

**BAB III**  
**METODE PENELITIAN**

**3.1 Alat dan Bahan Penelitian**

Untuk membantu simulasi penelitian ini dibutuhkan beberapa perangkat yang digunakan, terdiri atas perangkat keras (*hardware*) dan perangkat lunak (*software*)

1) Perangkat keras (*hardware*)

Berikut beberapa spesifikasi perangkat keras yaitu :

**Tabel 3.1 Spesifikasi Perangkat Keras**

NO	Perangkat	Spesifikasi	Keterangan
1	<i>Virtual Machine</i>	Prosesor Intel Core i7-11800H (2 CPUs), RAM 2 GB, <i>Hard disk</i> 80 Gb	Sebagai <i>server</i>
2	<i>Virtual Machine</i>	Prosesor Intel Core i7-11800H (4 CPUs), RAM 4 GB, <i>Hard disk</i> 80 Gb	Sebagai jembatan
3	<i>Virtual Machine</i>	Prosesor Intel Core i7-11800H (2 CPUs), RAM 2 GB, <i>Hard disk</i> 80 Gb	Sebagai penyerang

2) Perangkat lunak (*Software*)

a) *Server*

*Server* menggunakan sistem operasi Windows yang sudah terpasang dan memberikan layanan berupa *FTP Server* dan *Web Server*.

b) Jembatan

Jembatan menggunakan sistem operasi Ubuntu yang akan dipasang Snort, Wireshark, dan Ethtool.

c) Penyerang

Penyerang menggunakan sistem operasi Kali Linux yang dilengkapi dengan aplikasi menyerang seperti Hping, CUPP, dan Hydra.

### 3.2 Alur Penelitian

Penelitian ini dilakukan dengan beberapa tahapan untuk mencapai tujuan penelitian yang sudah ditentukan. Alur penelitian yang dilakukan peneliti pada penyusunan laporan ini ditunjukkan pada Gambar 3.1 Alur Penelitian.



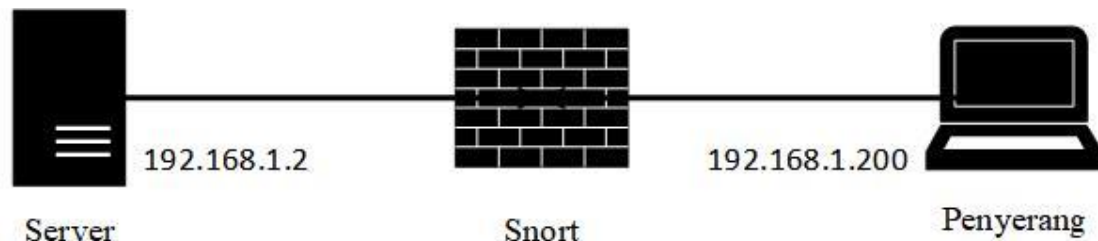
Gambar 3.1 Alur Penelitian

#### 3.2.1 Studi Literatur dan Observasi

Studi literatur penelitian ini dilakukan dengan mencari sumber teori yang berhubungan dengan penelitian yang akan dilakukan. Sumber yang digunakan berupa jurnal penelitian, buku dan *website* yang membahas sistem kerja IPS, konfigurasi Snort, jenis serangan. Selain bersumber dari buku, jurnal penelitian dan *website* dilakukan juga observasi. Tujuan observasi ini adalah untuk memperoleh jawaban tentang parameter-parameter yang penting dalam kestabilan sebuah *server* akibat mengalami penurunan kerja pada suatu *server* dari pemakaian sumber daya yang tidak wajar.

### 3.2.2 Analisis kebutuhan

Analisis kebutuhan penelitian ini, dilakukan menentukan kebutuhan yang akan membantu dalam penelitian ini. Kebutuhan yang perlu disiapkan antara lain persiapan perangkat keras dan perangkat lunak. Pada perangkat keras yaitu menyediakan perangkat hingga membuat perancangan jaringan untuk membentuk jaringan lokal. Sedangkan perangkat lunak meliputi menyediakan aplikasi yang akan dibutuhkan yaitu Snort serta aplikasi untuk melakukan penyerangan. Perancangan jaringan digunakan untuk mempermudah dalam menjelaskan perangkat yang terdapat pada jaringan berupa topologi jaringan. Topologi jaringan yang digunakan dapat dilihat pada Gambar 3.2 Topologi Jaringan.



**Gambar 3.2 Topologi Jaringan**

Pada Gambar 3.2 Topologi Jaringan merupakan bentuk topologi jaringan yang digunakan pada penelitian ini. *Server* menggunakan IP 192.168.1.2 kemudian penyerang menggunakan IP 192.168.1.200. Snort pada topologi ini akan berfungsi sebagai jembatan yang menghubungkan antara *server* dan penyerang. Penggunaan Snort sebagai jembatan bertujuan untuk mengelola data menuju *server* sehingga penyerangan yang dilakukan dapat digagalkan.

### 3.2.3 Instalasi dan Konfigurasi

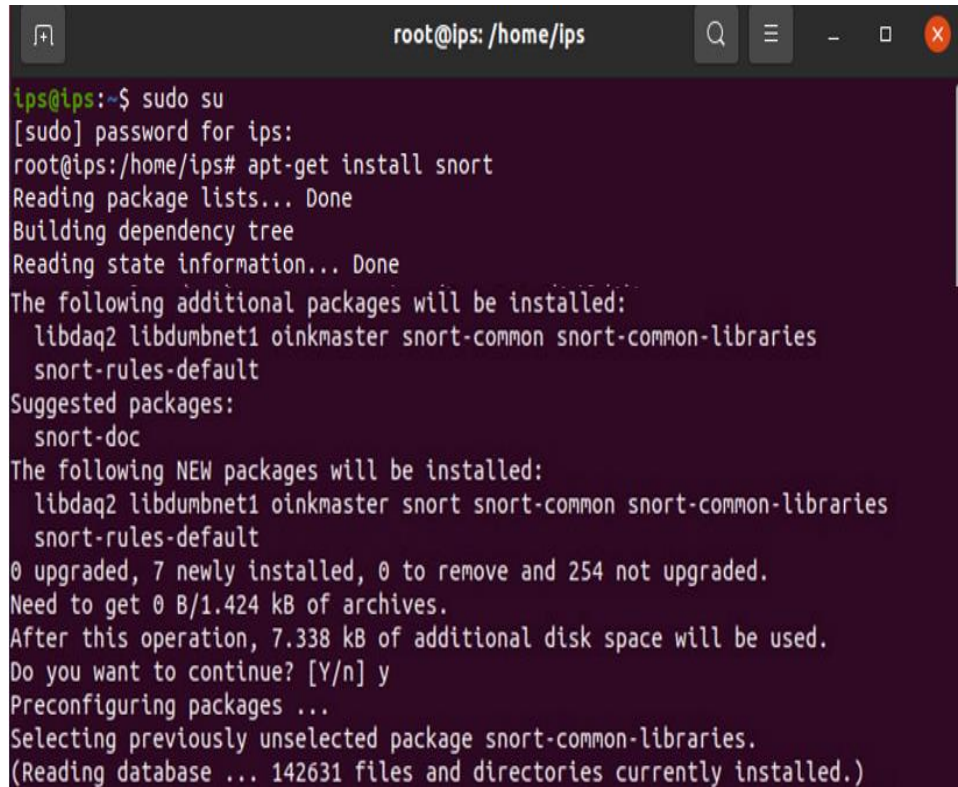
Pada tahap ini instalasi yang dilakukan dimulai dari pemasangan dan konfigurasi aplikasi Snort serta pemasangan aplikasi pendukung.

#### a) Konfigurasi IPS

Perangkat yang akan dipasang Snort telah dipersiapkan dan akan dikonfigurasi sebelum melakukan tahap pengujian. Berikut tahapan dari konfigurasi IPS:

## 1. Mengunduh dan memasang Snort

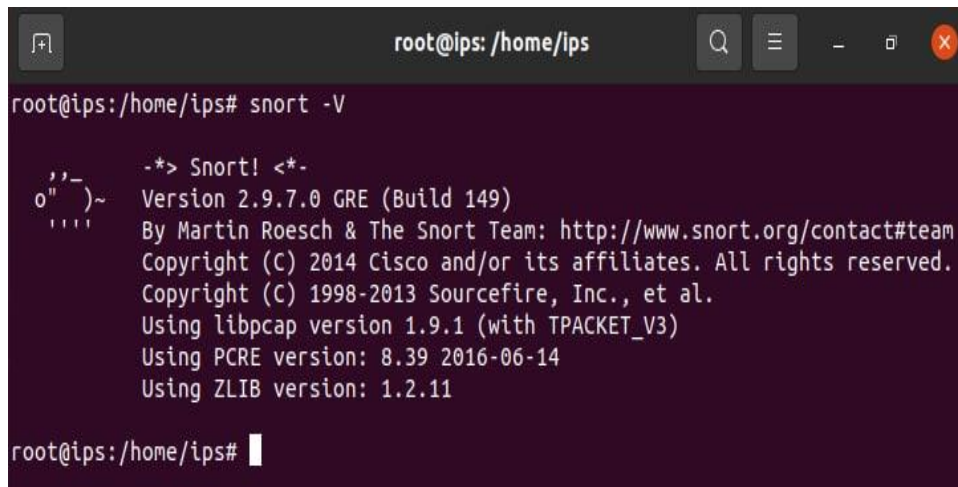
Sebelum dilakukan pemasangan Snort, komputer wajib terhubung dengan internet. Selain terhubung dengan internet, dalam pemasangan harus masuk menggunakan *superuser* yaitu *root* dengan perintah “*sudo su*” seperti pada Gambar 3.3 Pemasangan Snort.

A terminal window with a dark purple background and white text. The window title is 'root@ips: /home/ips'. The user 'ips@ips' has entered 'sudo su' and is now acting as root. The command 'apt-get install snort' has been executed. The terminal output shows the process of reading package lists, building a dependency tree, and listing additional packages to be installed: libdaq2, libdumbnet1, oinkmaster, snort-common, snort-common-libraries, and snort-rules-default. It also lists suggested packages like snort-doc. The user has responded 'y' to the prompt 'Do you want to continue? [Y/n]'. The terminal shows the packages are being preconfigured and installed. The final output indicates that 7 new packages were installed and 254 were not upgraded, with a total disk space requirement of 7.338 kB.

```
root@ips: /home/ips
ips@ips:~$ sudo su
[sudo] password for ips:
root@ips:/home/ips# apt-get install snort
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libdaq2 libdumbnet1 oinkmaster snort-common snort-common-libraries
  snort-rules-default
Suggested packages:
  snort-doc
The following NEW packages will be installed:
  libdaq2 libdumbnet1 oinkmaster snort snort-common snort-common-libraries
  snort-rules-default
0 upgraded, 7 newly installed, 0 to remove and 254 not upgraded.
Need to get 0 B/1.424 kB of archives.
After this operation, 7.338 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Preconfiguring packages ...
Selecting previously unselected package snort-common-libraries.
(Reading database ... 142631 files and directories currently installed.)
```

**Gambar 3.3 Pemasangan Snort**

Pada Gambar 3.3 Pemasangan Snort adalah awal proses konfigurasi IPS. Pemasangan Snort dapat dilakukan menggunakan perintah “*apt-get install snort*”. Pada proses pemasangan harus menjawab “y” pada pertanyaan “*Do you want to continue?*”. Pemasangan Snort berisi Snort dan termasuk aturan-aturan yang tersedia di *website* resmi Snort yaitu <https://www.snort.org>. Setelah terpasang dapat dilakukan uji coba untuk menjalankan Snort dengan perintah “*snort -V*” untuk mengecek versi dari Snort yang telah dipasang, seperti pada Gambar 3.4 Tampilan Versi Snort.



```
root@ips:/home/ips# snort -V

  ,,_-_*> Snort! <*-
o"  )~  Version 2.9.7.0 GRE (Build 149)
  '    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
        Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
        Copyright (C) 1998-2013 Sourcefire, Inc., et al.
        Using libpcap version 1.9.1 (with TPACKET_V3)
        Using PCRE version: 8.39 2016-06-14
        Using ZLIB version: 1.2.11

root@ips:/home/ips#
```

**Gambar 3.4 Tampilan Versi Snort**

Pada Gambar 3.4 Tampilan Versi Snort menunjukkan versi yang telah dipasang, pencipta serta hak cipta dari aplikasi Snort. Selain itu menjelaskan tentang versi dari libpcap, PCRE, dan ZLIB. libpcap merupakan seperangkat fungsi atau *library* yang berfungsi untuk menangkap semua paket data yang melewati *interface*. *Perl Compatible Regular Expressions* (PCRE) merupakan seperangkat fungsi yang digunakan dalam penulisan aturan sehingga dapat menggunakan bahasa pemrograman perl. *Z-Library* (ZLIB) adalah *library* yang berfungsi untuk kompresi data dan dekompresi data.

## 2. Konfigurasi *file* Snort.conf

Sebelum melakukan konfigurasi pada *file Snort.conf*, dapat dilakukan pengecekan ketersediaan *Data Acquisition library* (DAQ). DAQ sendiri digunakan untuk menggantikan fungsi libpcap sehingga operasi pada berbagai *interface* perangkat keras dan perangkat lunak tidak memerlukan perubahan pada Snort. Hal ini memungkinkan Snort dapat berjalan dalam tipe *inline* dan sebagainya. Untuk mengecek ketersediaan DAQ dapat menggunakan perintah “*snort --daq-list*” seperti pada Gambar 3.5 Ketersediaan DAQ.

```

root@ips: /home/ips
root@ips:/home/ips# snort --daq-list
Available DAQ modules:
pcap(v3): readback live multi unpriv
ipfw(v3): live inline multi unpriv
dump(v2): readback live inline multi unpriv
afpacket(v5): live inline multi unpriv
root@ips:/home/ips#

```

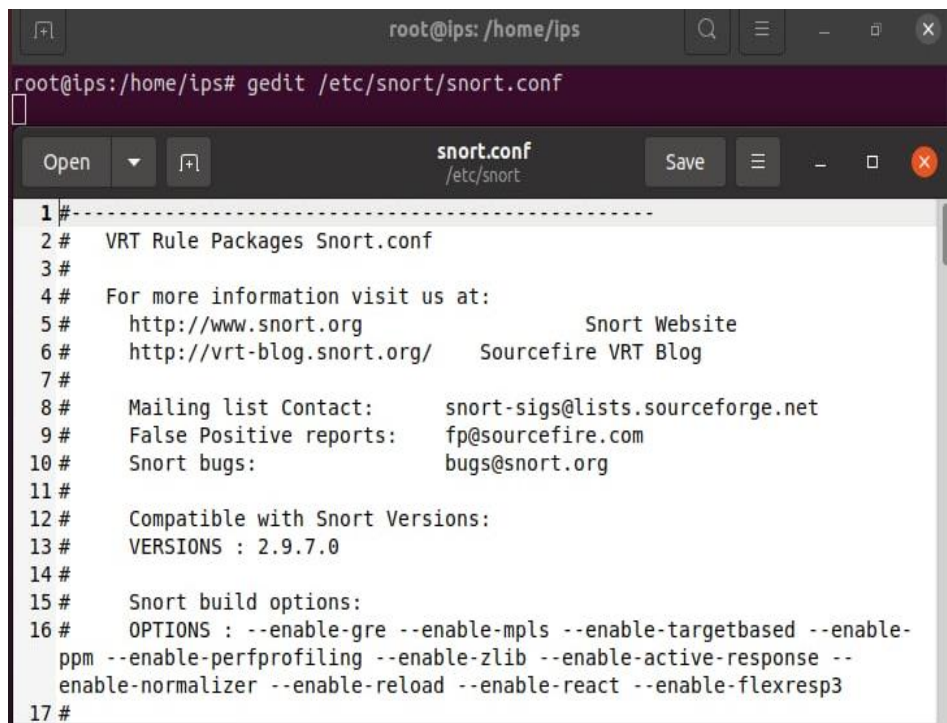
**Gambar 3.5 Ketersediaan DAQ**

Berdasarkan Gambar 3.5 Ketersediaan DAQ terdapat beberapa modul DAQ yang dapat digunakan, modul-modul tersebut diperoleh saat memasang aplikasi Snort bawaan. Penjelasan dari modul yang tersedia dapat dilihat pada Tabel 3.2 Modul DAQ.

**Tabel 3.2 Modul DAQ**

No	Modul	Fungsi
1	<i>pcap</i>	Digunakan untuk menangkap data yang melintas
2	<i>ipfw</i>	Digunakan untuk meneruskan data yang melintas
3	<i>dump</i>	Digunakan untuk membuang semua data yang melintas
4	<i>afpacket</i>	Digunakan untuk menyaring data yang melintas sehingga data membuang dan meneruskan data

Setelah memastikan ketersediaan DAQ, dilakukan konfigurasi pada *file snort.conf*. *File snort.conf* merupakan *file* yang berisi konfigurasi utama Snort. *File* ini dapat diakses pada *folder* “*/etc/snort/*”. Untuk merubah isi *file* tersebut harus masuk memakai *superuser* dan juga menggunakan perintah “*gedit /etc/Snort/Snort.conf*” selain *gedit* dapat memakai *nano*, seperti pada Gambar 3.6 Tampilan *File Snort.conf*.



```
1 #-----
2 # VRT Rule Packages Snort.conf
3 #
4 # For more information visit us at:
5 # http://www.snort.org           Snort Website
6 # http://vrt-blog.snort.org/     Sourcefire VRT Blog
7 #
8 # Mailing list Contact:         snort-sigs@lists.sourceforge.net
9 # False Positive reports:      fp@sourcefire.com
10 # Snort bugs:                  bugs@snort.org
11 #
12 # Compatible with Snort Versions:
13 # VERSIONS : 2.9.7.0
14 #
15 # Snort build options:
16 # OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-
ppm --enable-perfprofiling --enable-zlib --enable-active-response --
enable-normalizer --enable-reload --enable-react --enable-flexresp3
17 #
```

**Gambar 3.6 Tampilan File Snort.conf**

Berdasarkan Gambar 3.6 Tampilan *File Snort.conf*, menunjukkan cara mengakses dan tampilan sekilas dari *file Snort.conf*. Terdapat beberapa hal yang perlu diatur supaya aplikasi Snort dapat berjalan sesuai dengan yang ingin digunakan.

```
//Setting network yang akan dimonitor
ipvar HOME_NET 192.168.2.2/24

//Deklarasikan network dari EXTERNAL_NET
ipvar EXTERNAL_NET !$HOME_NET

//atur tipe dan mode untuk menjalankan mode inline
#config daq: <tipe>
#config daq_dir: <dir>
#config daq_mode: <mode>
#config daq_var: <var>

//Matikan semua rule yang tersedia kecuali local.rules
dengan menambahkan #
include $RULE_PATH/local.rules
include $RULE_PATH/app-detect.rules
....
include $RULE_PATH/web-php.rules
include $RULE_PATH/x11.rules
```



Berdasarkan pengaturan diatas, menunjukkan “*HOME\_NET*” digunakan untuk mendefinisikan alamat *host* yang akan diawasi. Kemudian pengaturan “*EXTERNAL\_NET*” digunakan untuk mendefinisikan alamat *network* yang dianggap sebagai serangan. Dilakukan juga pengaturan DAQ dengan cara mengubah *<tipe>* menjadi *afpacket* dan *<mode>* menjadi *inline*. Tipe DAQ dapat dilihat pada Tabel 4.1 Modul DAQ selain itu terdiri dari beberapa mode yaitu mode *read-file* berarti mode yang hanya dapat membaca data saja, mode *passive* berarti yang bertindak sebagai IDS dimana dapat membaca data dan memberikan peringatan, mode *inline* berarti mode yang dapat membaca data, memberikan peringatan, serta dapat menjatuhkan(*drop*) data.

### 3. Melakukan konfigurasi *rules*

Pada konfigurasi *rules*, dalam pemasangan Snort terdapat beberapa *file rules* yang telah disediakan. *Files* tersebut tersimpan pada folder “*/etc/snort/rules/*” seperti pada Gambar 3.7 Tampilan Folder *Rules*.

```

root@ips: /etc/snort/rules
root@ips:/etc/snort/rules# ls
attack-responses.rules      community-web-dos.rules      policy.rules
backdoor.rules              community-web-iis.rules      pop2.rules
bad-traffic.rules           community-web-misc.rules     pop3.rules
chat.rules                  community-web-php.rules      porn.rules
community-bot.rules         ddos.rules                   rpc.rules
community-deleted.rules     deleted.rules                rservices.rules
community-dos.rules         dns.rules                    scan.rules
community-exploit.rules     dos.rules                    shellcode.rules
community-ftp.rules         experimental.rules           smtp.rules
community-game.rules        exploit.rules                 snmp.rules
community-icmp.rules        finger.rules                  sql.rules
community-imap.rules        ftp.rules                     telnet.rules
community-inappropriate.rules icmp-info.rules              tftp.rules
community-mail-client.rules icmp.rules                    virus.rules
community-misc.rules        imap.rules                    web-attacks.rules
community-nntp.rules        info.rules                    web-cgi.rules
community-oracle.rules      local.rules                   web-client.rules
community-policy.rules      misc.rules                    web-coldfusion.rules
community-sip.rules         multimedia.rules              web-frontpage.rules
community-smtp.rules        mysql.rules                   web-iis.rules
community-sql-injection.rules netbios.rules                 web-misc.rules
community-virus.rules       nntp.rules                    web-php.rules
community-web-attacks.rules oracle.rules                   x11.rules

```

**Gambar 3.7 Tampilan Folder *Rules***



Berdasarkan pada Gambar 3.7 Tampilan Folder *Rules*, *rules* dipisah berdasarkan klasifikasinya namun semua *rules* tersebut dapat disimpan dalam satu *files* yaitu *local.conf*. Berikut beberapa *rule* yang diimplementasikan pada penelitian ini dimasukkan ke dalam *local.rules*.

```

1 # $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
2 # LOCAL RULES
3 # -----
4 # This file intentionally does not come with signatures.  Put your local additions here.
5 |
6 #ICMPRULE
7 alert icmp any any -> 192.168.2.2 any ( msg:"Ping icmp normal"; dsize:<100; classtype:icmp-event; sid:-
1000001; rev:4; )
8 #drop icmp any any -> 192.168.2.2 any ( msg:"Ping of Death Terdeteksi"; dsize:>100;
 classtype:attempted-dos; sid:1000002; rev:3; )
9 drop icmp any any -> 192.168.2.2 any ( msg:"ICMP Flood dan Ping of Death Terdeteksi"; dsize:>100;
 classtype:attempted-dos; detection_filter:track by_src, count 50, seconds 1; sid:1000003; rev:3; )
10
11 #UDPRULE
12 alert udp any any -> 192.168.2.2 any ( msg:"UDP normal"; classtype:misc-activity; sid:1000004; rev:-
1; )
13 drop udp any any -> 192.168.2.2 any ( msg:"UDP Flood Terdeteksi"; classtype:attempted-dos;
 detection_filter:track by_src, count 300, seconds 1; sid:1000005; rev:2; )
14
15 #SYNRULE
16 alert tcp any any -> 192.168.2.2 any ( msg:"SYN normal"; flags:A; classtype:tcp-connection; sid:-
1000006; rev:1; )
17 drop tcp any any -> 192.168.2.2 any ( msg:"SYN Flood Terdeteksi"; flags:S; classtype:attempted-dos;
 detection_filter:track by_src, count 1000, seconds 1; sid:10000011; rev:1; )
18 drop tcp any any -> 192.168.2.2 any ( msg:"SYN Flood Terdeteksi"; flags:ASR; classtype:attempted-dos;
 detection_filter:track by_src, count 1000, seconds 1; sid:1000007; rev:2; )
19
20 #FTPRULE
21 alert tcp 192.168.2.2 21 -> any any ( msg:"FTP sudah masuk"; content:"Logged on"; classtype:tcp-
connection; sid:1000008; rev:1; )
22 alert tcp 192.168.2.2 21 -> any any ( msg:"FTP gagal masuk"; content:"Login or password incorrect!";
 classtype:tcp-connection; sid:1000009; rev:1; )
23 drop tcp any any -> 192.168.2.2 21 ( msg:"FTP Bruteforce Terdeteksi"; content:"USER";
 classtype:suspicious-login; detection_filter:track by_src, count 10, seconds 1; sid:1000010; rev:2; )
24 drop tcp any any -> 192.168.2.2 21 ( msg:"FTP Bruteforce Terdeteksi"; content:"PASS";
 classtype:suspicious-login; detection_filter:track by_src, count 10, seconds 1; sid:1000012; rev:2; )

```

**Gambar 3.8 File *local.rules***

Merujuk pada Gambar 3.8 *File local.rules* berisi mengenai aturan yang telah dibuat. Pada baris 1 menunjukkan bahwa file tersebut adalah *local rule* dengan versi 1.11 dibuat pada tahun 2004. Baris 2 adalah judul file dengan nama *LOCAL RULE*. Baris 4 adalah keterangan dari aturan yang akan dibuat tidak memiliki *signatures* resmi. Berikut penjelasan dari tiap aturan yang telah dibuat:

```

6 #ICMPRULE
7 alert icmp any any -> 192.168.2.2 any ( msg:"Ping icmp normal"; dsize:<100; classtype:icmp-event; sid:-
1000001; rev:4; )
8 #drop icmp any any -> 192.168.2.2 any ( msg:"Ping of Death Terdeteksi"; dsize:>100;
 classtype:attempted-dos; sid:1000002; rev:3; )
9 drop icmp any any -> 192.168.2.2 any ( msg:"ICMP Flood dan Ping of Death Terdeteksi"; dsize:>100;
 classtype:attempted-dos; detection_filter:track by_src, count 50, seconds 1; sid:1000003; rev:3; )
10

```

**Gambar 3.9 Aturan ICMP**

Merujuk pada Gambar 3.9 Aturan ICMP ada pada baris 7 dan 9. Pada baris 7 bertujuan untuk memberikan peringatan berupa pesan “*Ping icmp normal*” pada paket protokol icmp yang menuju *server* (192.168.2.2) dari alamat IP manapun dengan kondisi batas ukuran paket kurang dari 100 *byte*. Aturan ini bertipe *icmp event* yang berarti acara umum ICMP dengan nomor aturan 1000001 dan sudah direvisi sebanyak 4 kali. Baris 8 terdapat tanda “#” yang menunjukkan bahwa aturan tersebut tidak terpakai. Pada baris 9 bertujuan untuk menjatuhkan paket dan memberikan peringatan berupa pesan “*ICMP Flood dan Ping of Death Terdeteksi*” pada paket protokol icmp yang menuju *server* (192.168.2.2) dari alamat IP manapun dengan kondisi jika ada 50 paket dalam 1 detik dengan deteksi berdasarkan sumber paket dan memiliki ukuran paket lebih dari 100 *byte*, maka aturan akan *me-drop* paket dengan kondisi sama pada paket yang akan datang. Aturan ini bertipe *attempted-dos* yang berarti percobaan DoS, nomor aturan 1000003 dan sudah direvisi sebanyak 3 kali.

```
11 #UDPRULE
12 alert udp any any -> 192.168.2.2 any ( msg:"UDP normal"; classtype:misc-activity; sid:1000004; rev:1; )
13 drop udp any any -> 192.168.2.2 any ( msg:"UDP Flood Terdeteksi"; classtype:attempted-dos; detection_filter:track by_src, count 300, seconds 1; sid:1000005; rev:2; )
```

### Gambar 3.10 Aturan UDP

Merujuk pada Gambar 3.10 Aturan UDP ada pada baris 12 dan 13. Pada baris 12 bertujuan untuk memberikan peringatan berupa pesan “*UDP normal*” pada paket protokol udp yang menuju *server* (192.168.2.2) dari alamat IP manapun. Aturan ini bertipe *misc-activity* yang berarti aktivitas lain-lain dengan nomor aturan 1000004 dan sudah direvisi sebanyak 1 kali. Pada baris 13 bertujuan untuk menjatuhkan paket dan memberikan peringatan berupa pesan “*UDP Flood Terdeteksi*” pada paket protokol udp yang menuju *server* (192.168.2.2) dari alamat IP manapun dengan kondisi jika ada 300 paket dalam 1 detik dengan deteksi berdasarkan sumber paket, maka aturan akan *me-drop* paket dengan kondisi sama pada paket yang akan datang. Aturan ini bertipe *attempted-dos* yang berarti percobaan DoS, nomor aturan 1000005 dan sudah direvisi sebanyak 2 kali.

```

15 #SYNRULE
16 alert tcp any any -> 192.168.2.2 any ( msg:"SYN normal"; flags:A; classtype:tcp-connection; sid:-
  1000006; rev:1; )
17 drop tcp any any -> 192.168.2.2 any ( msg:"SYN Flood Terdeteksi"; flags:S; classtype:attempted-dos;
  detection_filter:track by_src, count 1000, seconds 1; sid:10000011; rev:1; )
18 drop tcp any any -> 192.168.2.2 any ( msg:"SYN Flood Terdeteksi"; flags:ASR; classtype:attempted-dos;
  detection_filter:track by_src, count 1000, seconds 1; sid:1000007; rev:2; )

```

### Gambar 3.11 Aturan SYN

Merujuk pada Gambar 3.11 Aturan SYN ada pada baris 16 hingga 18. Pada baris 16 bertujuan untuk memberikan peringatan berupa pesan “SYN *normal*” pada paket protokol tcp yang menuju *server* (192.168.2.2) dari alamat IP manapun dengan kondisi paket *TCPflags* berjenis ACK (A) atau menunjukkan konfirmasi dari sesi koneksi dan siap mengirim permintaan dan data. Aturan ini bertipe *tcp-connection* yang berarti koneksi TCP terdeteksi dengan nomor aturan 1000006 dan sudah direvisi sebanyak 1 kali. Pada baris 17 bertujuan untuk menjatuhkan paket dan memberikan peringatan berupa pesan “SYN Flood Terdeteksi” pada paket protokol tcp yang menuju *server* (192.168.2.2) dari alamat IP manapun dengan kondisi jika ada 1000 paket *TCPflags* berjenis SYN (S) atau menunjukkan TCP akan membuat sesi koneksi dalam 1 detik dengan jenis deteksi berdasarkan sumber paket, maka aturan akan *me-drop* paket dengan kondisi sama pada paket yang akan datang. Aturan ini bertipe *attempted-dos* yang berarti percobaan DoS, nomor aturan 10000011 dan sudah direvisi sebanyak 1 kali. Pada baris 18 sama seperti pada baris 17 namun tujuannya untuk menjatuhkan paket dan memberikan peringatan berupa pesan “SYN Flood Terdeteksi” pada paket protokol tcp yang menuju *server* (192.168.2.2) dari alamat IP manapun dengan kondisi jika ada 1000 paket *TCPflags* berjenis SYN (S), ACK (A), dan RST (R) dalam 1 detik dengan jenis deteksi berdasarkan sumber paket, maka aturan akan *me-drop* paket dengan kondisi sama pada paket yang akan datang. Aturan ini bertipe *attempted-dos* yang berarti percobaan DoS, nomor aturan 1000007 dan sudah direvisi sebanyak 2 kali.

```

20 #FTPRULE
21 alert tcp 192.168.2.2 21 -> any any ( msg:"FTP sudah masuk"; content:"Logged on"; classtype:tcp-
connection; sid:1000008; rev:1; )
22 alert tcp 192.168.2.2 21 -> any any ( msg:"FTP gagal masuk"; content:"Login or password incorrect!";
classtype:tcp-connection; sid:1000009; rev:1; )
23 drop tcp any any -> 192.168.2.2 21 ( msg:"FTP Bruteforce Terdeteksi"; content:"USER";
classtype:suspicious-login; detection_filter:track by_src, count 10, seconds 1; sid:1000010; rev:2; )
24 drop tcp any any -> 192.168.2.2 21 ( msg:"FTP Bruteforce Terdeteksi"; content:"PASS";
classtype:suspicious-login; detection_filter:track by_src, count 10, seconds 1; sid:1000012; rev:2; )

```

### Gambar 3.12 Aturan FTP

Merujuk pada Gambar 3.12 Aturan FTP ada pada baris 21 hingga 24. Pada baris 21 bertujuan untuk memberikan peringatan berupa pesan “*FTP sudah masuk*” pada paket protokol tcp yang bersumber dari *server* (192.168.2.2) menuju alamat IP manapun dengan kondisi paket berisi konten “*Logged on*”. Aturan ini bertipe *tcp-connection* yang berarti koneksi TCP terdeteksi dengan nomor aturan 1000008 dan sudah direvisi sebanyak 1 kali. Pada baris 22 bertujuan untuk memberikan peringatan berupa pesan “*FTP gagal masuk*” pada paket protokol tcp yang bersumber dari *server* (192.168.2.2) menuju alamat IP manapun dengan kondisi paket berisi konten “*Login or password incorrect!*”. Aturan ini bertipe *tcp-connection* yang berarti koneksi TCP terdeteksi dengan nomor aturan 1000009 dan sudah direvisi sebanyak 1 kali. Pada baris 23 bertujuan untuk menjatuhkan paket dan memberikan peringatan berupa pesan “*FTP Bruteforce Terdeteksi*” pada paket protokol tcp yang menuju *server* (192.168.2.2) dari alamat IP manapun, dengan kondisi jika ada 10 paket berisi konten “*USER*” dalam 1 detik dengan jenis deteksi berdasarkan sumber paket, maka aturan akan me-*drop* paket dengan kondisi sama pada paket yang akan datang. Aturan ini bertipe *suspicious-login* yang berarti adanya upaya login dengan menggunakan *user* yang mencurigakan, nomor aturan 1000010 dan sudah direvisi sebanyak 2 kali. Pada baris 24 bertujuan untuk menjatuhkan paket dan memberikan peringatan berupa pesan “*FTP Bruteforce Terdeteksi*” pada paket protokol tcp yang menuju *server* (192.168.2.2) dari alamat IP manapun, dengan kondisi jika ada 10 paket berisi konten “*PASS*” dalam 1 detik dengan jenis deteksi berdasarkan sumber paket, maka aturan akan me-*drop* paket dengan kondisi sama pada paket yang akan datang. Aturan ini

bertipe *suspicious-login* yang berarti adanya upaya login dengan menggunakan *pass* yang mencurigakan, nomor aturan 1000012 dan sudah direvisi sebanyak 2 kali.

Dalam pembuatan aturan diatas memiliki urutan yaitu

- a. Tindakan aturan = untuk tindakan aturan yang dibuat seperti *alert* (menghasilkan peringatan), *drop* (blokir dan catat paket), *log* (mencatat paket ke dalam *log* tanpa peringatan), *pass* (meloloskan paket), dan *reject* (menolak paket dan mengirim TCP reset atau pesan tidak dapat dijangkau).
- b. Protokol = jenis protokol seperti TCP, UDP, ICMP, dan IP.
- c. Alamat sumber = mendefinisikan alamat IP sumber paket, jika *any* maka semua adalah sumber.
- d. *Port* sumber = mendefinisikan nomor *port* dari sumber paket.
- e. Operator arah = mendefinisikan arah dari *rule* contohnya '<->' dan '<->'
- f. Alamat tujuan = mendefinisikan alamat IP tujuan.
- g. *Port* tujuan = mendefinisikan nomor *port* tujuan.
- h. *msg;* = untuk memberi pesan dari *rule* yang dibuat.
- i. *flags;* = untuk menentukan *TCPflag* yang akan dipantau seperti FIN(F) menunjukkan TCP telah selesai, SYN(S) menunjukkan TCP akan membuat sesi koneksi, RST(R) menunjukkan bahwa koneksi akan digagalkan, ACK (A) menunjukkan konfirmasi dari sesi koneksi dan siap mengirim permintaan dan data.
- j. *dsize;* = untuk menentukan ukuran paket yang tidak normal contohnya >300.
- k. *content;* = untuk menetapkan aturan yang mencari konten tertentu dalam muatan paket.
- l. *classtype;* = untuk mengkategorikan aturan yang dibuat, dalam hal ini ada beberapa kategori seperti pada Tabel 3.3 Tipe Kelas Aturan.

**Tabel 3.3 Tipe Kelas Aturan**

Tipe Kelas	Keterangan	Prioritas
<i>attempted-admin</i>	Upaya Mendapatkan Hak Istimewa Administrator	Tinggi (1)
<i>attempted-user</i>	Perolehan Hak Istimewa Pengguna yang Dicoba	Tinggi (1)
<i>inappropriate-content</i>	Konten yang Tidak Pantas Terdeteksi	Tinggi (1)
<i>policy-violation</i>	Potensi Pelanggaran Privasi Perusahaan	Tinggi (1)
<i>shellcode-detect</i>	Kode yang dapat dieksekusi terdeteksi	Tinggi (1)
<i>successful-admin</i>	Keuntungan Hak Istimewa Administrator yang Berhasil	Tinggi (1)
<i>successful-user</i>	Keuntungan Hak Istimewa Pengguna yang Berhasil	Tinggi (1)
<i>trojan-activity</i>	Trojan Jaringan terdeteksi	Tinggi (1)
<i>unsuccessful-user</i>	Keuntungan Hak Istimewa Pengguna yang Tidak Berhasil	Tinggi (1)
<i>web-application-attack</i>	Serangan Aplikasi Web	Tinggi (1)
<i>attempted-dos</i>	Upaya Penolakan Layanan	Sedang (2)
<i>attempted-recon</i>	Percobaan Kebocoran Informasi	Sedang (2)
<i>bad-unknown</i>	Lalu Lintas Berpotensi Buruk	Sedang (2)
<i>default-login-attempt</i>	Mencoba masuk dengan nama pengguna dan kata sandi default	Sedang (2)
<i>denial-of-service</i>	Deteksi Serangan <i>Denial of Service</i>	Sedang (2)
<i>misc-attack</i>	Serangan Lain-lain	Sedang (2)
<i>non-standard-protocol</i>	Deteksi protokol atau acara non-standar	Sedang (2)
<i>rpc-portmap-decode</i>	Decode <i>Query RPC</i>	Sedang (2)
<i>successful-dos</i>	Kegagalan layanan	Sedang (2)
<i>successful-recon-largescale</i>	Kebocoran Informasi Skala Besar	Sedang (2)
<i>successful-recon-limited</i>	Kebocoran Informasi	Sedang (2)
<i>suspicious-filename-detect</i>	Nama <i>file</i> yang mencurigakan terdeteksi	Sedang (2)
<i>suspicious-login</i>	Upaya <i>login</i> menggunakan nama pengguna yang mencurigakan terdeteksi	Sedang (2)
<i>system-call-detect</i>	Panggilan sistem terdeteksi	Sedang (2)
<i>unusual-client-port-connection</i>	Seorang klien menggunakan <i>port</i> yang tidak biasa	Sedang (2)

**Tabel 3.3 Tipe Kelas Aturan (Lanjutan)**

Tipe Kelas	Keterangan	Prioritas
<i>web-application-activity</i>	Akses ke aplikasi web yang berpotensi rentan	Sedang (2)
<i>icmp-event</i>	Acara ICMP umum	Rendah (3)
<i>misc-activity</i>	Aktivitas lain-lain	Rendah (3)
<i>network-scan</i>	Deteksi Pemindaian Jaringan	Rendah (3)
<i>not-suspicious</i>	Lalu Lintas Tidak Mencurigakan	Rendah (3)
<i>protocol-command-decode</i>	Dekode Perintah Protokol Umum	Rendah (3)
<i>string-detect</i>	Sebuah string yang mencurigakan terdeteksi	Rendah (3)
<i>unknown</i>	Lalu Lintas Tidak Diketahui	Rendah (3)
<i>tcp-connection</i>	Koneksi TCP terdeteksi	Sangat Rendah (4)

- m. *detection\_filter*, *count*, *seconds*; = untuk melacak sumber dari paket baik dengan alamat IP sumber atau alamat IP tujuan. Dalam hal ini juga menentukan jumlah paket yang datang dalam bentuk *count* serta menentukan lama jumlah paket yang datang dalam bentuk *seconds*.
- n. *sid*; = untuk mengidentifikasi nomor dari aturan.
- o. *rev*; = untuk mengidentifikasi aturan yang dibuat telah direvisi berapa kali.

#### 4. Memasang Wireshark

Pemasangan Wireshark digunakan untuk menganalisis kinerja jaringan yang melingkupi berbagai hal, mulai dari proses menangkap paket atau informasi pada jaringan. Sebelum dilakukan pemasangan Snort, komputer wajib terhubung dengan internet dan harus masuk menggunakan *superuser* yaitu *root*. Pemasangan Snort dapat menggunakan perintah “*apt-get install Wireshark*” seperti pada Gambar 3.13 Pemasangan Wireshark.



```
root@ips: /home/ips
root@ips:/home/ips# apt-get install wireshark
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libdouble-conversion3 liblua5.2-0 libpcre2-16-0 libqt5score5 libqt5dbus5 libqt5gui5
  libqt5multimedia5
  libqt5multimedia5-plugins libqt5multimediagsttools5 libqt5multimediawidgets5 libqt5network5 li
  bqt5opengl5
  libqt5sprintssupport5 libqt5svg5 libqt5widgets5 libsmi2ldbl libsnappy1v5 libspandsp2 libssh-gcry
  pt-4 libwireshark-data
  libwireshark13 libwiretap10 libwsutil11 libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme
  qttranslations5-l10n
  wireshark-common wireshark-qt
Suggested packages:
  qt5-image-formats-plugins qtwayland5 snmp-mibs-downloader geoipupdate geoip-database geoip-dat
  abase-extra libjs-leaflet
```

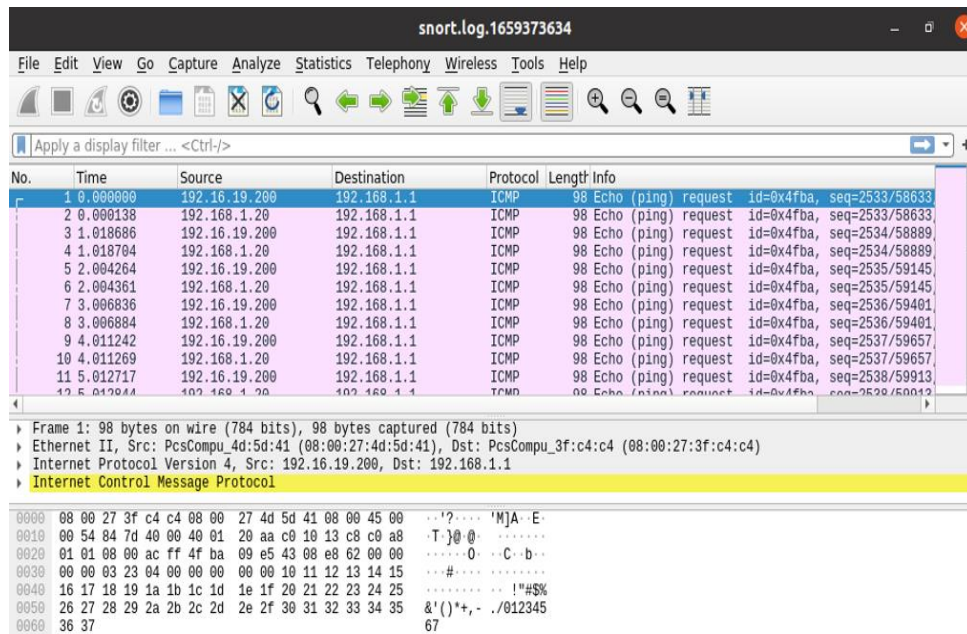
**Gambar 3.13 Pemasangan Wireshark**

Dalam penelitian ini Wireshark dipakai sebagai aplikasi pembantu dalam menganalisis kinerja jaringan dan melihat informasi pada data yang telah ditangkap oleh Snort dan tersimpan pada folder “/var/log/Snort”, seperti pada Gambar 3.14 Tampilan Folder Log.

```
root@ips: /var/log/snort
root@ips:/home/ips# cd /var/log/snort
root@ips:/var/log/snort# ls
192.16.19.200 snort.log.1659195344 snort.log.1659195657
192.168.1.20 snort.log.1659195558 snort.log.1659197002
alert snort.log.1659195616 snort.log.1659373634
root@ips:/var/log/snort# wireshark snort.log.1659373634
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
```

**Gambar 3.14 Tampilan Folder Log**

Berdasarkan Gambar 3.14 Tampilan Folder Log, dilakukan juga uji coba untuk mengakses file log dapat menggunakan perintah “Wireshark file log”. Contoh file log yang berhasil diakses dapat dilihat pada Gambar 3.15 Isi File Log.



Gambar 3.15 Isi File Log

## 5. Memasang Ethtool

Pemasangan Ethtool digunakan untuk memastikan bahwa kartu jaringan tidak memotong paket yang lebih besar 1518 *byte*.. Hal ini disebabkan beberapa kartu jaringan memiliki fitur bernama “*Large Receive Offload*” (*lro*) dan “*Generic Receive Offload*” (*gro*). Oleh sebab itu pihak Snort merekomendasikan untuk mematikan *lro* dan *gro*. Proses pemasangan Ethtool dapat dilihat pada Gambar 3.16 Pemasangan Ethtool.

```

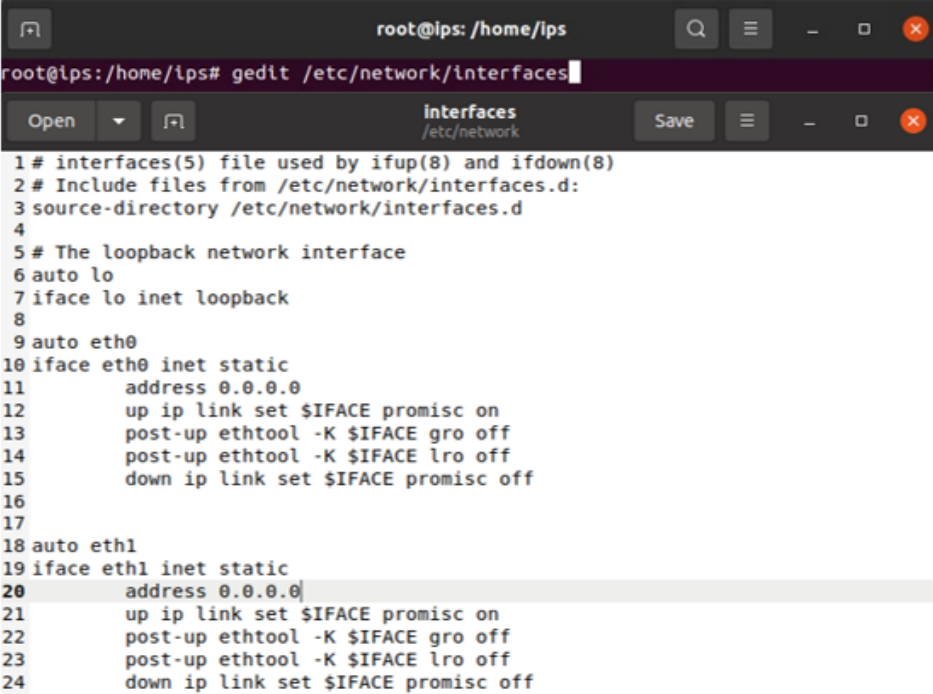
root@ips: /home/ips
root@ips:/home/ips# apt-get install -y ethtool
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 ethtool
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 134 kB of archives.
After this operation, 461 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 ethtool amd64 1:5.4-1 [134 kB]
Fetched 134 kB in 2s (62,3 kB/s)
Selecting previously unselected package ethtool.

```

Gambar 3.16 Pemasangan Ethtool

## 6. Mengatur IP

Proses pengaturan IP dapat menggunakan “*gedit /etc/network/interfaces*” seperti pada Gambar 3.17 Pengaturan IP pada IPS.



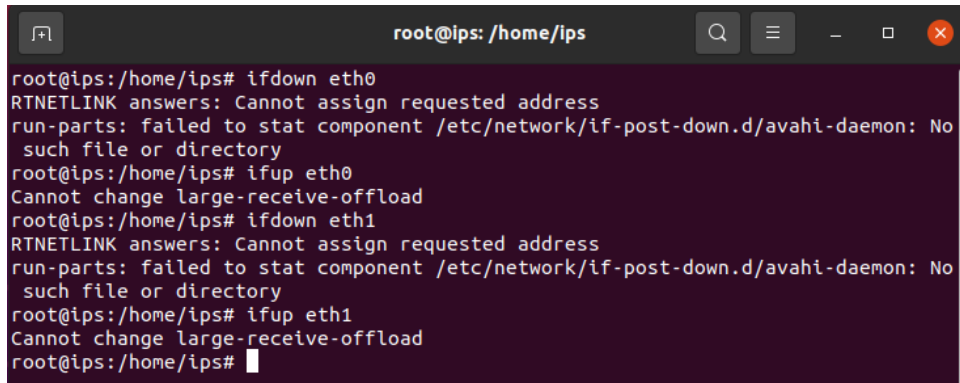
```
root@lps: /home/lps
root@lps: /home/lps# gedit /etc/network/interfaces
interfaces
/etc/network
1 # interfaces(5) file used by ifup(8) and ifdown(8)
2 # Include files from /etc/network/interfaces.d:
3 source-directory /etc/network/interfaces.d
4
5 # The loopback network interface
6 auto lo
7 iface lo inet loopback
8
9 auto eth0
10 iface eth0 inet static
11     address 0.0.0.0
12     up ip link set $IFACE promisc on
13     post-up ethtool -K $IFACE gro off
14     post-up ethtool -K $IFACE lro off
15     down ip link set $IFACE promisc off
16
17
18 auto eth1
19 iface eth1 inet static
20     address 0.0.0.0
21     up ip link set $IFACE promisc on
22     post-up ethtool -K $IFACE gro off
23     post-up ethtool -K $IFACE lro off
24     down ip link set $IFACE promisc off
```

**Gambar 3.17 Pengaturan IP pada IPS**

Berdasarkan Gambar 3.17 Pengaturan IP pada IPS, *interface eth0* menuju ke arah server sedangkan *interface eth1* menuju ke arah luar. Baris “address 0.0.0.0” memberikan ip pada tiap *interface* dengan ip 0.0.0.0, hal ini dikarenakan IPS akan digunakan sebagai *bridge*. Baris “up ip link set \$IFACE promisc on” artinya jika *interface* dinyalakan maka mode *promiscuous* akan diaktifkan. Baris “post-up ethtool -K \$IFACE gro off” digunakan untuk mematikan gro. Baris “post-up ethtool -K \$IFACE lro off” digunakan untuk mematikan lro. Baris “down ip link set \$IFACE promisc off” artinya jika *interface* dimatikan maka mode *promiscuous* akan mati. Pengaturan IP dalam mode *promiscuous* bertujuan supaya setiap *interface* yang akan dijembatani tanpa menggunakan IP. Hal ini mengakibatkan jika Snort tidak berjalan, komputer di satu segmen

jaringan tidak akan dapat berkomunikasi di seluruh sistem Snort ke komputer di segmen lainnya.

Dilanjutkan restart *interface* dengan perintah “*ifdown namainterafce*” untuk mematikan *interface* dan perintah “*ifup namainterafce*” untuk menyalakan *interface* kembali, hal ini dapat dilihat pada Gambar 3.18 Restart *Interface*.



```
root@ips: /home/ips
root@ips:/home/ips# ifdown eth0
RTNETLINK answers: Cannot assign requested address
run-parts: failed to stat component /etc/network/if-post-down.d/avahi-daemon: No
such file or directory
root@ips:/home/ips# ifup eth0
Cannot change large-receive-offload
root@ips:/home/ips# ifdown eth1
RTNETLINK answers: Cannot assign requested address
run-parts: failed to stat component /etc/network/if-post-down.d/avahi-daemon: No
such file or directory
root@ips:/home/ips# ifup eth1
Cannot change large-receive-offload
root@ips:/home/ips#
```

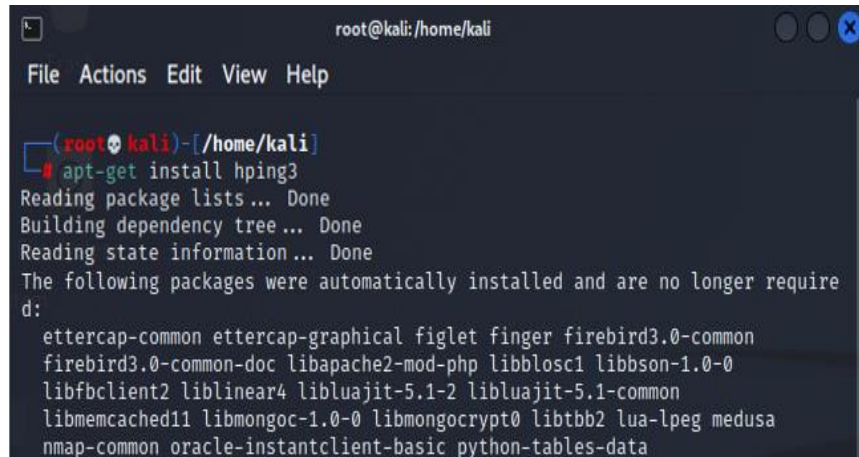
**Gambar 3.18 Restart *interface***

b) Konfigurasi Penyerang

Perangkat yang akan digunakan sebagai penyerang telah dipersiapkan dan akan dikonfigurasi sebelum melakukan tahap pengujian.

1. Memasang hping

Komputer wajib terhubung internet dan menggunakan *superuser* sebelum memasang hping seperti pada Gambar 3.19 Pemasangan Hping.



```
root@kali: /home/kali
File Actions Edit View Help

(root@kali)-[~/home/kali]
└─# apt-get install hping3
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer require
d:
  ettercap-common ettercap-graphical figlet finger firebird3.0-common
  firebird3.0-common-doc libapache2-mod-php libblosc1 libbson-1.0-0
  libfbclient2 liblinear4 liblua5.1-2 liblua5.1-common
  libmemcached11 libmongoc-1.0-0 libmongocrypt0 libtbb2 lua-lpeg medusa
  nmap-common oracle-instantclient-basic python-tables-data
```

**Gambar 3.19 Pemasangan Hping**

Berdasarkan Gambar 3.19 Pemasangan Hping menggunakan perintah “*apt-get install hping3*”. Pemasangan aplikasi Hping akan digunakan dalam penyerangan *ICMP Flood*, *UDP Flood* dan *SYN Flood*.

## 2. Memasang Hydra

Pemasangan aplikasi Hydra akan digunakan dalam *ftp bruteforce*. Untuk memasang aplikasi Hydra dapat menggunakan perintah “*apt-get install hydra*” seperti pada Gambar 3.20 Pemasangan Hydra.



```
root@kali: /home/kali
File Actions Edit View Help

(root@kali)-[~/home/kali]
└─# apt-get install hydra
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer require
d:
  ettercap-common ettercap-graphical figlet finger libapache2-mod-php
  libblosc1 liblua5.1-2 liblua5.1-common libtbb2 medusa
  oracle-instantclient-basic python-tables-data python3-aioredis
  python3-apscheduler python3-bottleneck python3-defusedxml python3-git
  python3-gitdb python3-llvmlite python3-numba python3-numexpr python3-odf
```

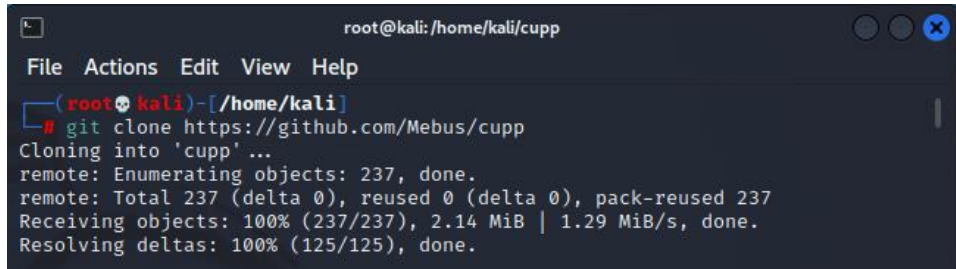
**Gambar 3.20 Pemasangan Hydra**

## 3. Memasang CUPP

Pemasangan CUPP digunakan untuk membuat *wordlist* atau kata yang digunakan untuk melakukan *bruteforce* baik itu *username* atau



password. Untuk memasang CUPP dapat memakai perintah “*git clone* <http://github.com/Mebus/cupp.git>” seperti pada gambar 3.21 Pemasangan CUPP.

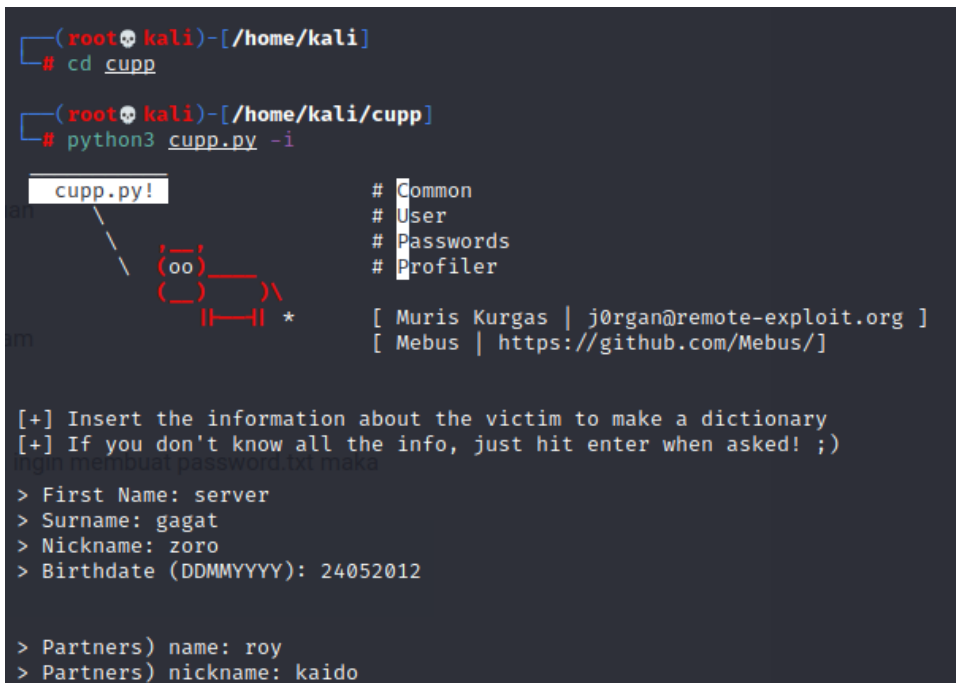


```
root@kali: /home/kali/cupp
File Actions Edit View Help
(root@kali)-[~/kali]
└─# git clone https://github.com/Mebus/cupp
Cloning into 'cupp'...
remote: Enumerating objects: 237, done.
remote: Total 237 (delta 0), reused 0 (delta 0), pack-reused 237
Receiving objects: 100% (237/237), 2.14 MiB | 1.29 MiB/s, done.
Resolving deltas: 100% (125/125), done.
```

**Gambar 3.21 Pemasangan CUPP**

#### 4. Membuat *Worldlist*

Pembuatan *wordlist* dapat menggunakan “*python3 cupp.py -i*” seperti pada Gambar 3.22 Pembuatan *Wordlist*.



```
(root@kali)-[~/kali]
└─# cd cupp
(root@kali)-[~/kali/cupp]
└─# python3 cupp.py -i
cupp.py!
# Common
# User
# Passwords
# Profiler
[ Muris Kurgas | j0rgan@remote-exploit.org ]
[ Mebus | https://github.com/Mebus/ ]

[+] Insert the information about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)
[+] membuat password.txt maka
> First Name: server
> Surname: gagat
> Nickname: zoro
> Birthdate (DDMMYYYY): 24052012

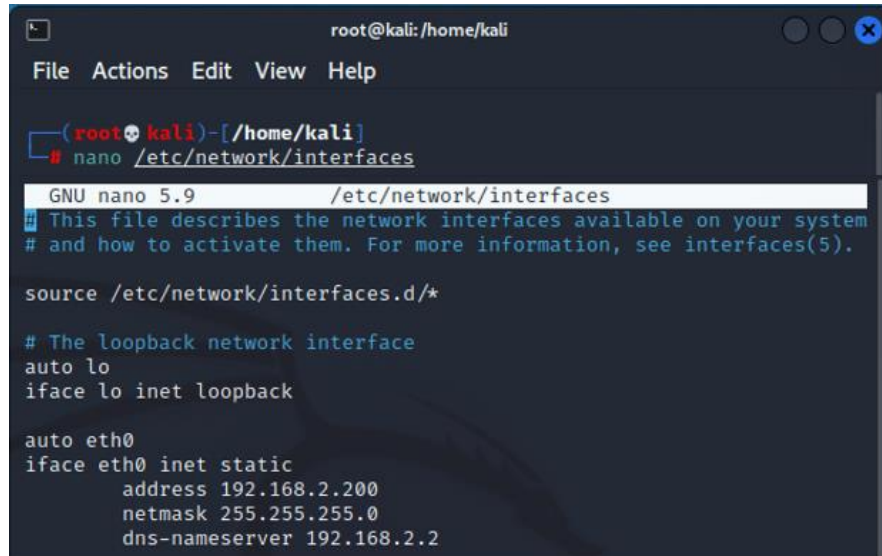
> Partners) name: roy
> Partners) nickname: kaido
```

**Gambar 3.22 Pembuatan *Wordlist***

Namun untuk menjalankan perintah seperti pada Gambar 3.22 Pembuatan *Wordlist*, harus berada pada folder CUPP. Pindah dengan menggunakan “*cd cupp*”. Selain itu harus sudah memiliki aplikasi Python.

## 5. Mengatur IP

Proses pengaturan IP dapat menggunakan “*gedit /etc/network/interfaces*” seperti pada Gambar 3.23 Pengaturan IP pada Penyerang. Kemudian *restart* dengan “*/etc/init.d/networking restart*”.



```
root@kali: /home/kali
File Actions Edit View Help
(root@kali)-[~/home/kali]
# nano /etc/network/interfaces
GNU nano 5.9 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.2.200
    netmask 255.255.255.0
    dns-nameserver 192.168.2.2
```

**Gambar 3.23 Pengaturan IP pada Penyerang**

### 3.2.4 Pengujian

Proses pengujian dilakukan selama lima kali untuk mengetahui kondisi normal suatu jaringan, kondisi saat terjadi serangan *ICMP Flood*, kondisi saat terjadi *UDP Flood*, kondisi saat terjadi *SYN Flood*, kondisi saat terjadi *Bruteforce*.

#### 1) Pengujian *Baseline*

Pengujian *Baseline* dilakukan untuk mengetahui kondisi disaat jaringan digunakan dalam keadaan normal. Untuk melakukan pengujian dapat dengan menyalakan Snort dalam mode *inline* dengan perintah seperti pada Gambar 3.24 Pengujian *Baseline*.



```

root@ips:/var/log/snort# snort -Q -i eth0:eth1 -c /etc/snort/snort.conf -A console > /home/ips/Documents/baseline.txt
Enabling inline operation
Running in IDS mode

      --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"

```

**Gambar 3.24 Pengujian *Baseline***

Merujuk pada Gambar 3.24 Pengujian *Baseline*, perintah tersebut memiliki arti Snort berjalan dalam mode *inline* dengan menjembatani dua *interface* (*eth0* dan *eth1*) memakai aturan dari *file snort.conf* dan peringatan dari aturan akan diubah menjadi dalam teks yang tersimpan pada *file baseline.txt*.

## 2) Pengujian *ICMP Flood*

Pengujian serangan *ICMP Flood* dilakukan dengan menyalakan Snort dalam mode *inline* seperti pada Gambar 3.25 Pengujian *ICMP Flood*.

```

root@ips:/var/log/snort# snort -Q -i eth0:eth1 -c /etc/snort/snort.conf -A console > /home/ips/Documents/icmpflood30s.txt
Enabling inline operation
Running in IDS mode

      --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"

```

**Gambar 3.25 Pengujian *ICMP Flood***

Merujuk pada Gambar 3.25 Pengujian *ICMP Flood*, perintah tersebut memiliki arti Snort berjalan dalam mode *inline* dengan menjembatani dua *interface* (*eth0* dan *eth1*) memakai aturan dari *file snort.conf* dan peringatan dari aturan akan diubah menjadi dalam teks yang tersimpan pada *file icmpflood30s.txt*. Dilanjutkan melakukan penyerangan *ICMP Flood* dengan besar data 300 seperti pada Gambar 3.26 Perintah *ICMP Flood*.

```
(root@kali)-[~/home/kali]
└─# hping3 -1 --data 300 --flood 192.168.2.2
HPING 192.168.2.2 (eth0 192.168.2.2): icmp mode set, 28 headers + 300 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.2.2 hping statistic ---
98157 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

**Gambar 3.26 Perintah ICMP Flood**

### 3) Pengujian UDP Flood

Pengujian serangan *UDP Flood* dilakukan dengan menyalakan Snort dalam mode *inline* seperti pada Gambar 3.27 Pengujian *UDP Flood*.

```
root@ips:/var/log/snort# snort -Q -i eth0:eth1 -c /etc/snort/snort.conf -A console > /home/ips/Documents/udpflood30s.txt
Enabling inline operation
Running in IDS mode

--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
```

**Gambar 3.27 Pengujian UDP Flood**

Merujuk pada Gambar 3.28 Pengujian *UDP Flood*, perintah tersebut memiliki arti Snort berjalan dalam mode *inline* dengan menjembatani dua *interface* (*eth0* dan *eth1*) memakai aturan dari *file snort.conf* dan peringatan dari aturan akan diubah menjadi dalam teks yang tersimpan pada *file udpflood30s.txt*. Dilanjutkan melakukan penyerangan *UDP Flood* dengan besar data 300 seperti pada Gambar 3.28 Perintah *UDP Flood*.

```
(root@kali)-[~/home/kali]
└─# hping3 -2 --data 300 --flood 192.168.2.2
HPING 192.168.2.2 (eth0 192.168.2.2): udp mode set, 28 headers + 300 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.2.2 hping statistic ---
1179825 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

**Gambar 3.28 Perintah UDP Flood**

#### 4) Pengujian *SYN Flood*

Pengujian serangan *SYN Flood* dilakukan dengan menyalakan Snort dalam mode *inline* seperti pada Gambar 3.29 Pengujian *SYN Flood*.

```
root@ips:/var/log/snort# snort -Q -i eth0:eth1 -c /etc/snort/snort.conf -A console > /home/ips/Documents/synflood30s.txt
Enabling inline operation
Running in IDS mode

--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
```

**Gambar 3.29 Pengujian *SYN Flood***

Merujuk pada Gambar 3.29 Pengujian *SYN Flood*, perintah tersebut memiliki arti Snort berjalan dalam mode *inline* dengan menjembatani dua *interface* (*eth0* dan *eth1*) memakai aturan dari *file snort.conf* dan peringatan dari aturan akan diubah menjadi dalam teks yang tersimpan pada *file synflood30s.txt*. Dilanjutkan melakukan penyerangan *SYN Flood* menuju *port 80* atau *website* (HTTP) seperti pada Gambar 3.30 Perintah *SYN Flood*.

```
(root@kali)~[/home/kali]
# hping3 -S --flood -p 80 192.168.2.2 130 x
HPING 192.168.2.2 (eth0 192.168.2.2): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
[send_ip] sendto: No buffer space available
```

**Gambar 3.30 Perintah *SYN Flood***

#### 5) Pengujian *FTP Bruteforce*

Pengujian serangan *FTP Bruteforce* dilakukan dengan menyalakan Snort dalam mode *inline* seperti pada Gambar 3.31 Pengujian *FTP Bruteforce*.

```
root@ips:/home/ips# snort -Q -i eth0:eth1 -c /etc/snort/snort.conf -A console > /home/ips/Documents/ftpbruteforce30s.txt
Enabling inline operation
Running in IDS mode

--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
```

**Gambar 3.31 Pengujian *FTP Bruteforce***

Merujuk pada Gambar 3.31 Pengujian *FTP Bruteforce*, perintah tersebut memiliki arti Snort berjalan dalam mode *inline* dengan menjembatani dua *interface* (*eth0* dan *eth1*) memakai aturan dari *file snort.conf* dan peringatan dari aturan akan diubah menjadi dalam teks yang tersimpan pada *file ftpbruteforce30s.txt*. Dilanjutkan melakukan penyerangan *FTP Bruteforce* menuju *port 80* atau *website* (HTTP) seperti pada Gambar 3.32 Perintah *FTP Bruteforce*.

```
(root@kali)-[~/home/kali]
└─# hydra -L /home/kali/Downloads/username.txt -P /home/kali/Downloads/rd.txt ftp://192.168.2.2
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use for military or secret service organizations, or for illegal purposes (on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-09:50
ka[WARNING] Restorefile (you have 10 seconds to abort ... (use option ip waiting)) from a previous session found, to prevent overwriting, restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 205761753982404 es (l:14344398/p:14344398), ~12860109623901 tries per task
[DATA] attacking ftp://192.168.2.2:21/
[STATUS] 20.00 tries/min, 20 tries in 00:01h, 205761753982390 to do 128318:40h, 10 active
```

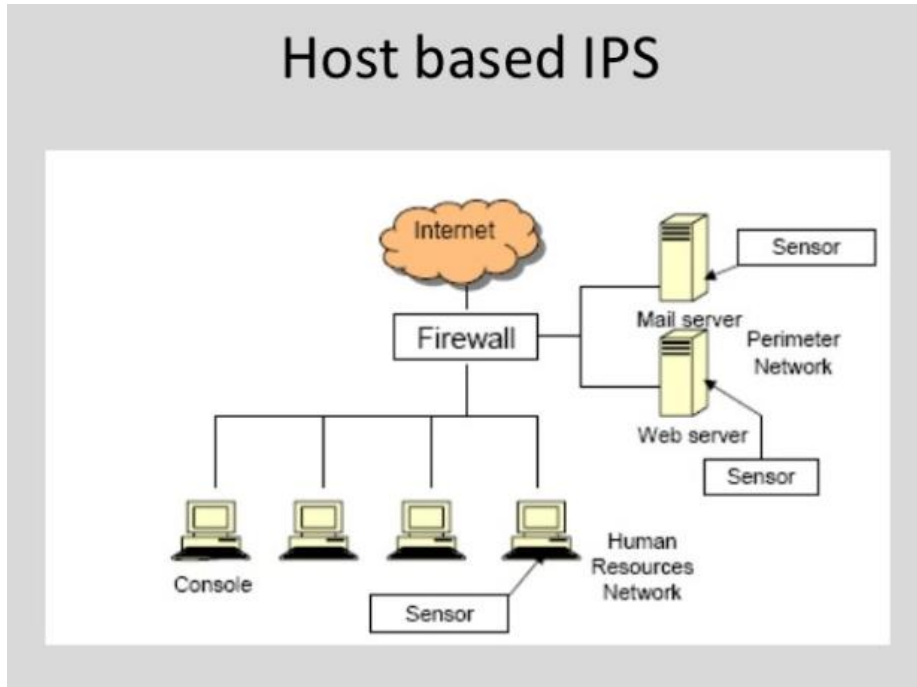
**Gambar 3.32 Perintah *FTP Bruteforce***

### 3.2.5 Hasil dan Kesimpulan

Setelah pengujian berhasil, dilanjutkan dengan pengumpulan data sebagai hasil dari penelitian yang dilakukan. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah mengambil data berupa rekaman dari trafik jaringan secara langsung. Proses pengambilan data ditentukan berdasarkan parameter yang telah ditentukan dalam menentukan masalah. Hasil yang telah diambil akan dianalisis berdasarkan dari *CPU usage*, *throughput*, *packet loss*, *delay*, dan *jitter*. Data dari hasil analisis dapat ditarik kesimpulan apakah hasil penelitian ini mampu membantu dalam mengurangi masalah dan apakah mampu mendeteksi serangan yang dilakukan oleh penyerang.

### 3.3 Alur Kerja *Host-Based Intrusion Prevention System*

Pada penelitian ini dilakukan pengamanan jaringan menggunakan keamanan jaringan *Host-Based Intrusion Prevention System* (HIPS). Alur kerja dari keamanan jaringan yang digunakan dapat dilihat pada Gambar 3.33 Alur Kerja HIPS.



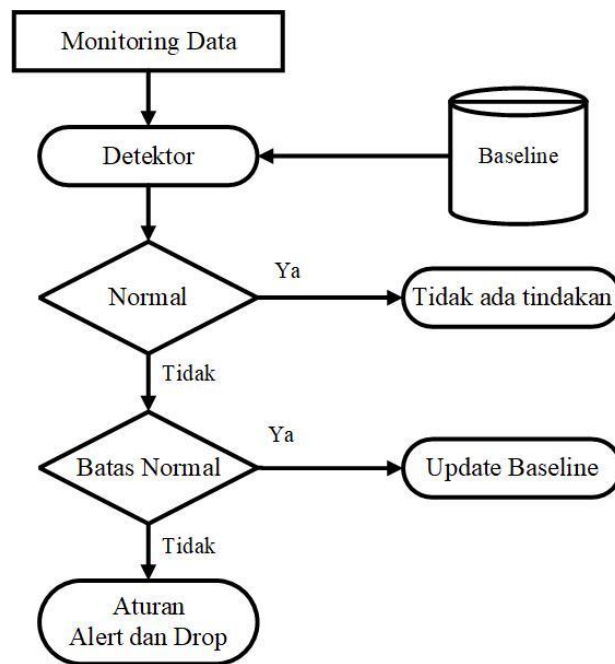
**Gambar 3.33 Alur Kerja HIPS [8]**

Berdasarkan pada Gambar 3.33 Alur Kerja HIPS, HIPS terpasang pada komputer yang berada di luar *server*. HIPS menggunakan aplikasi Snort dan memiliki aturan yang telah dibuat. Jika *client* dan *internet* ingin mengakses layanan dari *server* harus melewati HIPS. Dengan tujuan untuk memfilter aktivitas yang menuju ke *server*. Jika aktivitas tidak normal terdeteksi oleh aturan Snort maka aktivitas tersebut akan dimasukkan ke dalam *log* dan dijatuhkan [8].

### 3.4 Alur Kerja *Anomaly-Based*

Pada penelitian ini dilakukan keamanan jaringan dengan metode deteksi *Anomaly-Based*, dimana dalam menganalisis dengan cara membandingkan aktivitas data yang masuk dan keluar yang diamati dengan *baseline*. Alur kerja dari metode deteksi yang digunakan dapat dilihat pada Gambar 3.34 Alur Kerja *Anomaly-Based*.





**Gambar 3.34 Alur Kerja *Anomaly-Based* [9]**

Berdasarkan pada Gambar 3.34 Alur Kerja *Anomaly-Based*, *baseline* adalah perilaku atau aktivitas normal yang dipantau dengan mencatat setiap komunikasi yang keluar ataupun masuk dari *host* pada jaringan dan dikembangkan selama pembelajaran. Setiap informasi lewat akan dilakukan klasifikasi oleh detektor yang memeriksa peristiwa yang diamati terhadap *profile baseline*. Contoh dari aktivitas normal antara lain memutar video pada suatu *website*, mengecek kondisi suatu *website* dengan mengirimkan *ping* dan sebagainya. Normalnya saat mengecek kondisi *server* menggunakan *ping*, kita memerlukan waktu 32 ms untuk menerima balasan tapi saat anomali terjadi waktu yang dibutuhkan meningkat menjadi 1000 ms hingga *request time out*. Hal ini disebabkan terjadinya penumpukan request *ping* atau meningkatnya trafik pada *server*. Apabila terjadi peningkatan trafik atau perilaku tidak normal, terdapat kemungkinan bahwa suatu serangan sedang terjadi. Jika perilaku tidak normal berada di luar kisaran ambang batas, maka administrator akan memberi membuat aturan dengan perintah *alert* atau peringatan dan menjatuhkan serangan tersebut. Jika perilaku dalam kirsan ambang batas maka administrator akan melakukan *update profile* pada *baseline* [9].