# Bab 4 Analisis dan Hasil

Bab ini membahas tentang langkah-langkah penelitian, analisis dan hasil yang didapatkan dari penelitian ini. Pembahasan dalam bab ini meliputi tahap studi identifikasi system yang digunakan untuk objek penelitian target web server, tahap konfigurasi digunakan untuk mengkonfigurasi *Intrutionn Detection System* (IDS) yang digunakan untuk menguji dalam mendeteksi serangan flooding pada web server. Tahap analisis digunakan untuk mencari barang bukti dari hasil file log *Intrutionn Detection System* (IDS) Snort menggunakan model proses forensic.

## 4.1 Literatur Review

Pada tahapan ini dilakukan kajian literature terhadap penelitian terkait serangan *flooding*, penerapan *Intrusion Detection System* (IDS), dan Snort yang akan dijadikan landasan teknis dalam penelitian ini. Penelitian yang dilakukan oleh (Lanke & Jacob, 2014) menjelaskan bahwa teknik *flooding* merupakan serangan yang ditunjukkan untuk mengacaukan atau menghentikan sebuah layanan secara bersama-sama. Aktifitas *flooding* merupakan serangan terhadap sebuah komputer atau server di dalam jaringan internet dengan cara menghabiskan sumber (*resource*) yang dimiliki oleh komputer tersebut sampai komputer tersebut tidak dapat menjalankan fungsinya dengan benar sehingga secara tidak langsung mencegah pengguna lain untuk memperoleh akses layanan dari komputer yang diserang tersebut(Lanke & Jacob, 2014)(Lanke & Jacob, 2014)(Lanke & Jacob, 2014)(Lanke & Jacob, 2014)(Lanke & Jacob, 2014). Salah satu teknik yang digunakan untuk mendeteksi serangan flooding adalah penerapan *Intrusion Detection System* (IDS), seperti pada penelitian (Stiawan et al., 2012) yang menggunakan *Intrusion Detection System* (IDS) untuk mengidentifikasi adanya serangan dan dapat memberikan peringatan serangan yang terjadi pada *web server*.

Selanjutnya pada penelitian (Indra, 2010) menyebutkan bahwa untuk mengurangi resiko pada celah gangguan keamanan pada jaringan web server menggunakan tool Snort yang merupakan *Intrusion Detection System* (IDS) paling unggul dalam menganalisis lalu lintas dan *packet logging* IP *network.* Snort digunakan untuk mendeteksi ancaman seperti *buffer overflows, port*

*scanning*, nmap maupun *port scanner* lainya. Snort memberikan informasi serangan berupa alert log. Pada penelitian ini akan diterapkan sistem *Intrusion Detection System* (IDS) Snort untuk mendeteksi serangan *flooding* pada *web server* Universitas Muhammadiyah Magelang.

### 4.2. Identifikasi Sistem

Tahap identifikasi sistem jaringan *Intrusion Detection System* (IDS) Snort yang akan digunakan sebagai objek penelitian. Terdiri dari beberapa komponen berupa:

**a. Kebutuhan Perangkat Keras**

Kebutuhan perangkat keras dalam penelitian ini menggunakan beberapa komponen jaringan, berupa:

- PC dengan merk Samsung adalah sebagai berikut:

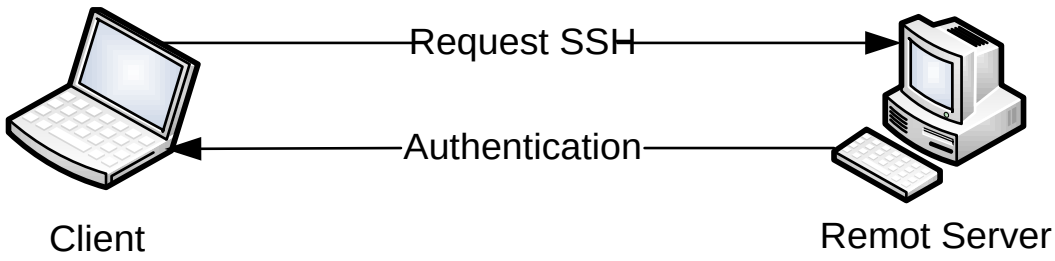
  Prosesor :Intel (R) Core (TM)2 /duo
  RAM : 4 GB
  HDD : 500 GB
  Graphic Card : Intel HD 3000 dan AMD Radeon
- *Router*
- *Switch*
- *Internet Interface Card* (NIC)

**b. Kebutuhan Perangkat Lunak**

Kebutuhan perangkat lunak ang digunakan untuk web server dan kebutuhan forensik dalam penelitian ini adalah:

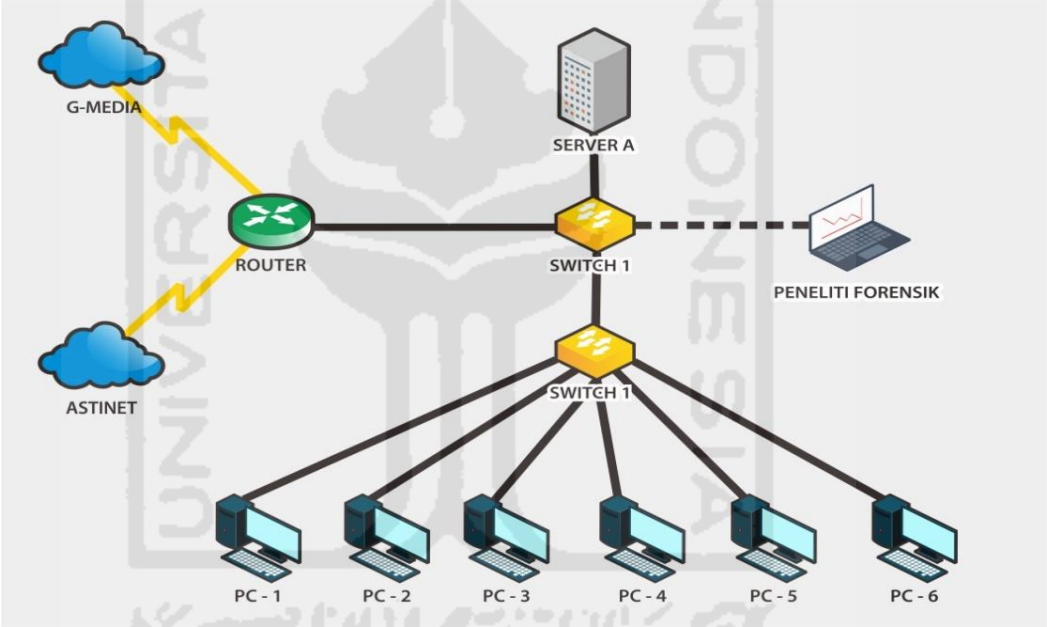
- OS Debian 8 (Apache Web Server, MySQL, Database Server)
- Snort

Penelitian forensic jaringan ini menggunakan server kampus yang bertindak sebagai target saerangan pada saat implementasi. Server ini menggunakan IP static 203.x.x.x yang diakses melalui jaringan internet. Remote server dapat dihubungi dengan menggunakan protocol ssh pada port 22 sehingga proses komunikasi menjadi lebih aman. Lihat Gambar 4.1 yang menunjukkan proses authentifikasi server.



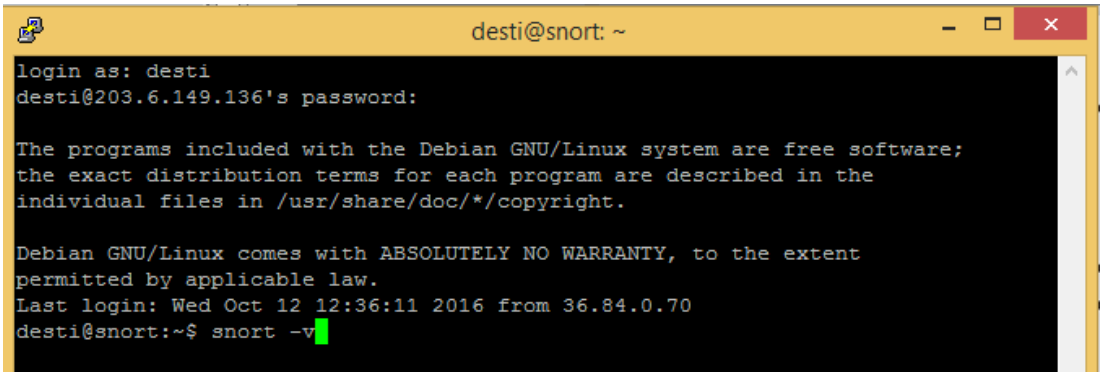

***Gambar 4. 1 Proses Authentifikasi Server***

Gambar 4.1 Melukiskan proses authentifikasi menggunakan *protocol* ssh yang lebih aman dikarenakan menggunakan enkripsi dalam pertukaran data. Letak dari server forensik jaringan dapat dilihat di Gambar 4.2 yaitu artsitektur dari forensik.




***Gambar 4. 2 Arsitektur Forensik Jaringan***

Selanjutnya, Gambar 4.3 Menggambarkan cara menggunakan *remote* pada server forensic jaringan dengan masuk ke *root* untuk konfigurasi snort.




***Gambar 4. 3 Remote Server***

Tahap awal konfigurasi merupakan proses instalasi IDS Snort kemudian memasukkan rule bawaan dari snort untuk mendeteksi serangan flooding. berikut contoh beberapa rule untuk deteksi serangan *flooding* sebagai berikut:

```
# alert tcp $HOME_NET 20432 -> $EXTERNAL_NET any (msg:"MALWARE-OTHER shaft
client login to handler"; flow:to_client,established; content:"login|3A|";
fast_pattern:only; metadata:ruleset community; reference:cve,2000-0138;
reference:url,security.royans.net/info/posts/bugtraq_ddos3.shtml;
classtype:attempted-dos; sid:230; rev:13;)
```

### 4.3 Simulasi Serangan Flooding

Proses simulasi merupakan merupakan tahap awal yang dilakukan untuk menguji konfigurasi *Intrucion Detection System* (IDS) Snort dalam mendeteksi serangan *flooding*. Simulasi serangan menggunakan alat bantu LOIC (*Low Orbit Ion Canon*). LOIC (*Low Orbit Ion Canon*) merupakan alat yang digunakan untuk menguji serangan pada target web server. Alat ini mempunyai kelebihan dapat melakukan pengiriman paket *request* berdasarkan protokol TCP, UDP maupun ICMP. Selain itu target pada *port* yang akan dikirim dapat ditentukan oleh penyerang. Dalam pengujian ini, LOIC digunakan untuk melakukan serangan ke port 80.

Alasan penggunaan port tersebut sebagai target adalah *port* tersebut merupakan *port* yang digunakan dalam mengakses web server yang digunakan oleh pengguna dalam mengakses informasi menggunakan jaringan internet. Proses pengujian serangan dengan memasukkan alamat IP target pada aplikasi LOIC dari mesin *attacker* pada menu 1 (satu) atau *select your target*, kemudian tetapkan alamat yang akan diserang menggunakan tombol *lock on* yang berada pada menu 1 (satu), selanjutkan menentukan target *port* adalah 80, target protokol adalah UDP, jumlah *threads* yang akan dikirimkan sebanak 10, dan kecepatan pengiriman paket pada tingkat faster pada menu 3 (tiga), Aplikasi LOIC ditunjukkan pada Gambar 4.4.

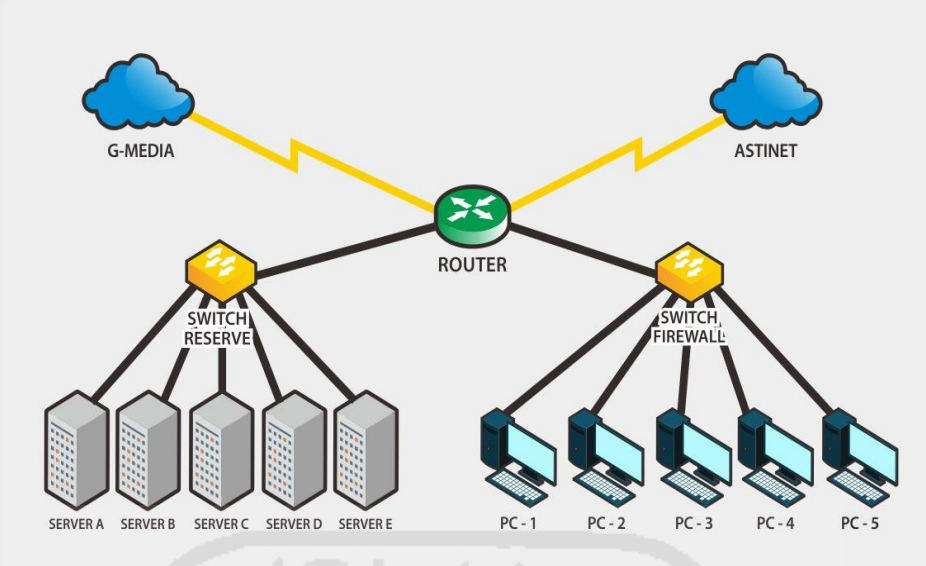

*Gambar 4. 4 Serangan Flooding*

Setelah semua konfigurasi dimasukkan pada aplikasi LOIC kemudian melakukan seranngan dengan menekan tombol *start flooding* untuk memulai atau *stop flooding* untuk menghentikan serangan. Serangan *flooding* dilakukan selama 5-15 menit.

### 4.4 Analisis dan Investigasi Forensik

(Lipeng, Xingyuan, Huilin, & Wang, 2013) teknologi IDS bertujuan untuk mengidentifikasi instruksi ilegal yang tersembunyi pada jaringan lalu lintas yang diserang, penelitian ini berkomitmen untuk menyediakan sutu metode generasi yang sistematis dan dan ilmiah untuk menghindari berbagai bentuk serangan dengan menggunakan kerangka kerja dan rekomendasi untuk pertahanan dari serangan yang masuk.

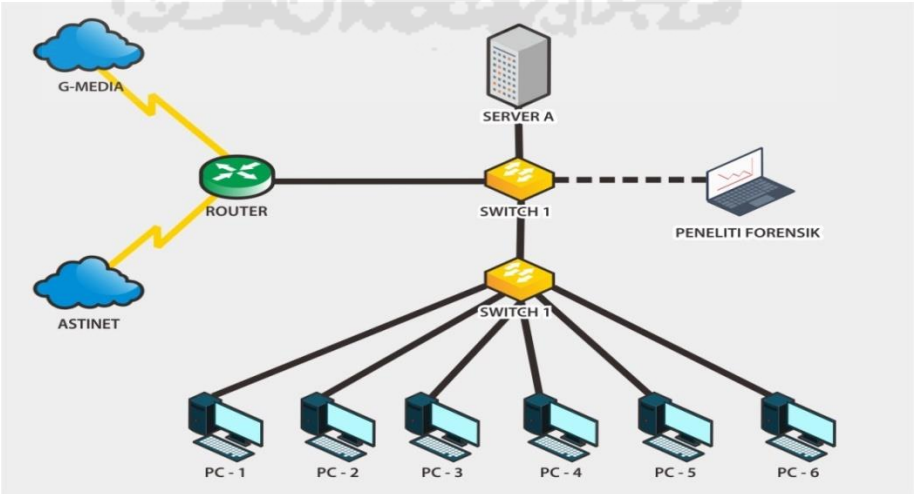
Pusat Pelayanan Teknologi Informasi dan Komunikasi UMMgl (TIK UMMgl) yang merupakan lembaga pelayanan yang berfokus pada pengolahan data, dan interkoneksi kampus. Gambar 4.5 adalah topologi jaringan Universitas Muhammadiyah Magelang adalah distributed (menyebar), pengembanangan dari topologi star, TIK UMMgl menjadi pusat jaringan sekaligus pembagi bandwidth dari tiap-tiap fakultas. Pembagian bandwith ditentukan berdasarkan fakultas.




***Gambar 4. 5 Topologi Jaringan UMMgl***

### 4.4.1 Implementasi Forensik Jaringan

Implementasi pada penelitian forensik jaringan terdapat pada rancangan arsitektur forensic jaringan seperti gambar yang ditunjukkan pada gambar 4.6 yang merupakan arsitektur forensik jaringan Universitas Muhammadiyah Magelang dalam mendeteksi serangan *flooding* menggunakan *Intrusion Detection System* (IDS) Snort. User yang ingin mengakses server yang ada di UMMgl harus melewati switch terlebih dahulu lalu masuk ke dalam proxy kemudian server. Server IDS diletakkan sejajar dengan core switch untuk mendeteksi tindakan illegal pada jaringan. Pengambilan data dilakukan oleh peneliti dengan cara paket *sniffer* yang ada pada *Intrusion Detection System* (IDS) Snort yang telah terekam.




***Gambar 4. 6 Arsitektur Forensik jaringan***

Setelah data tersebut terekam maka proses analisis dilakukan oleh peneliti dalam menguraikan karakteristik file log flooding yang telah terdeteksi oleh IDS Snort. Pada gambar 4.6 Peneliti bertindak sebagai investigator forensik jaringan dimulai dari mempersiapkan sistem, *Intrusion Detection System* (IDS) Snort untuk mendeteksi penyusup pada jaringan, dan menganalisis karakterinstik file log yang didapat dari IDS Snort. Pada Snort telah dimasukkan aturan sebagai pendeteksi pola pada jaringan.

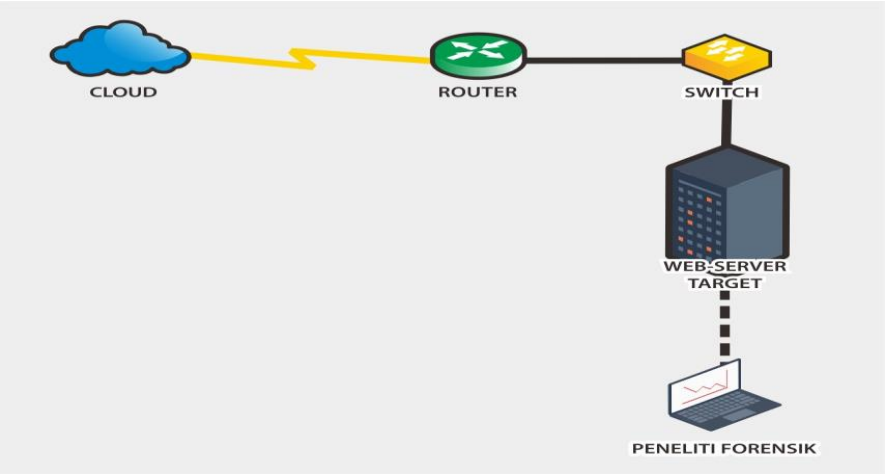
Setelah aturan ditentukan, lalu menangkap trafik jaringan yang memiliki pola yang sama dengan aturan yang telah dimasukkan ke *Intrusion Detection System* (IDS). Dan jika tidak sama pola nya maka tidak akan ditangkap oleh IDS-nya. Agar file log yang didapat bias dianalisis di wireshark maka perlu memasukkan skrip ke *Intrusion Detection System* IDS Snort dengan parameter "snort -r /var/log/snort/snort.log -l /var/log/snort/" sehingga yang dihasilkan dengan file log yang dihasilkan dengan angka biner misalnya snort.log.1498549117.

Selanjutnya, dilakukan data cleaning suatu file log terdiri dari banyak data sehingga perlu dilakukan cleaning agar data yang mau diproses sesuai dengan yang diinginkan. Pengambilan log dilakukan secara real time, kemudian dikumpulkan file log tersebut agar bias dianalisis oleh peneliti investigator. Dengan menggunaan model proses forensic, sebuah file log seharusnya bisa menjawab pertanyaan penyerangan apa yang terjadi, siapa yang menyerang dari IP addressnya, kapan terjadi penyerangan, di server manakah telah terjadi penyerangan, bagaimana itu bisa terjadi dan mengapa itu bisa terjadi.

### 4.4.2 Analisis Model Proses Forensik

**a. Tahap Pengoleksian (*collection*)**

Pengoleksian barang bukti pada penelitian ini menggunakan hasil *record* dari trafik IDS. IDS diimplementasikan kurang lebih selama 3 bulan selama penelitian berlangsung. Proses rekonstruksi dimulai setalah *Intrusion Detection System* (IDS) Snort menangkap trafik yang dianggap *rule* yang telah ditetapkan. Proses pengambilan *payload* sebagai data serangan *flooding* dalam penelitian ini sebagai berikut:



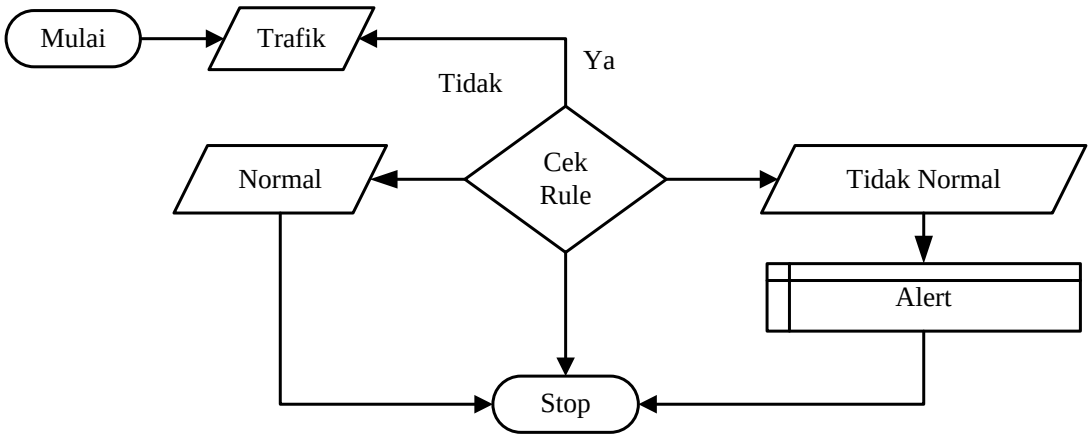

***Gambar 4. 7 Proses Pengambilan Data***

Gambar 4.7 melukiskan bahwa jaringan yang terhubung dan terkoneksi ke internet, kemudian switch tersebut terhubung dengan server *Intrusion Detection System* (IDS) Snort, sehingga apabila ada *traffic* yang bersifat *anomaly* maka akan langsung terdeteksi oleh Snort dan berbunyi alarm.

**b. Tahap Pemeriksaan (*Examination*)**

Peneliti menggunakan *Intrusion Detection System* (IDS) Snort untuk memeriksa penyusupan pada jaringan sehingga jika ingin mengambil *file log* dalam bentuk p.cap (*packet capture*) diperlukan skrip atau parameter untuk dipasang di Snort menggunakan "snort -r /var/log/snort/snort.log -l /var/log/snort/" agar *file log* yang digunakan dalam bentuk default p.cap.

Proses pemeriksaan *file log* dilakukan menggunakan hasil rekaman dari *Intrusion Detection System* (IDS) Snort, pemeriksaan rekaman data akan dikumpulkan dengan cara *packet sniffer* yang terdapat pada server *Intrusion Detection System* (IDS) Snort yang digunakan dalam mendeteksi adanya penyusupan maka akan terlihat urutan pada Gambar 4.8 berikut ini.




***Gambar 4. 8 Alur Intrusion Detection System (IDS) Snort***

Proses pemeriksaan pengambilan *log* yang tersimpan sebagai *alert* pada gambar diatas melukiskan jika ada *traffic* yang melewati server kampus maka *Intrusion Detection System* (IDS) Snort akan mendeteksi cek rule sebagai paket yang normal atau tidak normal dengan menggunakan aturan rule yang sudah ditentukan. Paket yang ditangkap akan diimplementasikan pada saat penelitian berlangsung sehingga akan menghasilkan rekaman trafik yang dianggap melanggar rule yang telah ditetapkan dalam bentuk data file default .pcap.

Data *file log* yang telah berhasil diperiksa selanjutnya akan diambil dalam bentuk *default* p.cap yang terdiri dari beberapa *file log* yang merupakan format paket yang ditangkap setelah memasukkan parameter tertentu pada IDS. Setelah itu, semua data file log dianalsisis menggunakan Wireshark, maka akan dapat dilihat urutan waktunya (*timestamp*).

c. **Tahap Analisis (*Analysis*)**

Pada tahap analisis file log telah diperiksa akan diselidiki lebih dalam. File log yang sudah digabungkan menjadi satu dapat dipakai untuk mengetahui perubahan pada jaringan dan untuk melihat *timestamp*. Data log tersebut diperiksa kembali untuk melihat jenis data apa saja yang berhasil ditangkap dan juga untuk mengetahui protocol apa saja yang banyak digunakan.

Dari aturan rule dibawah ini untuk mendeteksi serangan *flooding* memiliki banyak aturan diantaranya:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DDOS TFN
Probe"; icmp_id:678; itype:8; content:"1234";
reference:arachnids,443; classtype:attempted-recon; sid:221;
rev:4;)
```

Dengan keterangan sebagai berikut:

- `alert` adalah tanda peringatan.
- `icmp` adalah jenis protocol transport.
- `$EXTERNAL_NET any` adalah host asal yang melewati port manapun.
- `->` adalah aliran dari host asal ke host tujuan.
- `$HOME_NET any` adalah host tujuan yang melewati port manapun.
- `(msg:"DDOS TFN Probe"; icmp_id:678;`adalah pesan yang akan dikirimkan jika suatu event terjadi.
- `itype:8;` adalah jenis tipe.
- `content:"1234";` adalah konten tipe spesifik yang dicari.
- `reference:arachnids,443;` adalah referensi ke system pengidentifikasi serangan external.
- `classtype:attempted-recon;` adalah percobaan penyadapan informasi.
- `sid:221;` adalah id aturan snort.
- `rev:4;` adalah revisi aturan yang ke 4.

Aturan berikutnya:

```
alert tcp $HOME_NET any <> $EXTERNAL_NET any (msg:"DDOS shaft
synflood"; flow:stateless; flags:S,12; seq:674711609;
reference:arachnids,253; reference:cve,2000-0138;
classtype:attempted-dos; sid:241; rev:10;)
```

Dengan keterangan sebagai berikut:

- `alert` adalah tanda peringatan.
- `icmp` adalah jenis protocol transport.
- `$EXTERNAL_NET any` adalah host asal yang melewati port manapun.
- `<>` adalah aliran dari host host yang masuk.
- `$HOME_NET any` adalah host tujuan yang melewati port manapun.
- `(msg:"DDOS shaft synflood";`
  `flow:stateless;`
  `flags:S,12;`
  adalah pesan yang akan dikirimkan jika suatu event terjadi.
- `classtype:attempted-dos;` adalah jenis percobaan serangan Denial of Service (DOS).

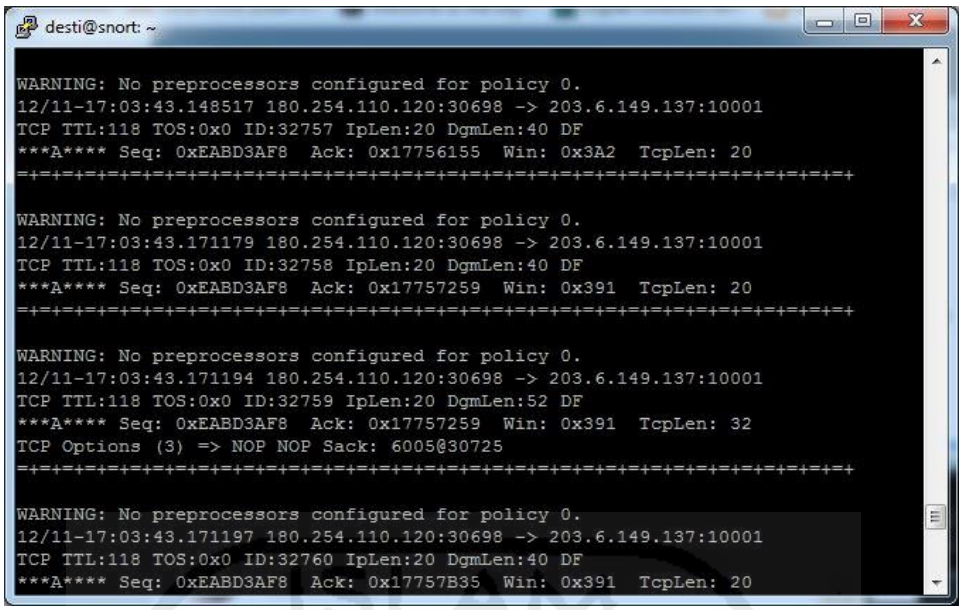
- `reference:arachnids,2533;` adalah referensi ke system pengidentifikasi serangan external.
- `sid:241;` adalah id aturan snort.
- `rev:10;` adalah revisi aturan yang ke 10.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 27665 (msg:"DDOS
Trin00 Attacker to Master default mdie password";
flow:established,to_server; content:"killme"; classtype:bad-
unknown; sid:235; rev:2;)
```

Dengan keterangan sebagai berikut:

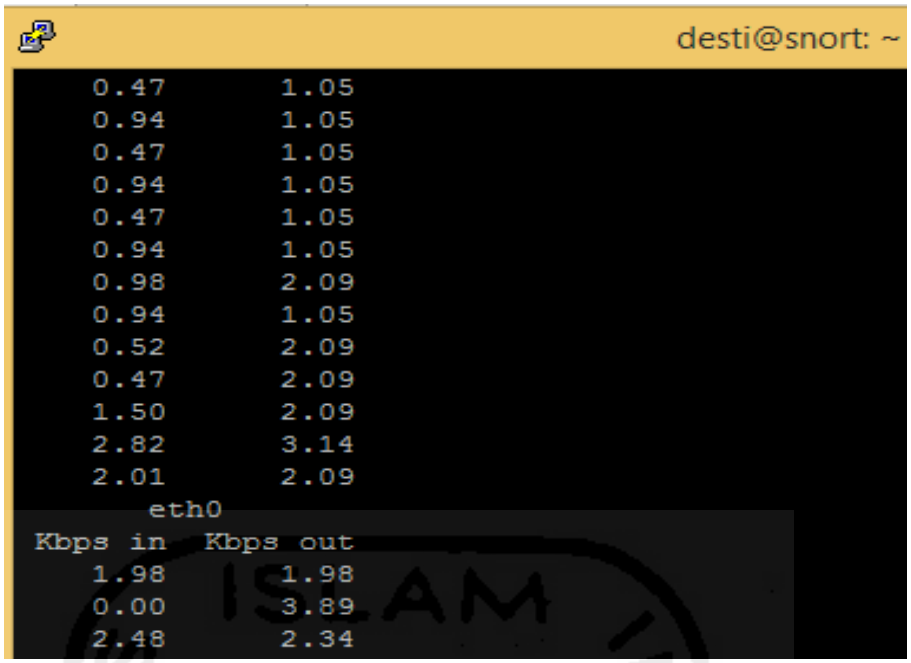
- `alert` adalah tanda peringatan.
- `tcp` adalah jenis protokol transport.
- `$EXTERNAL_NET any` adalah host asal yang melewati port manapun.
- `->` adalah aliran dari host asal ke host tujuan.
- `$HOME_NET 27665 (msg:"DDOS Trin00 Attacker to Master default mdie password";` adalah pesan yang diterima apabila terjadi sebuah event.
- `flow:established,to_server;` adalah koneksi TCP yang dibentuk dari client ke server.
- `content:"killme";` adalah konten spesifikasi yang dicari.
- `classtype:bad-unknown;` adalah trafik yang jelek atau rusak.
- `sid:235;` adalah id aturan snort.
- `rev:2;)`adalah refisi aturan yang ke 2.

Dari simulasi serangan yang telah dikirimkan dalam suatu jaringan akan terlihat *interface* trafik pada *Intrusion Detection System* (IDS) Snort menggunakan rule yang telah di pasang. Deteksi simulasi serangan *flooding* dapat dilihat pada Gambar 4.9 dibawah ini:



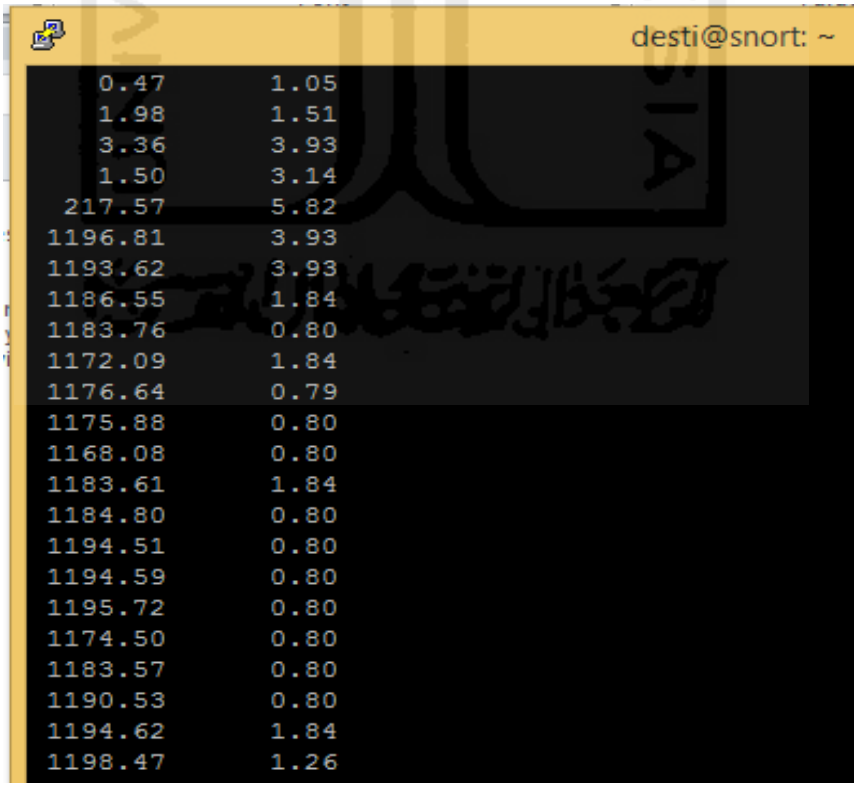

***Gambar 4. 9 Interface Trafik Intrusion Detection System (IDS) Snort***

Pengiriman *request* data berdampak pada turun dan naiknya aktifitas *traffic* selama penggunaanya. Sedangkan pada jam-jam sibuk, *traffic* suatu data akan sangat padat sehingga *traffic* data tersebut akan sangat mengganggu. Baik berupa data yang dikirim maupun data yang akan datang akan mengalami antrian yang mengakibatkan kelambatan dalam pengiriman maupun penerimaan data. Dengan kata lain, adanya serangan *flooding* dapat berdampak pada gangguan akses data yang digunakan oleh aktifitas internet web server dalam suatu *environment.* Di lain waktu data-data yang berada didalam *traffic* merupakan data yang tidak perlu. Data-data tersebut memang sengaja dikirim oleh seseorang untuk merusak jaringan data yang ada. Pengiriman data tersebut mengakibatkan kerugian lain. Sehingga akan terlihat *traffic* yang ada pada *Intrusion Detection System* (IDS) Snort meningkat dari *range* ukuran *kilobyte per second* (kbps) sampai rentang ukuran *megabyte per second* (mbps) tergantung dengan banyaknya paket yang di *capture*. Dibawah Gambar 4.9 merupakan *capture traffic* normal yang tidak memenuhi aturan *rule* sebagai serangan:



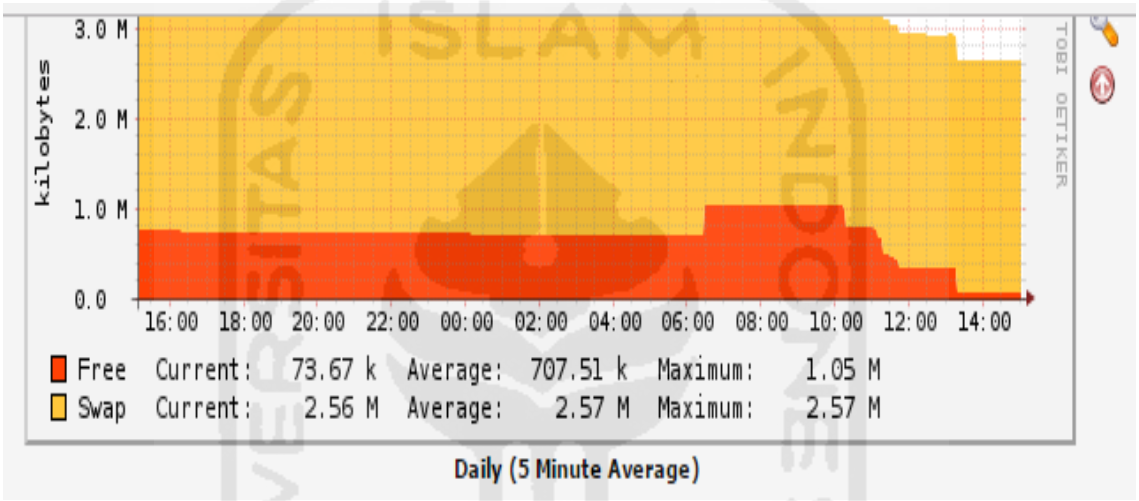

***Gambar 4. 10 Traffic Normal***

Gambar 4.10 diatas terlihat paket yang normal. Maka, setelah dikirim paket menggunakan tool LOIC trafik akan mengalami peningkatan sebagai serangan yang sesuai dengan rule yang telah ditetapkan.



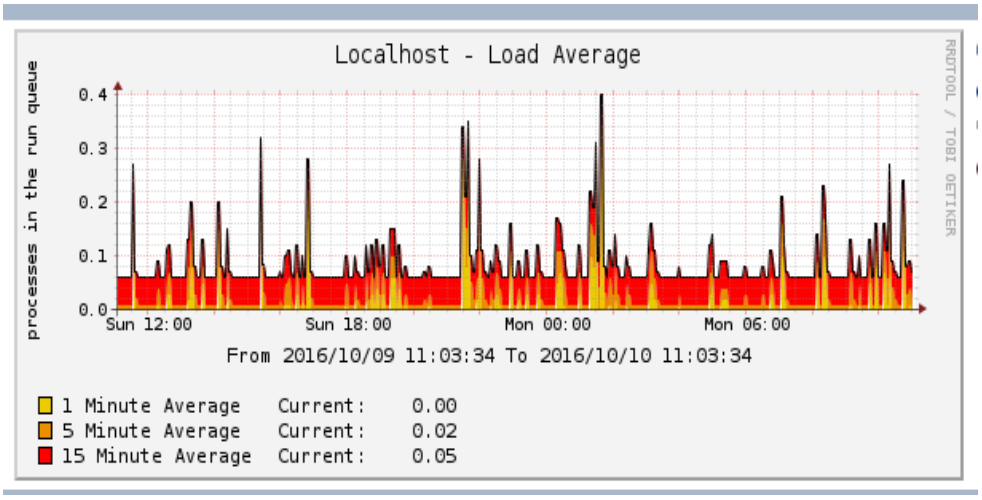

***Gambar 4. 11 Trafik Serangan***

Pada Gambar 4.11 menunjukan trafik yang mengalami peningkatan karena serangan paket yang dikirim oleh *attacker*. Dari *Intrusion Detection System* (IDS) Snort tersebut juga terlihat dari *graph* yang menunjukan dampak aktifitas dari masing-masing pemakaian pengiriman data yang meningkat dalam interval waktu setiap 5 menit yang ditandai dengan skema warna merah sebagai sisa memory yang tidak terpakai, kemudian warna kuning pada saat penggunaan data penggunanaan, atau biasa disebut dengan *log file* yang dapat di pantau pada Gambar 4.12.



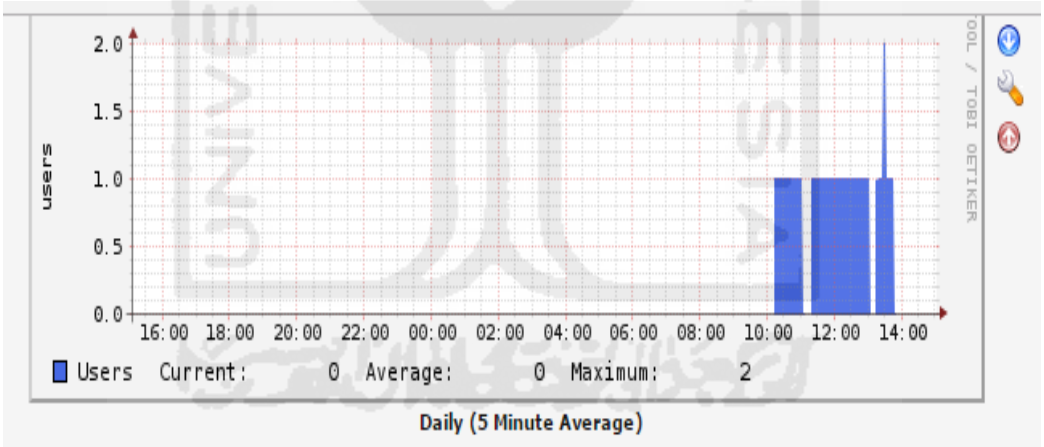

***Gambar 4. 12 Memory Usage***

Pada Gambar 4.12 *free current* mencapai ukuran 73,67 k dengan *average* 707,51 k pada batas *maximum* 1,05 m, selanjutnya *swap current* mencapai 2,56 m dengan *average* 2,57 m dan batas *maximum* 2,57 m. Penggunaan *memory usage* yang dapat dicek pada graph dapat di lihat pada gambar 4.13 apabila sedang terjadi peningkatan pada gambar 4.13 *load average* dalam waktu 15 menit kedepan, saat dilakukan proses simulasi serangan proses yang berjalan pada *load everage* ditandai dengan masing-masing warna, untuk skema warna kuning pemakaian data dalam waktu 1 menit, warna orange penggunaan pada interval 5 menit, selanjutnya warna merah penggunaan data pada interval 15 menit kedepan.



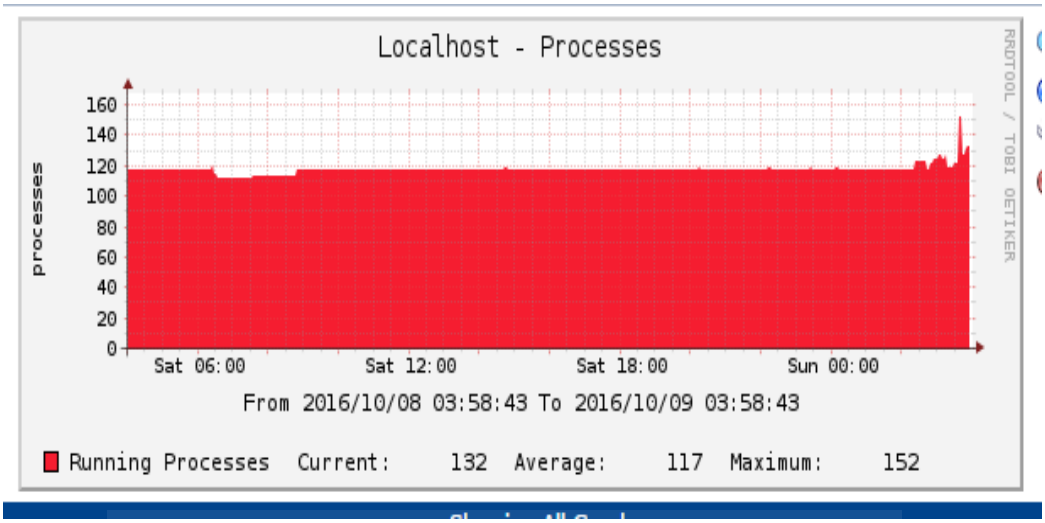

***Gambar 4. 13 Memory Usage***

Selanjutnya pada Gambar 4.13 *logged in users* ditunjukkan dengan skema warna biru yang meningkat tinggi karena kiriman paket yang banyak, serta proses running mencapai batas maximum 2 m pada pukul 14.00.



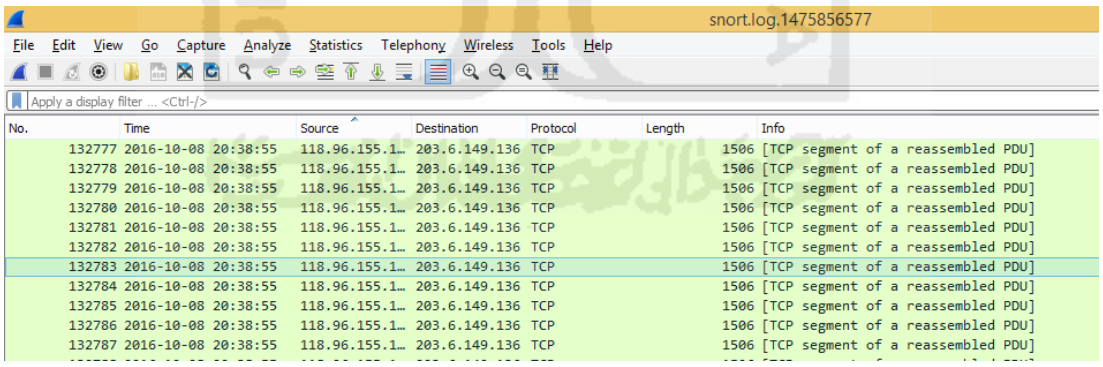

***Gambar 4. 14 logged In Users***

Selanjutnya Gambar 4.14 pada *running process* pada *traffic* meningkat mencapai *running process current* 132 dengan *average* 117 dan *maximum* 152 *user* per 5 menit dari tanggal 8 Oktober 2016 sampai tanggal 9 Oktober 2016.



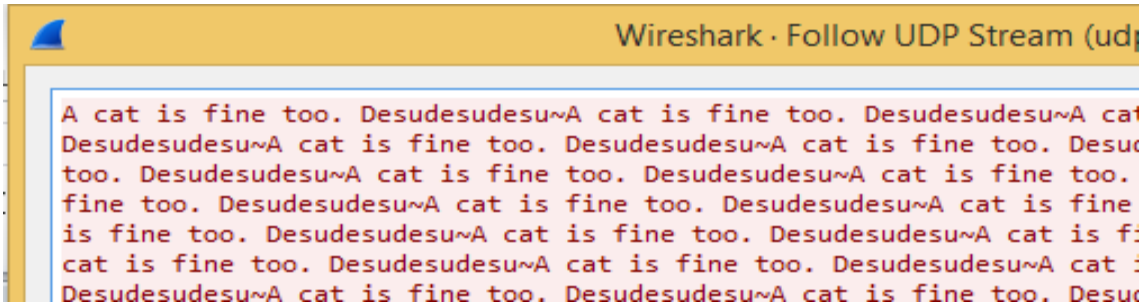

***Gambar 4. 15 Runing Processes***

Setelah melihat grafik yang masuk maka dapat diambil *file log* secara *real time* sehingga dapat dianalisis menggunakan Wireshark dalam mencari bukti penyerang yang mengirimkan paket *flooding* ke sever kampus. Dalam proses analisis aktivitas illegal di dalam jaringan, Wireshark mampu melihat atau menganalisis paket secara *offline* seperti pada Gambar 4.16 dengan menggunakan bantuan filter berdasarkan IP *address* (ip.scr). *Filter* yang dilakukan terhadap ip penyerang sebagai berikut:



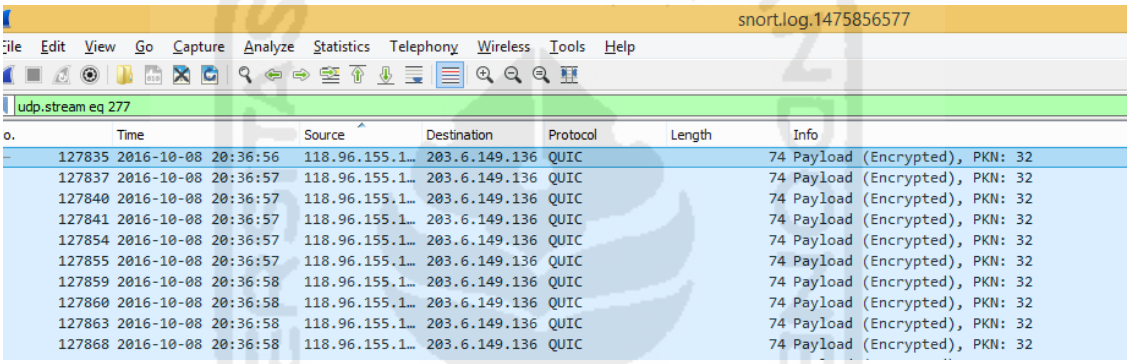

***Gambar 4. 16 Filter ip.src***

IP address 118.96.155.120 melakukan serangan *flooding* kemudian dapat menganalisis dengan filter ip.src= =118.96.155.120, pilih salah satu baris untuk melakukan analisis, kemudian klik kanan Follow UDP Stream pada Gambar 4.17 kemudian *clouse*.



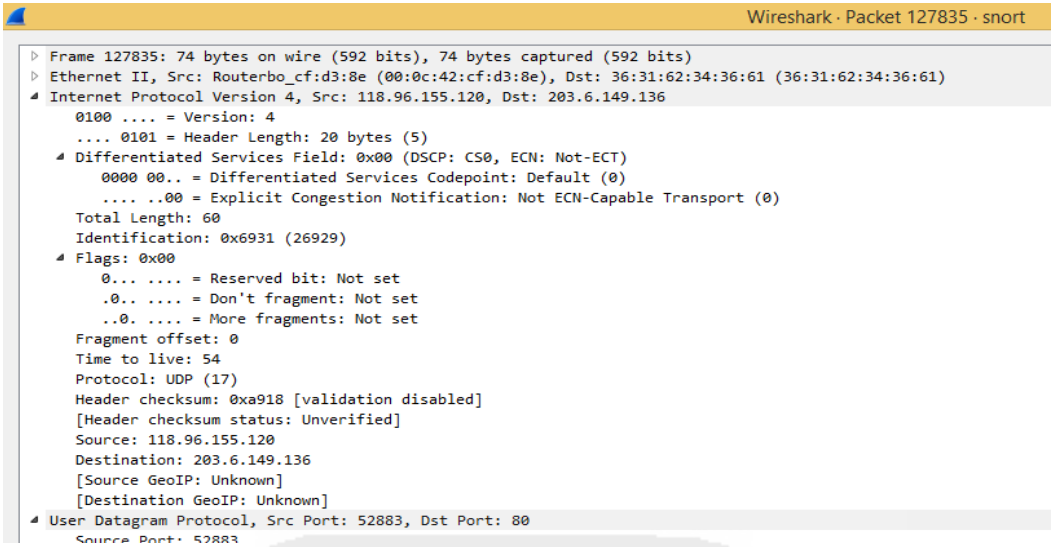

***Gambar 4. 17 Follow UDP***

Setelah memlilih *clouse* maka akan muncul Gambar 4.18 dibawah ini berupa *timestamp, source* IP, *destination* IP, *protocol,* dan *payload*.



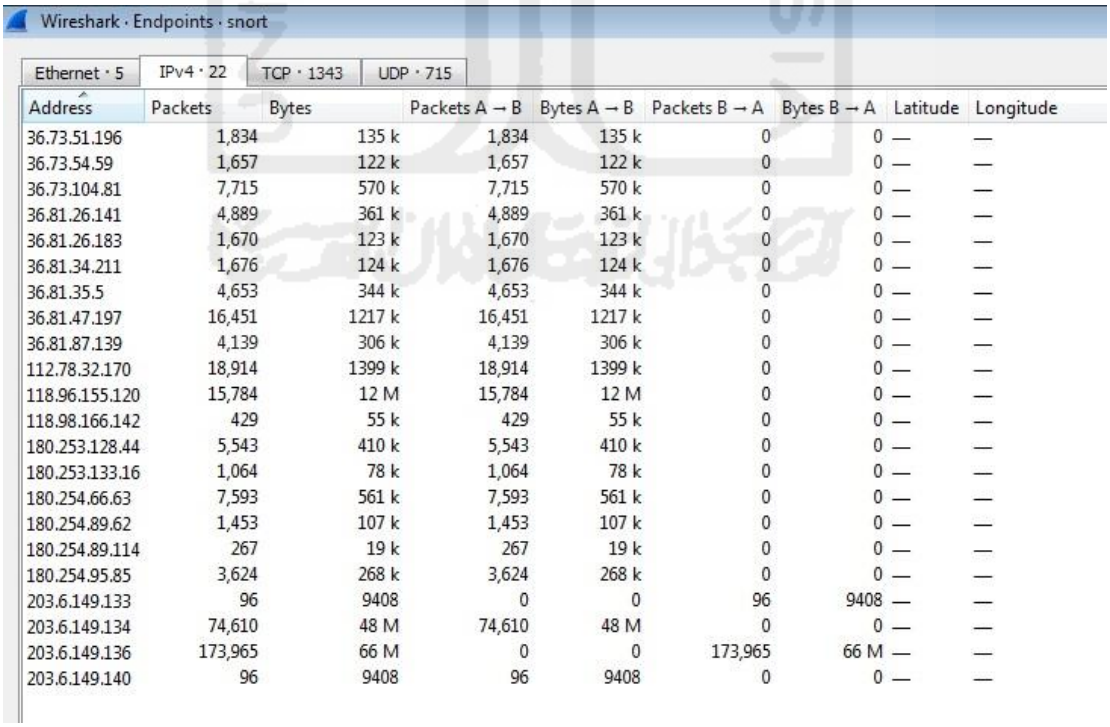

***Gambar 4. 18 Hasil follow UDP***

Dari kumpulan baris tersebut dapat dipilih salah satu baris untuk melakukan analisis pada setiap bagian *frame* yang mewakili sebuah *packet frame* pada serangan *flooding* dari IP address 118.96.155.120 memiliki panjang (length) berkisar pada angka 70-an Bytes (74 Bytes). Pada bagian Internet Protocol Version 4, source IP terbaca sebagai 118.96.55.120 dan destination IP address terbaca 203.6.149.136 dengan *header length* 20 *Bytes* dan total *length* 60. Pada bagian *User Datagram Protocol*, *source port* terbaca sebagai 52883 dan *destination port* terbaca sebagai 80. Apabila *filter* dikembalikan pada ip.src== 118.96.55.120 dan diteliti pada *frame* yang lain, *source port*-nya berubah-rubah, namun tetap dalam rentang yang besar (port 51000-64000). Hasil *capture frame* analisis IP *address* 118.96.55.120 pada Gambar 4.19.




***Gambar 4. 19 Hasil frame***

Selain itu, analisis dilanjutkan dengan modul statistik *endpoint* pada Wireshark yang digunakan untuk mengumpulkan total paket serangan yang terdapat pada *log file Intrusion Detection System* (IDS) Snort selama simulasi serangan berlangsung. Pada Gambar 4.20 dibawah ini menjelaskan bahwa IP address memiliki beban yang berbeda pada setiap paket dan kecepatan yang berbeda pada setiap bytes nya.


Wireshark · Endpoints · snort

Ethernet · 5 | IPv4 · 22 | TCP · 1343 | UDP · 715

| Address | Packets | Bytes | Packets A → B | Bytes A → B | Packets B → A | Bytes B → A | Latitude | Longitude |
|---|---|---|---|---|---|---|---|---|
| 36.73.51.196 | 1,834 | 135 k | 1,834 | 135 k | 0 | 0 | — | — |
| 36.73.54.59 | 1,657 | 122 k | 1,657 | 122 k | 0 | 0 | — | — |
| 36.73.104.81 | 7,715 | 570 k | 7,715 | 570 k | 0 | 0 | — | — |
| 36.81.26.141 | 4,889 | 361 k | 4,889 | 361 k | 0 | 0 | — | — |
| 36.81.26.183 | 1,670 | 123 k | 1,670 | 123 k | 0 | 0 | — | — |
| 36.81.34.211 | 1,676 | 124 k | 1,676 | 124 k | 0 | 0 | — | — |
| 36.81.35.5 | 4,653 | 344 k | 4,653 | 344 k | 0 | 0 | — | — |
| 36.81.47.197 | 16,451 | 1217 k | 16,451 | 1217 k | 0 | 0 | — | — |
| 36.81.87.139 | 4,139 | 306 k | 4,139 | 306 k | 0 | 0 | — | — |
| 112.78.32.170 | 18,914 | 1399 k | 18,914 | 1399 k | 0 | 0 | — | — |
| 118.96.155.120 | 15,784 | 12 M | 15,784 | 12 M | 0 | 0 | — | — |
| 118.98.166.142 | 429 | 55 k | 429 | 55 k | 0 | 0 | — | — |
| 180.253.128.44 | 5,543 | 410 k | 5,543 | 410 k | 0 | 0 | — | — |
| 180.253.133.16 | 1,064 | 78 k | 1,064 | 78 k | 0 | 0 | — | — |
| 180.254.66.63 | 7,593 | 561 k | 7,593 | 561 k | 0 | 0 | — | — |
| 180.254.89.62 | 1,453 | 107 k | 1,453 | 107 k | 0 | 0 | — | — |
| 180.254.89.114 | 267 | 19 k | 267 | 19 k | 0 | 0 | — | — |
| 180.254.95.85 | 3,624 | 268 k | 3,624 | 268 k | 0 | 0 | — | — |
| 203.6.149.133 | 96 | 9408 | 0 | 0 | 96 | 9408 | — | — |
| 203.6.149.134 | 74,610 | 48 M | 74,610 | 48 M | 0 | 0 | — | — |
| 203.6.149.136 | 173,965 | 66 M | 0 | 0 | 173,965 | 66 M | — | — |
| 203.6.149.140 | 96 | 9408 | 96 | 9408 | 0 | 0 | — | — |

***Gambar 4. 20 Statistik Endpoint Snort***

Kemudian, pada statistic *endpoint* Snort terdapat jumlah serangan *flooding* dalam bentuk TCP sejumlah 1343 serangan dan UDP sebanyak 715 serangan.

Selanjutnya dilakukan analisis IP *address* penyerang lainnya seperti langkah-langkah diatas maka dapat dikumpulkan data analisis untuk barang bukti pada tabel 4.1. Terdapat 15 IP *address* penyerang yang melakukan serangan terhadap web server Universitas Muhammadiyah Magelang.

### d. Laporan

Berdasarkan observasi yang dikumpulkan dan diurutkan dari *literature*, dan *eksperiment* yang diimplementasikan pada penelitian ini, membuktikan bahwa serangan *flooding* pada web server merupakan serangan yang memiliki tingkat tinggi di lingkungan Universitas Muhammadiyah Magelang yang memerlukan tindakan cepat untuk deteksi serangan khususnya serangan *flooding*. Selain mendeteksi serangan diperlukan proses investigasi forensic untuk menemukan barang bukti digital tindakan illegal pada web server.

***Tabel 4. 1 Prioritas klasifikasi serangan***

| No. | Timestamp | Source | Dest. IP | Protokol | Source Port | Dest. Port | Payload / Pesan |
|---|---|---|---|---|---|---|---|
| 1 | 7/10/2016 17:26 | 203.6.149.140 | 203.x.x.x | ICMP | - | - | 40ddf957603e0e006e69746f72696e6763616374692d6d6f… |
| 2 | 7/10/2016 16:32 | 112.78.32.170 | 203.x.x.x | UDP | 52658 | 80 | 69732066696e6520746f6f2e20446573756465737375646573… |
| 3 | 8/10/2016 19:28 | 36.73.51.196 | 203.x.x.x | UDP | 58894 | 80 | 69732066696e6520746f6f2e20446573756465737375646573… |
| 4 | 8/10/2016 20:36 | 118.96.155.120 | 203.x.x.x | UDP | 52882 | 80 | 69732066696e6520746f6f2e20446573756465737375646573… |
| 5 | 8/10/2016 19:45 | 180.253.133.16 | 203.x.x.x | UDP | 60052 | 80 | 69732066696e6520746f6f2e20446573756465737375646573… |
| 6 | 8/10/2016 20:26 | 180.253.128.44 | 203.x.x.x | UDP | 63749 | 80 | 69732066696e6520746f6f2e20446573756465737375646573… |
| 7 | 8/10/2016 20:09 | 180.254.95.85 | 203.x.x.x | UDP | 53820 | 80 | 69732066696e6520746f6f2e20446573756465737375646573… |
| 8 | 8/10/2016 20:15 | 180.254.89.62 | 203.x.x.x | UDP | 61246 | 80 | 69732066696e6520746f6f2e20446573756465737375646573… |

***Tabel 4. 2 Con't Prioritas klasifikasi serangan***

| No. | Timestamp | Source | Dest. IP | Protokol | Source Port | Dest. Port | Payload / Pesan |
|---|---|---|---|---|---|---|---|
| 9 | 8/10/2016 20:51 | 180.254.66.63 | 203.x.x.x | UDP | 54948 | 80 | 69732066696e6520746f6f2e20446 573756465737564657 3... |
| 10 | 8/10/2016 20:07 | 36.73.104.81 | 203.x.x.x | UDP | 53817 | 80 | 69732066696e6520746f6f2e20446 573756465737564657 3... |
| 11 | 8/10/2016 20:08 | 36.73.54.59 | 203.x.x.x | UDP | 53814 | 80 | 69732066696e6520746f6f2e20446 573756465737564657 3... |
| 12 | 8/10/2016 20:20 | 36.81.87.139 | 203.x.x.x | UDP | 63748 | 80 | 69732066696e6520746f6f2e20446 573756465737564657 3... |
| 13 | 8/10/2016 20:25 | 36.81.26.141 | 203.x.x.x | UDP | 63756 | 80 | 69732066696e6520746f6f2e20446 573756465737564657 3... |
| 14 | 10/10/2016 6:34 | 36.81.47.197 | 203.x.x.x | UDP | 55291 | 80 | 69732066696e6520746f6f2e204465 737564657375646573... |
| 15 | 10/10/2016 6:34 | 36.81.35.5 | 203.x.x.x | UDP | 56328 | 80 | 69732066696e6520746f6f2e204465 737564657375646573... |

Pada tabel 4.1 terdapat sejumlah 15 IP *address* yang mencoba melakukan serangan, selain itu terdapat *timestamp* yang menjelaskan kapan terjadinya serangan tersebut berlangsung. Selanjutnya *destination* IP *address* merupakan IP yang dijadikan target *attacker*, kemudian protokol merupakan pola serangan yang dilakukan oleh *attacker*. Selain itu, terdapat *port* yang diserang dan pesan/*payload* yang dikirim oleh *attacker* ketika melakukan percobaan serangan.

Berdasarkan informasi yang dikumpulkan dari litelature-litelature dan eksperimen yang diimplementasikan pada penelitian forensik jaringan ini, membuktikan bahwa hasil dari eksperimen dan simulasi deteksi serangan *flooding* pada web server di lingkungan Universitas Muhammadiyah Magelang telah berhasil dan didapatkan barang bukti digital forensik berupa *file log* Snort *Intrusion Detection System* (IDS) sebanyak 15 IP *address* yang mencoba melakukan serangan *flooding* selama penelitian berlangsung.