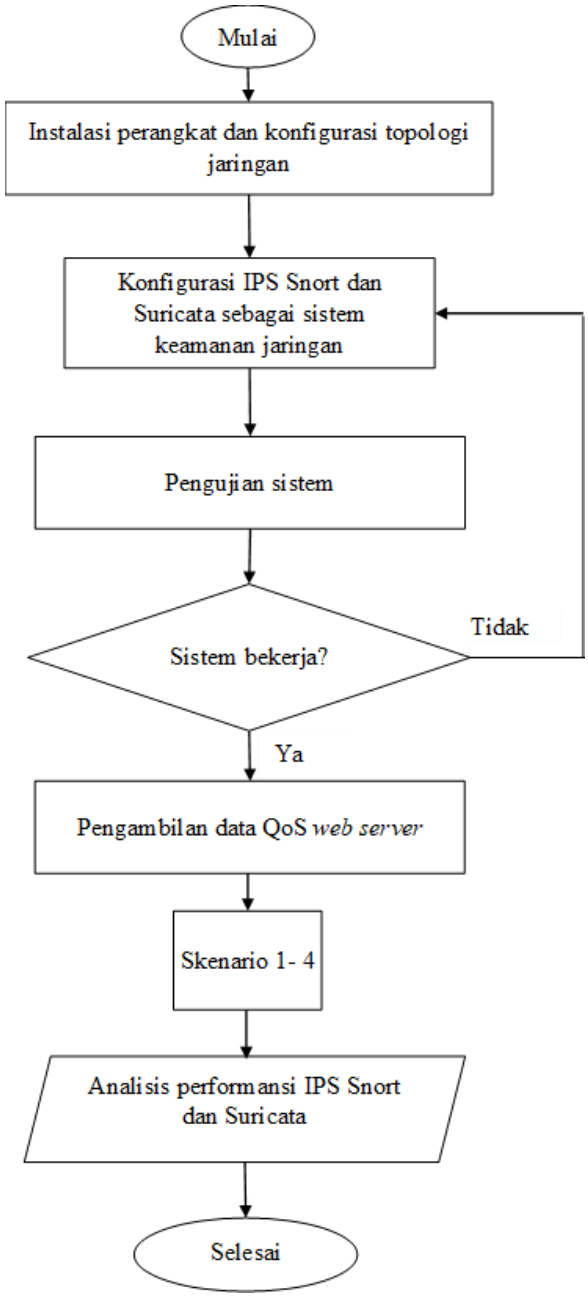
BAB III
METODE PENELITIAN

3.1 ALUR PENELITIAN

Penelitian ini dilakukan dengan menggunakan tahapan untuk memenuhi s egala aspek yang diperlukan. Gambar 3.1 merupakan *flowchart* yang menggambarkan alur kerja dari implementasi penelitian yang akan dilaksanakan.



Gambar 3. 1 *Flowchart* Implementasi Penelitian

Gambar 3.1 merupakan *flowchart* yang mengGambarkan alur implementasi dari penlitian yang akan dilakukan. Berdasarkan flowchart tersebut implementasi penelitian dimulai dengan tahap installasi perangkat yang akan digunakan dan konfigurasi topologi jaringan. Perangkat yang akan diinstall untuk penelitian ini antara lain yaitu pfSense, *client* dan *web server*, sedangkan untuk topologi jaringan menggunakan prinsip *client – server*. Tahap selanjutnya yaitu konfigurasi IPS Snort dan Suricata sebagai sistem keamanan jaringan.

Pada tahap ini dilakukan konfigurasi *rule* yang akan digunakan untuk mencegah serangan *TCP SYN flood* dari *client*. Kemudian dilakukan pengujian terhadap *rule* yang sudah dikonfigurasi. Pengujian dilakukan dengan melakukan *TCP SYN flood* dari *client* ke *web server*. Jika IPS Snort dan Suricata dapat melakukan blok terhadap *TCP SYN flood* maka akan dilanjutkan ke tahap selanjutnya, sedangkan jika *rule* dari IPS Snort atau Suricata tidak dapat melakukan blok terhadap *TCP SYN flood* maka konfigurasi IPS dari Snort atau Suricata akan diulang.

Apabila sistem IPS Snort dan Suricata sudah berfungsi dengan baik maka dilakukan pengambilan data dari QoS *web server*. Pengambilan data akan dilakukan pada 4 skenario. Skenario 1 yaitu ketika kondisi normal atau tidak ada serangan *TCP SYN flood*. Skenario 2 yaitu kondisi ketika *web server* diserang tanpa mengaktifkan IPS Snort atau Suricata. Skenario 3 yaitu kondisi ketika *web server* diserang dengan mengaktifkan IPS Snort. Skenario 4 yaitu kondisi ketika *web server* diserang dengan mengaktifkan IPS Suricata. Selanjutnya dilakukan analisis perbandingan performa IPS Snort dan IPS Suricata dalam mengatasi *TCP SYN flood* berdasarkan QoS *web server* pada 4 skenario pengujian tersebut.

3.2 ALAT DAN BAHAN

Dalam penelitian ini membutuhkan alat dan bahan untuk menunjang pengujian dan pengambilan data. Alat dan bahan yang dibutuhkan berupa *hardware* dan *software* yang akan digunakan dalam proses pengujian dan pengambilan data pada penelitian ini.

3.2.1 *HARDWARE*

Beberapa perangkat keras (*hardware*) yang dibutuhkan untuk implemntasi penelitian ini yaitu :

1. *Personal computer* (PC)

   Pada Tabel 3.1 merupakan spesifikasi dari PC yang akan digunakan untuk implementasi penelitian. Perangkat ini akan digunakan untuk instalasi VM Ubuntu *Server* sebagai *web server*, kali linux sebagai *attacker*, dan perangkat pfSense yang didalamnya terdapat IPS Snort dan Suricata.

Tabel 3.1 Spesifikasi PC

Spesifikasi	Data
Operating System	Windows 10
Processor	Intel *Core* i5 6500 @3.2 Ghz x 4
RAM	8 GB
SSD	120 GB

2. Laptop

   Spesifikasi laptop yang digunakan dapat dilihat pada Tabel 3.2 dimana laptop tersebut digunakan sebagai *client* yang mengakses *web server*. Perangkat laptop menggunakan Apacje Jmeter untuk melakukan *request* HTTP dan aplikasi Wireshark untuk melakukan *capture* lalu lintas jaringan

Tabel 3.2 Spesifikasi Laptop

Spesifikasi	Data
Operating System	Windows 11
Processor	AMD Dual-*core* A9-9420e
RAM	4 GB
Hard disk	1 TB
SSD	120 GB

3.2.2 *SOFTWARE*

Beberapa perangkat lunak yang digunakan pada penelitian ini antara lain yaitu :

a. Ubuntu *Server*

Pada sistem yang akan dibuat menggunakan Ubuntu *Web server* sebagai *web server*. Perangkat lunak ini akan diinstall sebagai virtual machine. Perangkat ini akan menggunakan Apache2 sebagai layanan *web server*. Layanan Apache2 *web server* tersebut akan menjadi target serangan *data flooding* oleh *attacker*. *Software* ini merupakan virtual machine dengan spesifikasi seperti pada Tabel 3.3.

Tabel 3.3 Spesifikasi VM Ubuntu *Server*

Spesifikasi	**Data**
Operating System	Ubuntu (64 bit)
Processor	1 *core*
RAM	1 GB
Storage	15 GB

b. Perangkat pfSense

Perangkat pfSense merupakan *virtual machine* yang digunakan sebagai *router* dengan banyak tambahan fitur yang akan digunakan pada sistem. Pada perangkat lunak ini akan diterapkan sistem keamanan jaringan menggunakan IPS Snort dan Suricata. IPS Snort dan Suricata merupakan fitur dalam bentuk paket yang dapat diinstall pada pfSense. IPS Snort dan Suricata akan diaktifkan secara bergantian dalam proses pengambilan data. Tabel 3.4 merupakan spesifikasi *virtual machine* pfSense.

Tabel 3.4 Spesifikasi *Virtual Machine* pfSense

Spesifikasi	**Data**
Operating System	FreeBSD (64bit)
Processor	1 *core*
RAM	2 GB
Storage	16 GB

c. Kali Linux

Kali linux merupakan *virtual machine* yang berperan sebagai *attacker*. Pada *virtual machine* ini akan diinstal *tools* Nping yang akan digunakan untuk mengirimkan *TCP SYN flood* ke *web server*. Spesifikasi *Virtual machine* kali linux ditunjukkan pada Tabel 3.5.

Tabel 3.5 Spesifikasi *Virtual Machine* Kali Linux

Spesifikasi	**Data**
Operating System	Debian (64 bit)
Processor	2 *core*
RAM	4 GB
Storage	80 GB

d. Wireshark

Wireshark merupakan perangkat lunak yang berfungsi sebagai *network protocol analyzer*. Perangkat ini dapat merekam semua paket yang lewat serta menyeleksi dan menampilkan data tersebut dengan detail. Perangkat ini terpasang langsung pada laptop untuk memperoleh data QoS layanan *web server* dari sisi *client*.

5. Apache Jmeter

Aplikasi Apache Jmeter merupakan perangkat lunak yang akan digunakan untuk menguji kinerja *web server* pada beberapa skenario pengujian. Aplikasi ini dipasang pada perangkat laptop yang akan berperan sebagai *client* yang mengakses *web server* secara konstan.

6. Tshark

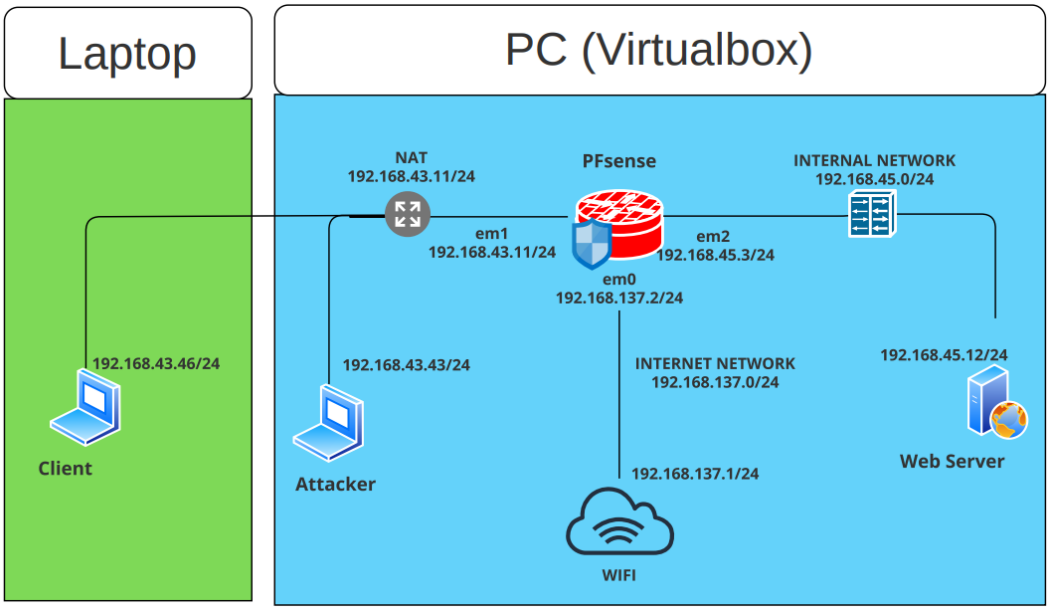
Aplikasi Tshark merupakan perangkat lunak yang digunakan untuk menganalisis *file* dari hasil *Capture* wireshark. Pada penelitian ini Tshark dipasang pada perangkat laptop yang digunakan untuk menganalisis *Round Trip Time* dari setiap pengujian.

3.3 IMPLEMENTASI PENELITIAN

Implementasi penelitian merupakan tahap implementasi dari penelitian berdasarkan alur penelitian. Implementasi penelitian dipersiapkan dengan merancang dan mengkonfigurasi topologi jaringan, konfigurasi *web server*, konfigurasi pfSense, konfigurasi IPS Snort dan IPS Suricata dan konfigurasi Nping. Setelah itu implementasi penelitian dilanjutkan dengan melakukan pengujian berdasarkan skenario pengujian. Berdasarkan pengujian tersebut maka dilakukan pengambilan data untuk membandingkan kinerja IPS Snort dan IPS Suricata.

3.3.1 TOPOLOGI JARINGAN

Topologi jaringan yang digunakan pada penelitian ini akan menunjukkan peran dari perangkat keras dan perangkat lunak yang dijalankan. Topologi jaringan yang diterapkan pada penelitian ini ditunjukkan pada Gambar 3.2 sebagai berikut :



Gambar 3.2 Topologi Jaringan

Topologi jaringan pada Gambar 3.2 terdiri dari 2 perangkat keras yaitu 1 PC dan 1 laptop. Perangkat PC pada topologi Gambar 3.2 berfungsi untuk menjalankan 3 *virtual machine* (VM) yang digunakan pada penelitian ini. VM yang digunakan anatara lain yaitu Ubuntu *server*, pfSense, dan Kali linux. VM Ubuntu *server*

berperan sebagai *server* yang menyajikan layanan *web server* kepada *client*. VM pfSense berperan sebagai *router* yang menghubungkan *client* dan *attacker* untuk dapat mengakses *web server*. Selain itu pfSense juga menerapkan IPS Snort dan Suricata sebagai sistem kemanan jaringan yang melindungi kinerja *web server*. VM Kali linux berperan sebagai *attacker* yang melakukan serangan terhadap *web server* dengan cara mengirimkan SYN *flood* menggunakan *tools* Nping.

Perangkat laptop pada topologi Gambar 3.2 berfungsi sebagai *client* yang akan mengakses layanan *web server*. Perangkat PC dan laptop dihubungkan menggunakan perangkat wifi. Laptop menggunakan aplikasi Apache Jmeter untuk melakukan *request* layanan HTTP secara konstan selama waktu pengujian yaitu 120 detik. Pada perangkat ini juga menjalankan aplikasi wireshark untuk menangkap (*capture*) lalu lintas jaringan antara *client* dan *server*. Hasil *capture* lalu lintas jaringan akan digunakan untuk menganalisis dampak serangan *data flooding* dan performansi IPS Snort dan Suricata dalam mengatasi serangan *data flooding*.

Pada topologi tersebut terdapat penjelasan internal *network*, *internet network* dan NAT . Internal *network* merupakan jaringan yang menghubungkan VM *web server* secara langsung (*direct access*) dengan pfSense, sehingga jaringan ini harus dilindungi dari serangan *attacker*. *Internet network* merupakan *network* yang digunakan *virtual machine* untuk mendapatkan layanan *internet* dari perangkat wifi . NAT merupakan *WAN address* dari pfSense yang menghubungkan *client* dan *attacker* ke alamat IP *web server* menggunakan konsep *port forwarding*. Alamat IP yang digunakan pada topologi jaringan ditampilkan pada Tabel 3.6.

Tabel 3.6 Pengalokasian Alamat IP

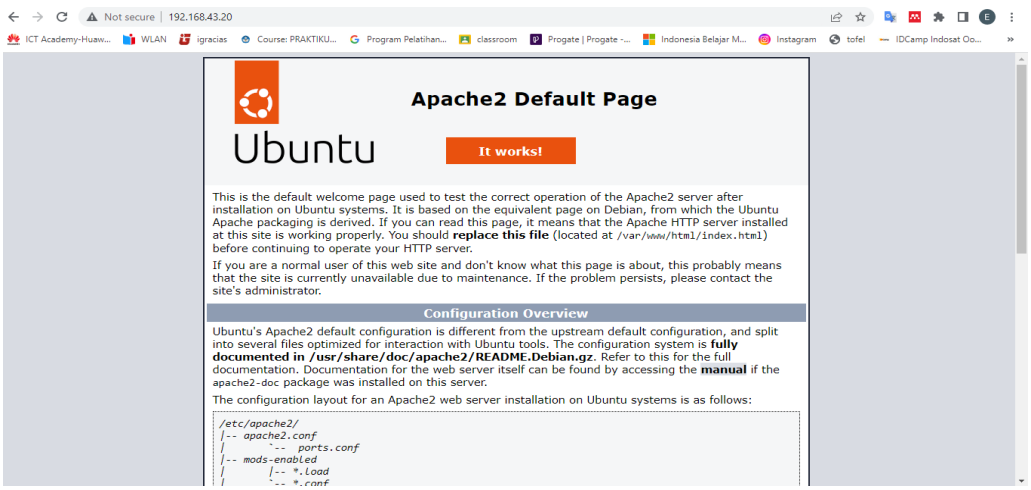
Perangkat	*Interface*	Alamat IP
pfSense	em0	192.168.43.11/24
	em1	192.168.137.2/24
	em2	192.168.45.3/24
Web server	Eth0	192.168.45.12/24
Client	Eth0	192.168.43.46/24
Attacker	Eth0	192.168.43.44/24
Wifi	*Host Only Adapter*	192.168.137.1/24

3.3.2 KONFIGURASI *WEB SERVER*

Web server dipasang pada virtual mesin Ubuntu *server*. *Web server* yang digunakan pada penelitian ini yaitu Apache2 *web server*. Konfigurasi *web* pada VM *Ubuntu web server* menggunakan mode *command-line Interface* (CLI). *Web server* dikonfigurasi menggunakan alamat IP 192.168.45.12. Proses instalasi Apache2 adalah sebagai berikut :

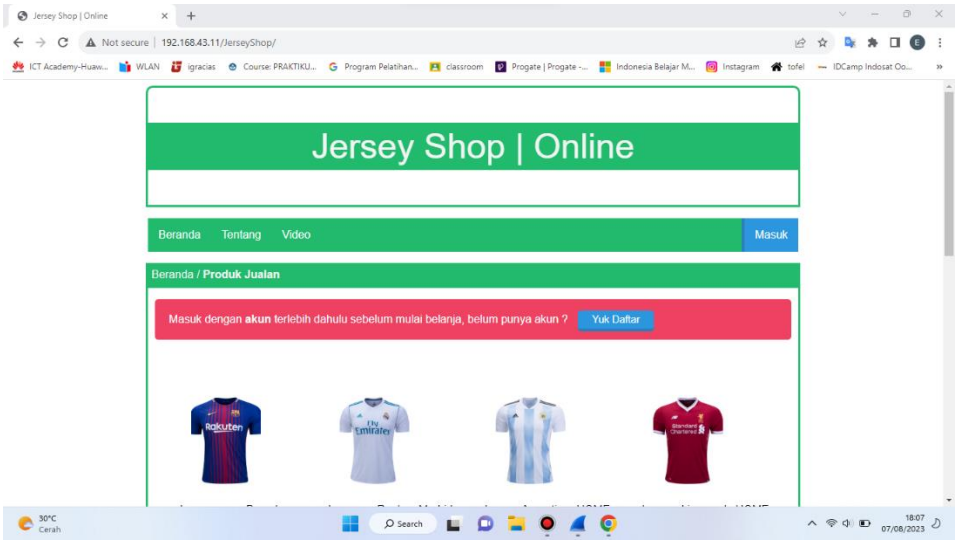
1. *Sudo apt update*
2. *Sudo apt install Apache2*
3. *Sudo ufw app list*
4. *Sudo ufw allow 'Apache'*
5. *Sudo ufw status*
6. *Sudo service Apache2 start*

Setelah proses instalasi Apache2 selesai, dilanjutkan dengan test *service* dari *web server* Apache2. *Test service* dilakukan dengan cara membuka alamat *web server* pada perangkat *client*. *Client* memasukan alamat *web server* Apache 2 pada *web* browsernya. hasil test *service web server* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Default Page Apache2

Selanjutnya, pada penelitian ini menambahkan halaman *web jersey shop* sebagai layanan yang dijalankan web server. Tampilan beranda dari halaman *web* yang digunakan pada penelitian ini ditampilkan pada Gambar 3.4.

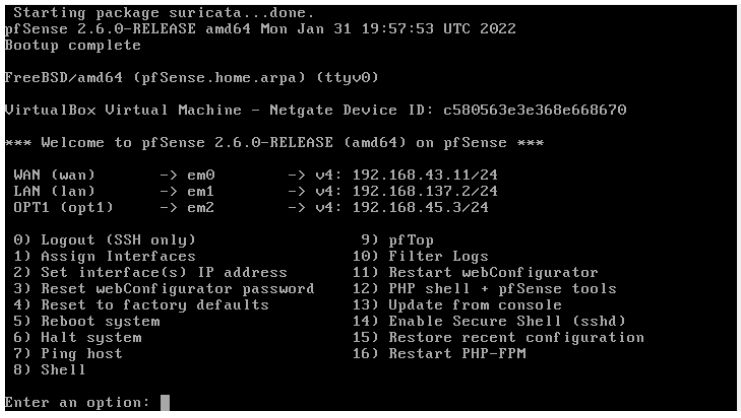


Gambar 3.4 Tampilan Beranda *Jersey Shop*

3.3.3 KONFIGURASI PFSENSE

Virtual machine (VM) pfSense perlu dikonfigurasi agar dapat terhubung dengan *VM web server* , *VM client* dan *VM attacker*. Konfigurasi meliputi pengaturan alamat IP *Interface* pada pfSense. *Interface* yang dikonfigurasi pada pfSense ditunjukkan pada Gambar 3.5.

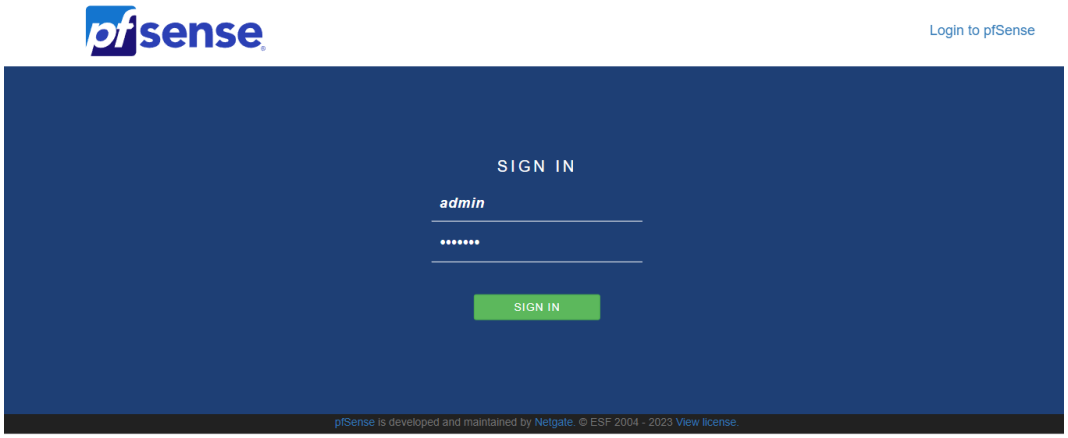
1. em0 (WAN) = 192.168.43.11/24
2. em1 (LAN) = 192.168.137.2/24
3. em2 (OPT1) = 192. 168.45.3/24



Gambar 3.5 Konfigurasi *Interface* pfSense

Setelah konfigurasi *Interface* selesai, selanjutnya pfSense dapat diakses melalui halaman *browser*. Akses pada halaman pfSense dengan cara memasukan alamat LAN pfSense yaitu 192.168.137.2 pada browser. Selanjutnya Admin harus

memasukan username dan *password* untuk dapat masuk ke halaman pfSense untuk dapat menggunakan layanan pfSense. Halaman pfSense dapat dilihat pada Gambar 3.6.

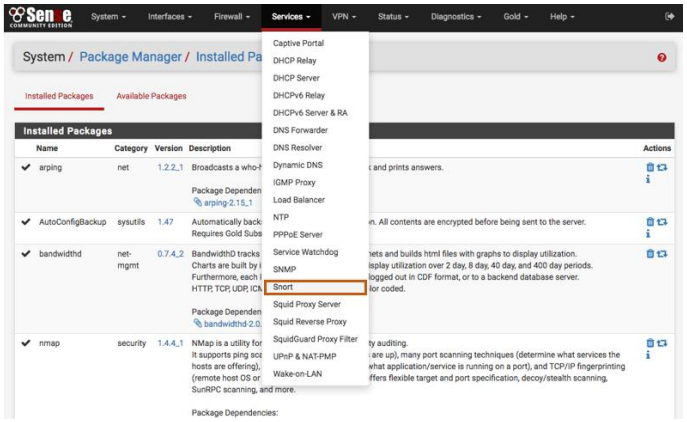


Gambar 3.6 Halaman *Login* pfSense

3.3.4 KONFIGURASI IPS SNORT

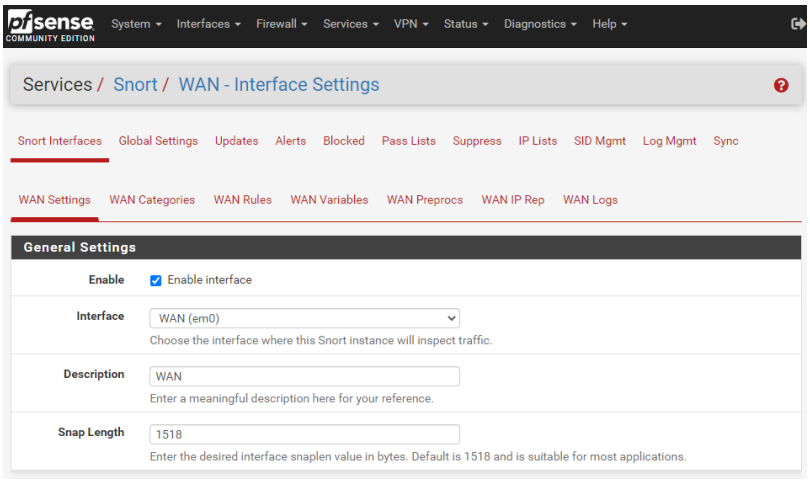
Snort tersedia dalam bentuk paket yang dapat diinstal pada GUI pfSense. Konfigurasi Snort dilakuakan dengan langkah – langkah sebagai berikut :

1. Insatalasi Snort berada pada menu *system* > Package manager > available packages. Jika sudah terinstal maka Snort akan muncul pada menu *service* pada GUI pfSense yang ditampilakan pada Gambar 3.7.



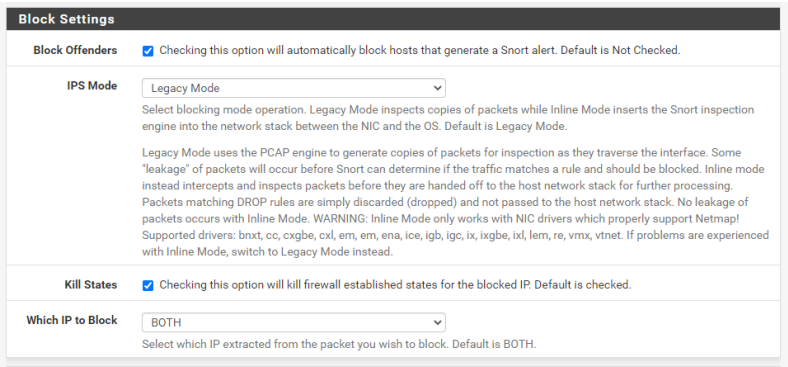
Gambar 3.7 *Service* Snort Pada GUI pfSense

2. Kemudian menambahkan Snort pada *Interface* yang akan menerapkan Snort sebagai IPS. Pada penelitian ini akan menerapkan Snort pada *interface* WAN seperti pada Gambar 3.8.



Gambar 3.8 Snort *Interface*

3. Setelah *Interface* Snort ditetapkan, maka dilakukan konfigurasi pengaturan *blocking* seperti pada Gambar 3.9. Untuk mengaktifkan metode IPS pada Snort dilakukan dengan *cheklist* pada *menu Block Offenders*. IPS pada penelitian ini menggunakan *Legacy Mode*. *Legacy mode* merupakan operasi pemblokiran menggunakan mesin PCAP untuk menghasilkan salinan paket yang kemudian diperiksa saat melintasi *interface* jaringan. Paket yang disalin kemudian dicocokan dengan *rule* yang sudah diterapkan untuk menentukan action terhadap lalu lintas jaringan.



Gambar 3.9 *Block Settings*

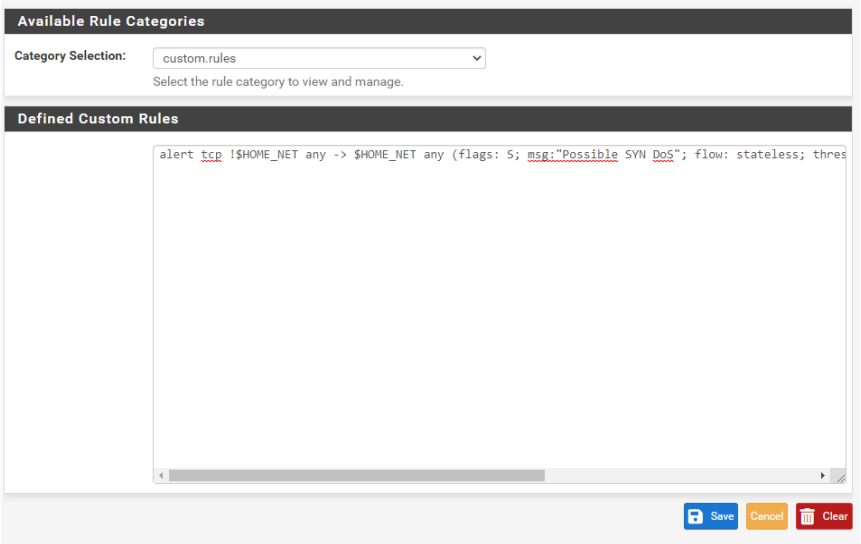
4. Selanjutnya melakukan konfigurasi *costum rule* untuk mengatasi serangan *data flooding* TCP SYN *flood*. *Costum rule* yang digunakan seperti pada Gambar 3.10 yaitu :

   alert tcp !$HOME_NET any -> $HOME_NET any (flags: S; msg:"Possible SYN DoS"; flow: stateless; threshold: type both, track by_dst, count 1000, Seconds 1; sid:10002;rev:1;)

   Berikut merupakan penjelasan dari *costum rule* yang digunakan :

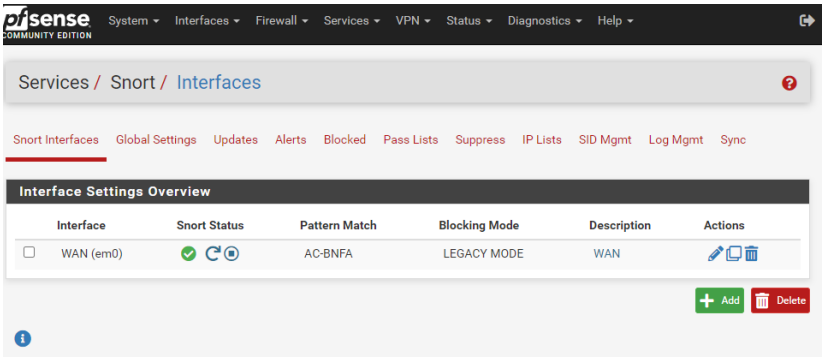
   1. *Alert* adalah *action* dari *rule* yang diterapkan berupa perigatan yang muncul pada menu alerts. Menu *alerts* akan memberikan beberapa informasi yaitu *date, action, priority, protocol, class, source IP, source port, destination IP, destination port,* GID:SID, dan description. Pada IPS mode, *alert* akan menjadi acuan untuk melakukan pemblokiran alamat IP.
   2. TCP merupakan jenis dari protokol yang diamati pada lalu lintas jaringan
   3. Pada *rule* ini sumber paket ditentukan dari *network* yang terdaftar pada *HOME_NET* dengan sembarang identitas *port* dan target paket yang menuju *HOME_NET* di sembarang identitas *port*.
   4. Sintaks msg merupakan pesan yang akan tampil pada *log file* atau *alert* ketika ada serangan yang terdeteksi
   5. Sintaks *flags* berfungsi untuk menentukan jenis paket yang akan diamati yaitu pada penilitian ini didefinisikan S yang artinya paket SYN.
   6. Sintaks *flow: stateless:* Opsi ini memberitahu Snort bahwa aturan ini beroperasi pada paket-paket tanpa status (*stateless*). Ini berarti aturan ini cocok dengan paket-paket individu daripada melacak status koneksi TCP.
   7. Sintaks *sid* mendefinisikan id dari *signature* berdasarkan *rule* yang diterapkan
   8. Sintaks *threshold: type both, track by_dst, count 1000, Seconds 1:* Opsi ini menetapkan ambang batas untuk memicu aturan. Opsi ini

menentukan bahwa ketika aturan cocok dengan 1000 paket dalam waktu 1 detik untuk setiap alamat IP tujuan, maka akan dihasilkan peringatan. Kata kunci by_dst menunjukkan bahwa ambang batas ini dilacak per alamat IP tujuan.



Gambar 3. 10 IPS *Rules*

5. IPS Snort dapat langsung dijalankan dengan menekan tombol *run*, kemudian Snort akan menampilkan ceklist hijau pada Snort status seperti pada Gambar 3.11.

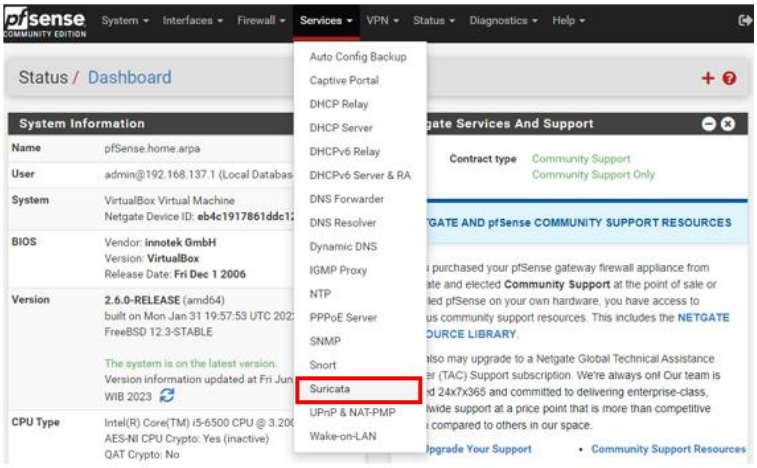


Gambar 3.11 *Run* IPS Snort

3.3.5 KONFIGURASI IPS SURICATA

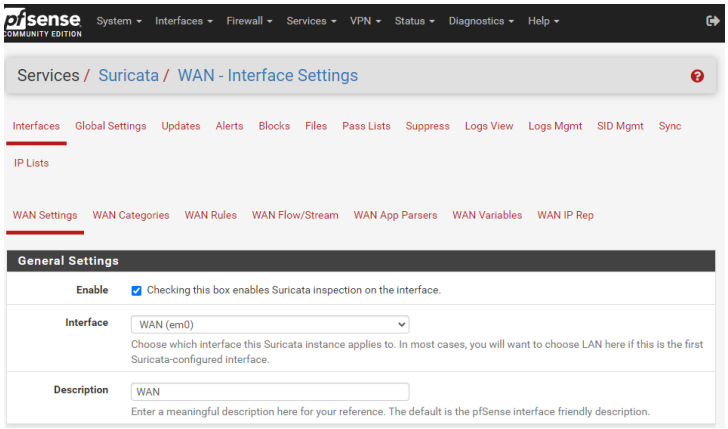
Suricata juga tersedia dalam bentuk paket yang dapat diinstal pada GUI pfSense. Konfigurasi Snort dilakuakan dengan langkah – langkah sebagai berikut :

1. Insatalasi Suricata berada pada menu *system* > *Package manager* > *available packages.* Jika sudah terinstal maka Snort akan muncul pada menu *service* pada GUI pfSense yang ditampilkan pada Gambar 3.12.



Gambar 3.12 *Service* Suricata Pada GUI PfSense

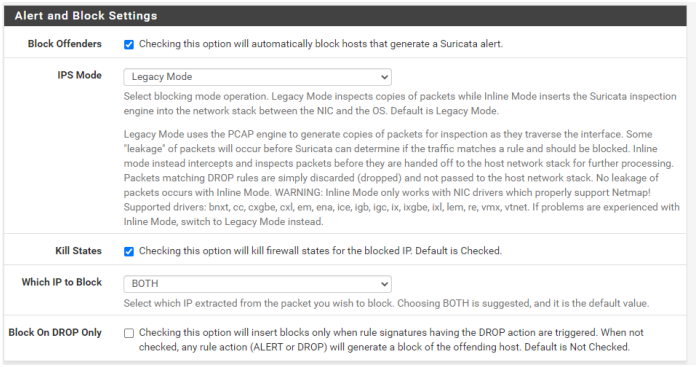
2. Kemudian menambahkan Suricata pada *interface* yang akan menerapkan Suricata sebagai IPS. Pada Gambar 3.13 menunjukkan *interface* yang menerapkan Suricata yaitu WAN *interface*.



Gambar 3.13 Suricata *Interface*

3. Setelah Suricata *Interface* ditetapkan, maka dilakukan konfigurasi pengaturan *blocking* seperti pada Gambar 3.14. Untuk mengaktifkan metode IPS pada Suricata dilakukan dengan cheklist pada menu *Block*

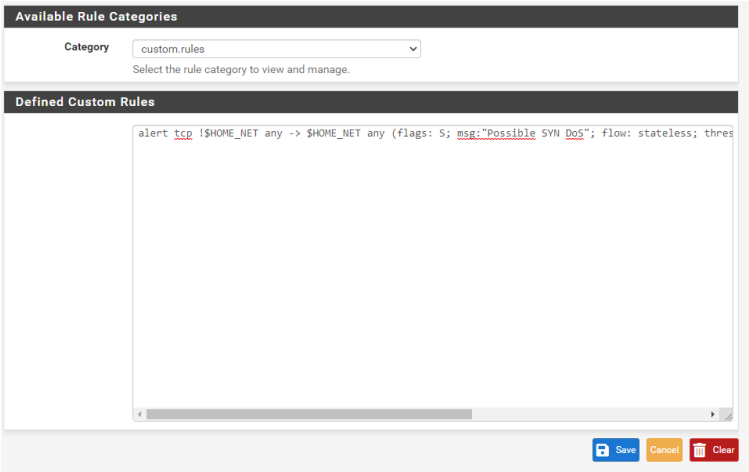
Offenders. IPS Suricata pada penelitian ini menggunakan *Legacy mode* seperti IPS Snort.



Gambar 3.14 *Block Settings*

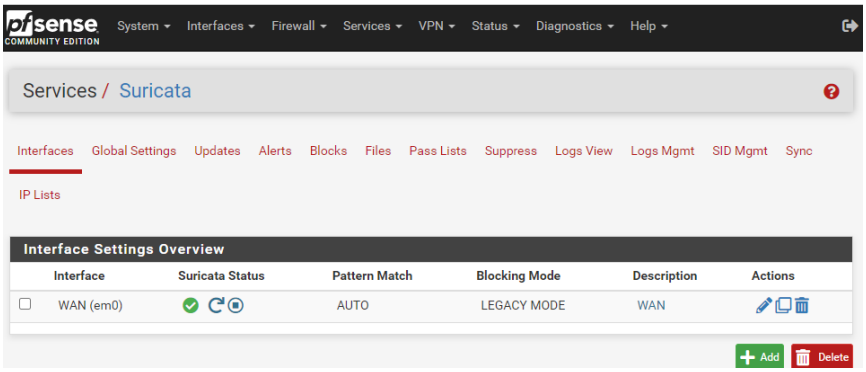
4. Selanjutnya melakukan konfigurasi *costum rule* untuk mengatasi serangan *data flooding* TCP SYN *flood*. *Costum rule* yang digunakan sama seperti yang digunakan pada *costum rule* Snort, yang ditunjukan pada Gambar 3.15 :

   alert tcp !$HOME_NET any -> $HOME_NET any (flags: S; msg:"Possible SYN DoS"; flow: stateless; threshold: type both, track by_dst, count 1000, Seconds 1; sid:10002;rev:1;)



Gambar 3.15 *Costum Rule* Suricata

5. IPS Suricata dapat langsung dijalankan dengan menekan tombol run, kemudian Suricata akan menampilkan ceklist hijau pada Snort status seperti pada Gambar 3.16.



Gambar 3.16 *Run* IPS Suricata

3.3.6 KONFIGURASI NPING

Nping merupakan aplikasi yang digunakan untuk menyerang *web server*. Nping diinstal pada virtual machine kali linux yang berfungsi sebagai *attacker*. Cara kerja Nping pada penelitian ini yaitu dengan mengirimkan permintaan koneksi TCP dan melakukan koneksi penuh/tiga langkah *handshake* TCP sebanyak mungkin pada alamat IP *web server* pada mode TCP *connect*. Pada penelitian melakukan percobaan dengan metode serangan TCP *connect* dengan perintah sebagai berikut :

Nping –tcp-connect -rate=1500 -c 30000 -q 192.168.43.11

Command Nping yang digunakan memiliki beberapa atribut yaitu :

1. *–tcp-connect* merupakan perintah untuk menggunakan mode koneksi TCP tanpa hak is*time*wa. Metode ini menghubungkan ke *port* yang ditentukan pada *host* yang dituju menggunakan protokol TCP.
2. *-rate* merupakan perintah yang menentukan jumlah paket yang akan dikirimkan per detik. Pada penelitian ini jumlah paket yang dikirimkan yaitu 90000 paket per detik.
3. -c merupakan perintah yang menunjukkan jumlah putaran atau ronde probe yang akan dilakukan. Pada penelitian ini menggunakan 900000 putaran.
4. -q merupakan perintah yang digunakan untuk menyederhanakan *output* hasil serangan

3.3.7 SKENARIO PENGUJIAN

Pada penelitian ini dilakukan beberapa pengujian untuk membandingkan kinerja sistem keamanan antara IPS Snort dan IPS Suricata dalam mengatasi serangan *data flooding* terhadap *web server*. *Data flooding* dilakukan menggunakan aplikasi Nping. Jenis *data flooding* yang akan dilakukan pada pengujian yaitu *TCP*

connect flood. Pengujian akan dilakukan dengan 4 skenario agar dapat membandingkan kinerja sistem IPS Snort dan Suricata. Skenario pengujian ditampilkan pada Tabel 3.7.

Tabel 3.7 Skenario Pengujian

Skenario	**Durasi *Capture* paket**	**Jumlah Percobaan**	**Kondisi *web server***	**IPS Snort**	**IPS Suricata**
1	120 detik	30	Tanpa serangan	*OFF*	*OFF*
2	120 detik	30	Menerima serangan	*OFF*	*OFF*
3	120 detik	30	Menerima serangan	*ON*	*OFF*
4	120 detik	30	Menerima serangan	*OFF*	*ON*

Tabel 3.7 menjelaskan skenario pangujian yang akan dilakukan pada penelitian. Pada skenario pertama dimana *web server* belum dilakukan *data flooding* dan tidak menerapkan IPS baik Snort atau Suricata. Kemudian *client* mencoba mengakses layanan *web server*. Setelah itu dilakukan pengukuran Qos layanan *web server*. Skenario ini bertujuan untuk mengetahui Qos dari layanan *web server* ketika belum diserang *data flooding* dan belum menerapkan IPS Snort atau Suricata.

Pada skenario kedua, *web server* akan diserang dengan *data flooding* menggunakan aplikasi Nping tanpa menerapkan IPS Snort atau Suricata. Kemudian *client* mencoba mengakses layanan *web server*. Setelah itu dilakukan pengukuran QoS layanan *web server*. Skenario ini bertujuan untuk mengetahui dampak serangan *data flooding* terhadap Qos dari layanan *web server* yang tidak menerapkan IPS Snort atau Suricata.

Pada skenario ketiga, *web server* yang diserang *data flooding* akan menerapakan IPS Snort untuk mengatasi serangan *data flooding*. Kemudian *client* mencoba mengakses layanan *web server*. Setelah itu dilakukan pengukuran Qos layanan *web server*. Skenario ini bertujuan untuk mengetahui performa IPS Snort dalam mengatasi serangan *data flooding*.

Pada skenario keempat, *web server* akan menonaktifkan *service* dari IPS Snort dan mangaktifkan *service* IPS Suricata. Setelah itu *web server* akan dilakukan serangan *data flooding*. Kemudian *client* akan mencoba mengakses layanan *web server*. Pada saat itu dilakukan pengukuran Qos dari layanan *web server*. Skenario ini bertujuan untuk mengetahui performa IPS Suricata dalam mengatasi serangan *data flooding*.

3.3.8 PENGAMBILAN DATA

Pada peneltian ini terdapat beberapa parameter untuk memperoleh data Qos *web server*. Parameter Qos yang akan diambil pada penelitian ini meliputi data *delay*, paket loss, *throughput*, dan *jitter*. Data – data tersebut diperoleh dari hasil paket *Capture* yang dilakukan di sisi *client* pada 4 skenario pengujian.

3.3.8.1 *DELAY*

Pengambilan data *Delay* dilakukan untuk mengetahui total waktu tunda suatu paket. *Delay* didapatkan dari total selisih waktu data yang dikirim dan diterima pada saat *client* mengakses *web server*. Pad ja penelitian ini *Delay* diambil dari sisi *client* pada 4 skenario pengujian. *Delay* didapatkan dari filter "tcp&& ip.addr == 192.168.43.11 && ip.addr == 192.168.43.46 && tcp.*time*_delta > 0 and tcp.flags.fin==0 && tcp.flags.reset==0 " yang kemudian diex*port* dalam bentuk csv pada *mikrosoft* excel. Pengujian dilakukan pada kurun waktu 120 detik dengan 30 kali percobaan pada setiap skenario. Untuk mendapatkan nilai rata – rata *Delay* menggunakan persamaan :

$$Delay\ (s) = \frac{Jumlah\ Delay}{Jumlah\ Paket\ yang\ Diterima}$$

3.3.8.2 *PACKET LOSS*

Pengambilan data *Packet Loss* dilakukan untuk mengetahui jumlah total paket pengiriman data yang tidak sampai dari sisi *client* ketika mengakses *web server*. Pada penelitian ini *Packet Loss* diambil dari sisi *client* pada 4 skenario pengujian dengan waktu 120 detik. Persentase *Packet Loss* didapatkan dari persamaan berikut :

$$Packet\ Loss = \frac{jumlah\ paket\ retransmission}{jumlah\ paket\ data\ dikirim} \times 100$$

Packet *retransmission* merupakan paket data yang tidak sampai sehingga ditransmisikan ulang pada proses komunikasi antara *client* dan *server*. Packet yang ditransmisikan ulang didapatkan melalui filter "tcp && ip.dst == 192.168.43.11 && ip.src== 192.168.43.46 &&tcp.analysis.retransmission". Sedangkan paket data yang dikirim didapatkan dari filter "tcp && ip.dst == 192.168.43.11 && ip.src== 192.168.43.46".

3.3.8.3 *THROUGHPUT*

Pengambilan data *Throughput* dilakukan untuk mengetahui laju data pada pengiriman data. Pada penelitian ini *Throughput* diambil dari sisi *client* pada 4 skenario pengujian. *Throughput* didapatkan dari filter "tcp&& ip.addr == 192.168.43.11 && ip.addr == 192.168.43.46", yang kemudian mendapatakan nilai *Throughput* dari *Capture file* properties. Pengujian dilakukan pada kurun waktu 60 detik dengan 30 kali percobaan pada setiap skenario. Untuk mendapatkan nilai *Throughput* menggunakan persamaan :

$$Throughput = \frac{Paket\ data\ diterima}{Lama\ Pengamatan}$$

3.3.8.4 *ROUND TRIP TIME (RTT)*

Pengambilan data RTT dilakukan untuk mengetahui waktu yang dibutuhkan untuk mengirim paket SYN sampai paket ACK diterima untuk paket tersebut. Nilai RTT berfungsi untuk memperkirakan jumlah waktu yang dibutuhkan ketika *client* meminta *file* base HTML sampai seluruh *file* diterima oleh *client*. Hal ini bertujuan untuk menganalisis kestabilan dalam proses koneksi dan transfer data antara *web server* dan *client*. Ketika nilai RTT lebih kecil maka terjadi peningkatan *throughput* yang tinggi.

Perhitungan data RTT dilakukan dengan menjumlahkan waktu yang digunakan pada proses *three-way handsake* antara *client* dan *server*. Data yang dihitung merupakan hasil capture lalu lintas jaringan dari setiap percobaan. Kemudian dari data capture lalu lintas jaringan dihitung waktu yang digunakan pada proses *three-way handshake* menggunakan aplikasi Tshark. Aplikasi Tshark

melakukan *convert* data *pcapng* yang merupakan hasil *Capture* Wireshark. *File pcapng* dikonversi menggunkan *tools* Tshark dengan perintah ".\tshark.exe -q -r N_30.pcapng -z conv,tcp > N_30.csv" yang memiliki fungsi sebagai berikut :

1. Perintah .\tshark.exe berfungsi untuk menjalankan *tools* Tshark.
2. Perintah -q berfungsi untuk menampilkan tampilan yang minimal.
3. Perintah -r N_30.pcapng berfungsi untuk menunjukkan *file* yang akan dianalisis yaitu N_30.pcapng.
4. Perintah -z conv, tcp berfungsi untuk menampilkan data RTT pada setiap TCP *stream*.
5. Perintah > N_30.csv berfungsi untuk melakukan konversi tampilan output Tshark ke *file* .csv.