN DATA

Tahap selanjutnya adalah skenario pengujian dan pengambilan data, di mana skenario yang digunakan adalah lalu lintas DDoS dan Normal. Kemudian data yang dibutuhkan pada penelitian ini adalah akurasi deteksi, efektivitas deteksi, kecepatan deteksi, dan penggunaan sumber daya sebagai evaluasi kinerja Snort.

3.6.1 SKENARIO PENGUJIAN

Pada Tabel 3.6 menunjukkan skenario dan metode pengujian yang digunakan untuk pengujian. Skenario pengujian yaitu DDoS dan Normal. Kasus pengujian yaitu TCP *Flood*, UDP *Flood*, TCP Normal, serta UDP Normal. Jadi, artinya skenario DDoS dan Normal disebar atau dibagi kembali menjadi 4 kasus atau jenis paket yaitu TCP *Flood*, UDP *Flood* untuk skenario DDoS. Sedangkan TCP Normal, UDP Normal untuk skenario Normal.

Tabel 3.6 Skenario pengujian dan kasusnya

Skenario Pengujian	Kasus Pengujian	
DDoS	TCP Flood	UDP Flood
Normal	TCP Normal	UDP Normal

Tabel 3.7 Hubungan skenario terhadap Hping3

Pengirim	Tools Pengiriman	Skenario	Paket		Penerima
PC Attacker	Hping3	DDoS	TCP Flood	UDP Flood	Target Server
PC Client1	Hping3	Normal	TCP Normal	UDP Normal	Target Server

Seperti pada Tabel 3.7 menunjukkan hubungan skenario terhadap Hping3, bahwa untuk mendapatkan cara pengujian deteksi, pencegahan dan notifikasi akan dilakukan pengiriman paket melalui PC *Attacker* dan PC *Client1* kepada Target *Server*. Pada PC *Attacker* nantinya akan menggunakan Hping3 untuk mengirimkan serangan DDoS dengan jenis paket TCP *Flood* dan UDP *Flood*. Kemudian pada PC *Client1* nantinya akan menggunakan Hping3 juga, namun untuk mengirimkan lalu

lintas Normal dengan jenis paket TCP Normal dan UDP Normal. Dengan demikian, dari pengiriman paket-paket tersebut, sistem nantinya mendeteksi paket melalui Snort dan menghasilkan *alert* sesuai dengan *custome rules* pada sub bab 3.4.2, yang selanjutnya pencegahanpun dilakukan melalui *pfSense* sebagai *firewall* dengan memblokir IP serangan. Setelah deteksi dan pencegahan dilakukan, notifikasi *email*pun diterapkan sebagai media pengiriman pesan *log* Snort kepada *admin*. Kemudian dari pesan *log* Snort, maka akan menjadi pengambilan data dari pengujian topologi yang selanjutnya sebagai evaluasi kinerja Snort melalui perhitungan parameter pengujian.

Tabel 3.8 Hubungan skenario terhadap deteksi dan pencegahan

Tools Pengujian	Fungsi	Skenario	Hasil	Notifikasi Email
Snort	Pendeteksi	DDoS dan Normal	Alert	log Snort
pfSense	Pencegahan	DDoS	Pemblokiran IP	-

Seperti pada Tabel 3.8 menunjukkan hubungan skenario terhadap deteksi dan pencegahan, bahwa dari pengiriman paket-paket skenario DDoS dan Normal, melalui Snort akan terdeteksi dan menghasilkan *alert* sesuai dengan *custome rules* pada sub bab 3.4.2, yang selanjutnya melalui *pfSense* dilakukan pemblokiran IP sebagai langkah pencegahan terhadap skenario DDoS. Setelah deteksi dan pencegahan dilakukan, notifikasi *email*pun diterapkan melalui fitur *pfSense* sebagai media pengiriman pesan *log alert* Snort setelah mendeteksi semua skenario kepada *admin.* Kemudian dari pesan *log* Snort, maka akan menjadi pengambilan data dari pengujian topologi serta sebagai evaluasi kinerja Snort melalui perhitungan parameter pengujian.

Tabel 3.9 Hubungan skenario terhadap parameter dan metode evaluasi

Parameter Pengujian	Skenario	Metode Evaluasi Kinerja Snort
Akurasi Deteksi	DDoS dan Normal	Klasifikasi Log Snort
Efektifitas Deteksi		
Kecepetan Deteksi	DDoS dan Normal	Standar Deviasi Detection Latency
Penggunaan Sumber Daya	DDoS dan Normal	Standar Deviasi CPU Usage

Seperti pada Tabel 3.9 menunjukkan hubungan skenario terhadap parameter dan metode evaluasi kinerja Snort, bahwa pada parameter akurasi dan efektifitas deteksi akan dilakukan evaluasi kinerja Snort melalui metode klasifikasi *log* Snort.

Jika mengacu pada *log* Snort, skenario DDoS dan Normal merupakan kumpulan dari *alert* yang meliputi TCP *Flood*, UDP *Flood*, TCP Normal, serta UDP Normal. Kemudian dari klasifikasi *log* Snort, skenario DDoS dan Normal merupakan kelas target untuk mengidetentifikasi sebuah *alert* dalam *log* Snort termasuk dalam kategori DDoS atau Normal. Selain itu, dari parameter akurasi juga menunjukkan ada korelasi dengan parameter efektifitas, di mana dari kedua parameter tersebut masuk pada hasil kinerja Snort yang sama yaitu nilai atay metrik sebaran dari *confusion matrix*, meliputi *accuracy*, *precision*, *recall* (TPR), *specificity*, *f1-score*, serta FPR.

Lalu pada parameter kecepatan deteksi akan dilakukan evaluasi kinerja Snort melalui metode *detection latency* terhadap skenario DDoS dan Normal, di mana dari metode tersebut akan memperoleh hasil kinerja Snort melalui standar devisi sebagai pengukuran untuk mengetahui variasi atau fluktuasi dari nilai rata-rata setiap skenario. Kemudian pada parameter penggunaan sumber daya akan dilakukan evaluasi kinerja Snort melalui penggunaan sumber daya CPU terhadap skenario DDoS dan Normal, di mana dari penggunaan sumber daya tersebut akan memperoleh hasil kinerja Snort melalui standar devisi sebagai pengukuran untuk mengetahui variasi atau fluktuasi dari nilai rata-rata setiap skenario.

3.6.2 PENGAMBILAN DATA

Pengambilan data pada penelitian ini meliputi *log* Snort, klasifikasi, dan standar deviasi. Pengambilan data ini nantinya sebagai evaluasi kinerja Snort berdasarkan parameter yang sudah dijelaskan sebelumnya pada sub bab 3.6.1.

3.6.2.1 LOG SNORT

Pengambilan *log* Snort dilakukan dengan mengirimkan satu kali skenario DDoS dan satu kali skenario Normal, artinya dari setiap skenario ada 2 pengiriman kasus atau paket. Total ada 4 kali pengiriman yaitu TCP *Flood*, UDP *Flood*, TCP Normal, UDP Normal. Seperti pada Tabel 3.10 yang menunjukkan pengambilan data *log* Snort.

Tabel 3.10 Pengambilan data *log* Snort

Skenario	Jenis paket	Pengiriman	Waktu	Hasil notifikasi
Skenario DDoS	TCP Flood	1 kali	30 menit	1510 Alert
	UDP Flood	1 kali	30 menit	1804 Alert
Skenario Normal	TCP Normal	1 kali	10 menit	518 Alert
	UDP Normal	1 kali	10 menit	610 Alert

Jadi, proses atau cara pengiriman dilakukan sesuai pada sub bab 3.5. Setiap pengiriman skenario DDoS dikirimkan dalam waktu 30 menit, lalu pada setiap pengiriman skenario Normal dikirimkan dalam waktu 10 menit. Kemudian dari proses pengiriman sebelumnya, akan diperoleh pesan *email* melalui notifikasi *email* secara *periodic* sebagai pengambilan data *log* Snort yang berisi *alert* atau peringatan. Jumlah peringatan terdiri dari 1510 *alert* TCP *Flood*, 1804 *alert* UDP *Flood*, 518 *alert* TCP Normal, serta 610 *alert* UDP Normal. Jadi paket-paket tersebut mewakili skenario DDoS dan Normal dengan total keduanya berjumlah 4442 alert. Kemudian dari skenario dan jumlah alert tersebut, akan dilakukan klasifikasi terhadap *log* snort.

3.6.2.2 KLASIFIKASI

Pengambilan data klasifikasi yaitu dengan cara mengambil data riwayat paket atau *alert* (*log* Snort) melalui notifikasi *email* secara *periodic* yang dikirimkan kepada *admin*. Kemudian data tersebut dikelompokkan dalam bentuk tabel *Excel* menjadi beberapa atribut, meliputi atribut Protokol, IP sumber, *Port* Sumber, IP tujuan, *Port* tujuan, dan *Label*. Total isi *log* Snort tersebut berjumlah 4442 *alert* dan digunakan menjadi 1 data untuk mengklasifikasikan suatu *alert* termasuk dalam kategori lalu lintas DDoS atau Normal. Proses klasifikasi menggunakan algoritma *decision tree*. Selanjutnya dari proses klasifikasi menggunakan algoritma tersebut akan menghasilkan nilai *confusion matrix* sebagai nilai kinerja Snort. Nilai kinerjanya meliputi *accuracy, precision, recall* (TPR), *specificity, F1-Score* serta FPR. Pada Tabel 3.11 menunjukkan deskripsi atribut yang dipakai dalam proses klasifikasi *log* Snort.

Tabel 3.11 Deskripsi Atribut

Atribut	Deskripsi
IP Sumber	Alamat IP dari Pengirim
IP Tujuan	Alamat IP dari Penerima
Protokol	Protokol yang dikirimkan
Port Sumber	Port dari Pengirim
Port Tujuan	Port dari Penerima
Label	Identitas lalu lintas jaringan

Metode analisis ini digunakan untuk menghitung persentase akurasi dari hasil pengujian berdasarkan akurasi deteksi. Sehingga, data-data yang didapatkan akan diolah menjadi bentuk persentase angka dengan skala yang menentukan seberapa akurat Snort dalam mendeteksi serangan skenario DDoS dan Lalu lintas Normal dengan masing-masing jenis protokol yaitu TCP dan UDP. Jadi, untuk menghitung dan mengolah keakuratan data-data yang dihasilkan, pada penelitian ini menerapkan metode klasifikasi *log* pada Snort secara sederhana menggunakan *Software Orange*. Metode ini juga menerapkan algoritma *decision tree*. *Output* data berupa nilai prediksi benar dan salah melalui *confusion matrix*, meliputi nilai TP, TN, FP, FN yang kemudian dari nilai tersebut akan menghasilkan nilai persentase *accuracy, precision, recall, specificity, f1-score*, serta FPR.

Tabel 3.12 *Confusion Matrix*

	Predicted Class	
Actual Class	Predicted DDoS	Predicted Normal
Actual DDoS	*True Positive* (TP)	*False Negative* (FN)
Actual Normal	*False Positive* (FP)	*True Negative* (TN)

Seperti pada Tabel 3.12 merupakan *Confusion matrix* 2x2 dalam mengklasifikan *log* Snort dengan *label* atau target DDoS dan Normal. Confusion matrix pada dasarnya merupakan salah satu metode evaluasi yang digunakan dalam analisis klasifikasi untuk mengukur kinerja model atau algoritma. *Confusion matrix* memberikan gambaran tentang sejauh mana model mampu mengklasifikasikan data dengan benar atau salah. Jadi *confusion matrix* akan menghasilkan empat nilai: *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN)[39].

A. Akurasi Deteksi

Salah satu sebaran matrik dari *confusion matrix* adalah akurasi (*accuracy*). Tujuan dari perhitungan akurasi adalah mengukur sejauh mana model mampu mengklasifikasikan dengan benar seluruh sampel. Seperti pada persamaan 3.1 merupakan cara perhitungan matrik akurasi.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \quad (3.1)$$

Keterangan:

a. TP = *True Positive* (trafik paket data yang diidentifikasikan DDoS dan merupakan paket DDoS)
b. TN = *True Negative* (trafik paket data yang diidentifikasikan paket data normal dan merupakan paket data normal)
c. FP = *False Positive* (trafik paket yang diidentifikasikan paket data normal dan merupakan paket data DDoS)
d. FN = *False Negative* (trafik paket data yang diidentifikasikan DDoS dan merupakan sebagai paket normal)

B. Efektifitas Deteksi

Confusion matrix juga memiliki sebaran metrik yang lain selain *accuracy*, meliputi metrik *Precision, Recall, Specificity, F1-Score* serta FPR. Di mana kelima metrik tersebut termasuk dari efektifitas deteksi, artinya ada korelasi antara akurasi deteksi dengan efektifitas deteksi, Seperti pada persamaan 3.2 yang menunjukkan cara perhitungan metrik *precision*.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP+FP})} \quad (3.2)$$

Tujuan dari perhitungan *Precision* adalah mengukur sejauh mana prediksi positif yang dilakukan oleh sistem benar-benar merupakan positif.

Setelah perhitungan metrik *precision* selesai, maka selanjutnya dapat melakukan perhitungan metrik *Recall* (TPR), seperti pada persamaan 3.3 berikut.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP+FN})} \quad (3.3)$$

Tujuan dari perhitungan *Recall* adalah mengukur sejauh mana sistem mampu mendeteksi atau mengambil kembali *instance* positif yang sebenarnya.

Kemudian melakukan perhitungan metrik *Specificity*, yaitu lawan perhitungan dari metrik *Recall*, seperti pada persamaan 3.4 berikut.

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{(\boldsymbol{TN+FP})} \qquad (3.4)$$

Tujuan dari perhitungan *Specificity* adalah mengukur sejauh mana sistem dapat mengidentifikasi semua contoh negatif dengan benar.

Jika perhitungan *Precision* dan *Recall* selesai, kemudian dapat melakukan perhitungan metrik *F1-Score*, seperti pada persamaan 3.5 berikut.

$$\boldsymbol{F1 - Score} = \frac{\boldsymbol{2\times(precision\times recall)}}{(\boldsymbol{precision+recal})} \quad (3.5)$$

Tujuan dari perhitungan *F1-Score* adalah mengetahui ukuran yang menggabungkan *Precision* dan *Recall* dalam satu angka, memberikan keseimbangan antara keduanya.

Selain perhitungan *Recall*, *Specificity*, dan *F1-Score*, ada juga perhitungan mengenai *False Positive Rate* (FPR), seperti pada persamaan 3.6 berikut.

$$\boldsymbol{FPR} = \frac{\boldsymbol{FP}}{(\boldsymbol{FP+TN})} \qquad (3.6)$$

Tujuan dari perhitungan *False Positive Rate* (FPR) adalah untuk mengukur seberapa sering sistem memberikan hasil positif palsu, yaitu ketika sistem secara keliru mengklasifikasikan data yang sebenarnya negatif (Normal) sebagai positif (DDoS).

3.6.2.3 STANDAR DEVIASI

Pengambilan data Standar Deviasi dilakukan untuk mengetahui variasi atau fluktuasi dari nilai rata-rata yang kemudian sebagai nilai kinerja Snort berdasarkan parameter kecepatan deteksi dan penggunaan sumber daya. Pada persamaan 3.7 berikut merupakan persamaan dari Standar Deviasi[40].

$$s = \sqrt{\left[\sum \frac{(xi-\bar{x})^2}{(n-1)}\right]} \quad (3.7)$$

Keterangan:

1. *s* adalah standar deviasi tunggal
2. *Σ* adalah simbol penjumlahan
3. *xi* adalah nilai setiap data dalam sampel
4. $\bar{x}$ adalah rata-rata sampel
5. *n* adalah jumlah total data dalam sampel

A. Kecepatan Deteksi

Pada pengambilan data kecepatan deteksi, yaitu dengan cara mengumpulkan data *detection latency* dari 10 percobaan setiap skenario yaitu DDoS dan Normal, yang masing-masing skenario dibagi menjadi 2 jenis paket. Total ada 40 percobaan. Seperti pada Tabel 3.13.

Tabel 3.13 Pengambilan data *Detection latency*

Skenario	Detection Latency	Percobaan	Waktu/percobaan
Skenario DDoS	TCP Flood	10	5 menit
	UDP Flood	10	5 menit
Skenario Normal	TCP Normal	10	5 menit
	UDP Normal	10	5 menit

Data *detection latency* diperoleh dari setiap pengiriman paket dan *packet capture* melalui *pfSense* selama 5 menit, kemudian dari hasil *packet capture* direpresentasikan melalui *Wireshark*. *Detection latency* pada penelitian ini adalah selisih antara waktu deteksi terhadap paket tersebut oleh Snort (T2) dengan waktu awal mulainya paket serangan atau normal yang masuk atau tercapture *pfSense* (T1). Seperti pada persamaan 3.8, merupakan persamaan selisih T2 dengan T1[41].

$$Detection\ Latency\ = T2 - T1 \quad (3.8)$$

Keterangan:

a. T1 = Waktu awal mulai paket DDoS atau Normal masuk ke *pfSense*

b. T2 = Waktu paket DDoS atau Normal terdeteksi oleh Snort

Setelah hasil *Detection Latency* diperoleh kemudian melakukan standar deviasi dengan cara menghitung rata-ratanya dari data yang dimiliki. Rata-rata dapat dihitung dengan menjumlahkan semua nilai *detection latency* dan membaginya dengan jumlah pengambilan data. Seperti pada persamaan 3.9 berikut.

$$\bar{x} = \frac{\sum xi}{n} \qquad (3.9)$$

Setelah rata-rata kecepatan deteksi diperoleh maka selanjutnya dapat menghitung selisih antara setiap data individu dengan rata-rata kemudian dikuadratkan, seperti pada persamaan 3.10 berikut.

$$Kuadrat\ Selisih\ = (xi - \bar{x})^2 \qquad (3.10)$$

Jika kuadrat selisih diperoleh maka selanjutnya menjumlahkan setiap kuadrat selisih yang dihitung, seperti pada persamaan 3.11 berikut.

$$Jumlah\ Kuadrat\ Selisih\ = \sum (xi - \bar{x})^2 \qquad (3.\ 11)$$

Setelah Jumlah Kuadrat Selisih diperoleh, maka melakukan perhitungan akar kuadrat dari rata-rata kuadrat sebagai langkah terakhir untuk mendapatkan nilai standar deviasi dari *detection latency*.

B. Penggunaan Sumber Daya

Pada pengambilan data penggunaan sumber daya akan diperoleh melalui *pfSense* untuk memantau penggunaan CPU terhadap skenario DDoS dan Normal dalam bentuk persentase. Pada pengumpulan data penggunaan CPU skenario DDoS nantinya diambil dari kondisi atau lonjakan awal paket TCP *Flood* dan UDP *Flood* dikirimkan, lalu pengumpulan data penggunaan CPU skenario Normal nantinya dari kondisi terakhir 200 paket TCP Normal dan UDP Normal dikirimkan. Seperti pada Tabel 3.14, menunjukkan pengambilan data penggunaan sumber daya.

Tabel 3.14 Pengambilan data Penggunaan sumber daya

Skenario	CPU Usage	Percobaan	Kondisi pengambilan data
Skenario DDoS	TCP Flood	10	Kondisi awal paket dikirimkan
	UDP Flood	10	Kondisi awal paket dikirimkan
Skenario Normal	TCP Normal	10	Kondisi terakhir paket dikirimkan
	UDP Normal	10	Kondisi terakhir paket dikirimkan

Kemudian, untuk proses perhitungannya adalah dengan mengubah nilai persentase yang diperoleh menjadi *decimal* sebagai langkah awal perhitungan. Ini dilakukan karena persentase mewakili perbandingan terhadap seratus yang memberikan nilai desimal yang sesuai. Seperti pada persamaan 3.12 berikut, yang menunjukkan cara menghitung nilai *decimal* dari persentase.

$$Decimal = \frac{xi}{n} \qquad (3.12)$$

Jadi, xi adalah nilai penggunaan CPU setiap percobaan yang diperoleh, dan n adalah 100 yang mewakili 100% dari nilai persentase. Setelah memperoleh kumpulan data persentase dengan nilai *decimal* penggunaan CPU maka selanjutnya dapat menghitung rata-ratanya. Lalu jika rata-rata penggunaan CPU diperoleh, maka dilakukanlah perhitungan varian populasinya seperti pada persamaan 3.10 dan persamaan 3.11 yaitu kuadrat selisih dan jumlah kuadrat selisih, yang selanjutnya dapat menghitung standar deviasi penggunaan CPU untuk mengukur variasi atau fluktuasi dari nilai rata-rata dalam penggunaan CPU.