



**IMPLEMENTASI KINERJA *INTRUSION PREVENTION SYSTEM* (IPS)  
SEBAGAI SISTEM KEAMANAN PADA JARINGAN WIRELESS**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh  
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi  
Universitas Pembangunan Panca Budi  
Medan

---

**SKRIPSI**

---

**OLEH**

**NAMA : CAHYO SETIAWAN**  
**NPM : 1514370267**  
**PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS PEMBANGUNAN PANCA BUDI  
MEDAN  
2019**

## ABSTRAK

CAHYO SETIAWAN

### Implementasi Kinerja Intrusion Prevention System (IPS) Sebagai Sistem Keamanan Pada Jaringan Wireless 2019

Penelitian ini berdasarkan studi pustaka tentang sistem keamanan jaringan. Seiring semakin banyaknya penggunaan komputer sebagai media penghubung antara pengguna 1 dengan yang lainnya, maka diperlukannya sistem keamanan yang baik untuk mengamankan data ataupun hal yang penting pada saat menggunakan jaringan komputer. Dalam keamanan jaringan memiliki beberapa metode yang digunakan untuk mengamankan jaringan tersebut. Pada penelitian ini menggunakan metode *Intrusion Prevention System* yang merupakan sebuah metode keamanan yang menggabungkan antara identifikasi dan penindakan. Metode ini terdapat pada aplikasi *snort* dan dijalankan pada sistem operasi *Ubuntu 16.4*. Dalam penerapannya *Intrusion Prevention System* akan terhubung pada 3 PC dan 1 *wireless router*, pembagiannya sebagai berikut : PC 1 sebagai server, PC 2 sebagai *bridge* yang sudah di *install Intrusion Prevention System*, PC 3 sebagai penyerang dan *wireless router* sebagai penghubung dan penyedia jaringan. Metode ini adalah metode yang paling mudah untuk diterapkan pada sebuah jaringan komputer, oleh sebab itu maka penulis mengangkat judul ini untuk memudahkan pembaca dalam penerapan metode *Intrusion Prevention System*.

**Kata Kunci :** Sistem keamanan jaringan, *Intrusion Prevention System*, *wireless*, *router*, *Ubuntu 16.4*, *snort*, Metode.

## DAFTAR ISI

<b>KATA PENGANTAR</b> .....	<b>i</b>
<b>DAFTAR ISI</b> .....	<b>ii</b>
<b>DAFTAR GAMBAR</b> .....	<b>iv</b>
<b>DAFTAR TABEL</b> .....	<b>v</b>
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	4
1.6 Metode Penelitian .....	4
<b>BAB II LANDASAN TEORI</b>	
2.1 Pengertian Implementasi .....	6
2.2 <i>Brute Force Atack</i> .....	6
2.3 <i>Intrusion Detection System (IDS)</i> .....	7
2.4 Pengertian Keamanan Jaringan .....	8
2.5 <i>Wireless Network</i> .....	9
2.6 <i>Intrusion Prevention System (IPS)</i> .....	10
2.7 Jenis Serangan <i>Cyber</i> .....	11
2.8 <i>Snort</i> .....	12
2.9 <i>Firewall</i> .....	12
2.10 Pengertian <i>Flowchart</i> .....	13
2.11 <i>GNU/Linux</i> .....	15
2.12 Topologi Jaringan .....	16
2.13 Router .....	16
2.14 Local Area Network (LAN) .....	17
2.15 Web Server .....	18
<b>BAB III METODE PENELITIAN</b>	
3.1 Tahapan Penelitian .....	19
3.2 Analisis Masalah .....	21
3.2.1 Perangkat yang Digunakan untuk Mengimplementasikan <i>Intrusion Prevention Sytem (IPS)</i> pada Jaringan <i>Wireless</i> .....	22
3.2.2 Teknik Pemecahan Masalah .....	23
3.3 Konsep Pengamanan Jaringan <i>Wireless</i> Menggunakan Metode <i>Intrusion Prevention System</i> .....	24

3.3.1	Skema kinerja <i>Intrusion Prevention System (IPS)</i> .....	24
3.3.2	Topologi Jaringan sistem keamanan dengan metode <i>Intrusion Prevention System (IPS)</i> .....	25
3.4	Membangun Network Server.....	26
3.4.1	Konfigurasi Server.....	26
3.4.2	Topologi Jaringan Server.....	30
3.5	Membangun Intrusion Prevention System (IPS).....	32
3.5.1	Instalasi Snort.....	33
3.5.2	Konfigurasi Snort.....	35
3.5.3	Konfigurasi Snort Inline Menggunakan Metode Afpacket.....	37
3.5.4	Flowchart Kinerja Intrusion Prevention System.....	40
3.6	Mesin Penyerang.....	41
3.6.1	Installasi Mesin Penyerang.....	41
3.6.2	Topologi Mesin Penyerang.....	43
3.6.3	Flowchart Kinerja Attacker.....	45
3.7	Konfigurasi Router TP-LINK WR840.....	46
3.7.1	Konfigurasi IP Address Router.....	46
3.8	Rincian Biaya Penelitian.....	48

#### **BAB IV IMPLEMENTASI DAN HASIL**

4.1	Serangan <i>Brute Force Attack</i> .....	49
4.1.1	<i>Script</i> untuk membuat serangan Brute force.....	49
4.1.2	Menyerang <i>Server</i> Dengan <i>Software</i> .....	50
4.2	Pengukuran Kinerja Keamanan.....	52
4.2.1	Kinerja Snort.....	53
4.2.2	Kinerja Router.....	54
4.3	Penyerangan dengan 1 Penyerang.....	55
4.3.1	Hasil Pengukuran Kinerja IDS pada VPS.....	56

#### **BAB V PENUTUP**

5.1	Kesimpulan.....	57
5.2	Saran.....	58

#### **DAFTAR PUSTAKA**

#### **BIOGRAFI PENULIS**

#### **LAMPIRAN-LAMPIRAN**

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Saat ini kebutuhan internet merupakan hal yang sangat penting dalam kehidupan di masyarakat karena internet sangat membantu dalam kehidupan sehari-hari, oleh sebab itu dibutuhkan suatu jaringan internet yang cepat dan stabil untuk efisiensi pekerjaan dalam segala bidang. Dengan semakin berkembangnya jaringan internet, sekarang jaringan internet tidak hanya disalurkan melalui kabel bisa juga tanpa kabel, jaringan tanpa kabel disebut jaringan *wireless*, *Wireless* bekerja menggunakan gelombang elektromagnetik sebagai media pengganti kabel. Ada banyak jenis-jenis jaringan *wireless* seperti *Wi-Fi*, *Hotspot*, dan lain-lain.

Dalam jaringan internet yang berbasis *wireless*, keamanan merupakan suatu hal yang sangat penting, karena keamanan merupakan suatu bagian terpenting untuk melindungi sebuah data yang ada pada komputer. Dengan semakin terbukanya pengetahuan tentang *hacking* dan *cracking* yang didukung dengan semakin banyaknya *tools* yang bisa didapatkan secara gratis, maka semakin tinggi pula kejahatan yang ditimbulkan. Banyak metode yang bisa digunakan untuk memecahkan sebuah kode keamanan pada sebuah jaringan internet, salah satu metode yang paling sering digunakan yaitu *brute force attack*. *Brute force attack* merupakan sebuah metode untuk memecahkan kode dalam suatu jaringan dengan mengacak dan mencari password yang cocok secara otomatis, untuk masuk ke sebuah jaringan tanpa sepengetahuan si pemilik jaringan tersebut. Serangan *brute force attack* sangat merugikan untuk pemilik jaringan

internet terutama pemilik jaringan *wifi*, karena pemilik tidak mengetahui bahwa *wifi* sedang digunakan oleh orang lain.

Oleh sebab itu untuk mengamankan sebuah jaringan *wireless*, dibutuhkan sebuah metode keamanan. Banyak metode yang bisa digunakan untuk mengamankan sebuah jaringan, salah satunya IPS (*Intrusion Prevention System*). IPS merupakan gabungan sebuah system keamanan dari *blocking capabilities*, *firewall* dan *IDS (Intrusion Detection System)*. IPS merupakan suatu perangkat lunak atau keras yang berfungsi untuk mengamankan *traffic* data dalam sebuah jaringan, serta bisa melakukan pencegahan terhadap serangan yang terjadi pada jaringan internat yang berbasis *wireless*. Sehingga banyak yang menggunakan IPS karena memiliki tingkat keamanan yang kokoh dan sulit untuk ditembus. Maka dari itu saya mengangkat judul skripsi saya tentang IPS yang berjudul: **“Implementasi kinerja intrusion prevention system (IPS) sebagai sistem keamanan pada jaringan wireless”**.

## 1.2 Rumusan Masalah

Berdasarkan pembahasan diatas mengenai latar belakang bahwa dapat disimpulkan sebagai berikut:

1. Bagaimana cara kinerja sistem keamanan IPS pada jaringan *wireless*?
2. Bagaimana mengimplementasikan IPS pada keamanan jaringan *wireless*?

### 1.3 Batasan Masalah

Dari penelitian yang dilakukan maka terdapat beberapa batasan-batasan, sehingga nantinya dari penelitian tersebut dapat sesuai dengan harapan. Berikut batasan masalah yang akan dibahas sebagai berikut:

1. *Intrusion Prevention System* diimplementasikan untuk keamanan *server* yang terhubung pada jaringan *wireless*.
2. Serangan yang akan di uji coba pada sistem adalah *Brute force Attack* karena serangan ini sering digunakan pada jaringan *wireless*.
3. Target serangan akan dilakukan pada *server* yang terhubung pada jaringan *wireless*, yang dijalankan menggunakan *Wi-Fi*.

### 1.4 Tujuan Penelitian

Adapun tujuan dari implementasi kinerja IPS pada jaringan *wireless* adalah sebagai berikut:

1. Untuk mengetahui kinerja IPS (*Intrusion Prevention System*) dalam mengamankan sebuah jaringan komputer yang terhubung pada jaringan *Wireless*.
2. Untuk mengamankan data yang ada di komputer, sehingga dapat tetap terjaga kerahasiaannya.

### **1.5 Manfaat Penelitian**

Dari penelitian yang dilakukan, bahwa di sebuah perusahaan besar sangat memerlukan tingkat keamanan yang terjaga kerahasiaannya. Maka dari itu hasil manfaat penelitian ini sebagai berikut:

1. Untuk menambah sumber pengetahuan penulis tentang keamanan jaringan wireless dengan menggunakan IPS.
2. Agar IPS ini dapat digunakan sebagai pengamanan jaringan yang bertujuan untuk mengamankan data si pengguna.
3. Hasil dari penelitian ini tentunya dapat digunakan di sebuah perusahaan, perbankan maupun dalam pemerintahan.

### **1.6 Metode Penelitian**

Adapun teknik-teknik pengumpulan data mengenai penulisan skripsi ini sebagai berikut:

1. Penelitian ini bersifat teoritis yang akan dikutip melalui buku bacaan, skripsi, jurnal maupun artikel yang berhubungan dengan permasalahan tentang keamanan jaringan.
2. Analisa Keputusan
3. Pada tahapan ini dilakukan menentukan solusi yang paling berpengaruh dalam memecahkan permasalahan yang dibahas. Dimana sangat berkaitan dengan sebuah perangkat keras maupun perangkat lunak untuk mendukung berjalannya penggunaan sistem yang ada.
4. Desain topologi



5. Desain topologi pada skripsi ini akan menjelaskan tentang model desain, desain *Output* dan *Input*, serta keamanan apa yang digunakan.
6. Instalasi dan konfigurasi
7. Pembuatan sistem melalui metode instalasi dan konfigurasi pada sebuah sistem yang sudah berjalan.
8. Implementasi Sistem
9. Kemudian implementasi pada sistem ini akan menjelaskan tentang isi keseluruhan yang meliputi, tampilan dan kegunaannya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Implementasi**

implementasi adalah aktivitas yang saling menyesuaikan proses interaksi antara tujuan dan tindakan untuk mencapainya serta memerlukan jaringan pelaksana, yang efektif. Kemudian dalam proses untuk melaksanakan kebijakan menjadi tindakan ke dalam administrasi. Pengembangan kebijakan dalam rangka penyempurnaan suatu program.

Dalam *perspektif* hasil, program dapat dinilai berhasil kalau program itu menghasilkan dampak seperti yang diinginkan. Satu program yang mungkin saja berhasil dilihat dari sudut proses, tetapi bisa saja gagal ditangan dan dampak yang dihasilkan atau sebaliknya, untuk mengukur kinerja dari implementasi kebijakan publik pada dasarnya baru memperhatikan variabel-variabel. ( Rini Hadiyanti, 2013)

#### **2.2 Brute Force Attack**

*Bruteforce* merupakan salah satu serangan yang banyak digunakan pada jaringan yang menggunakan sistem keamanan *password* seperti WPA, WPE2K, dan WPS. *Bruteforce* menggunakan metode *matchmaking list* atau pencocokkan kata hingga menemukan *password* yang dicari. Lama waktu pencocokkan kata ini tergantung pada tingkat kerumitan, panjang kata, dan kombinasinya. Penyerang akan menyediakan sebuah *list* yang berisi daftar kata dari mulai satu digit hingga beberapa

digit. *List* yang dibuat oleh penyerang sangat mempengaruhi kecepatan kinerja serangan *bruteforce*. Semakin lengkap *list* yang dibuat oleh penyerang, maka semakin cepat pula *bruteforce* dapat mendapatkan *password* yang diinginkan. Selanjutnya penyerang juga dapat mengatur perulangan pencocokkan hingga *password* didapatkan.

Penyerang yang telah berhasil mendapatkan *password*, dapat dengan mudah masuk ke dalam jaringan dan mengakses berbagai informasi dan data yang terdapat dalam jaringan. Hal ini tentu sangat berbahaya karena akan mengancam keamanan dan kerahasiaan data yang terdapat dalam jaringan. Berdasarkan permasalahan tersebut, maka dibutuhkan sebuah sistem yang dapat meningkatkan keamanan pada jaringan *wireless*, yaitu dengan membangun sebuah sistem yang dilengkapi dengan *IPS (Intrusion Prevention System)* yang dapat mencegah serangan *bruteforce*. Pada penelitian ini, *IPS* yang digunakan adalah *file2ban*. Sistem ini diharapkan dapat membantu administrator jaringan dalam mencegah serangan terhadap jaringan, khususnya serangan *bruteforce*. (Fariz Ayep Prayogi, 2016).

### **2.3 Intrusion Detection System (IDS)**

*Intrusion Detection System (IDS)* adalah suatu perangkat lunak (*software*) atau suatu sistem perangkat keras (*hardware*) yang bekerja secara otomatis untuk memonitor kejadian pada jaringan komputer dan dapat menganalisis masalah keamanan jaringan. Serangan yang terjadi terhadap jaringan komputer selalu meningkat pada infrastruktur keamanan perusahaan dan organisasi yang

menggunakan komputer sebagai alat bantu untuk menyelesaikan pekerjaan. Tipe dasar dari *IDS* adalah :

1. *Rule based system*

Berdasarkan pada signature dan rule yang tersimpan di *database*. Jika *IDS* mencatat lalu-lintas yang sesuai dengan *rule* dan *signature* yang ada, maka langsung dikategorikan sebagai serangan.

2. *Adaptive system*

Mempergunakan metode yang lebih canggih. Tidak hanya berdasarkan *database* yang ada, tetapi juga membuka kemungkinan untuk mendeteksi bentuk-bentuk serangan baru. ( Didit Suhartono , et al.2015)

## 2.4 Pengertian Keamanan Jaringan

Keamanan jaringan pada intinya adalah mengendalikan akses terhadap sumber daya jaringan. Akses jaringan dikontrol agar bisa diakses oleh siapa saja yang berhak dan menghalangi orang atau subjek yang tidak terdaftar untuk mengaksesnya. Prinsip keamanan jaringan di klasifikasikan menjadi 3 bagian :

1. *Confidentiality* ( Kerahasiaan)

*Confidentiality* mengacu pada kerahasiaan dalam sebuah objek, dimana sebuah objek akan dijaga agar tidak diakses oleh subjek yang tidak berhak. Contoh data-data yang sifatnya pribadi adalah nomor kartu kredit, nomor paspor, nama, nomor telepon, *password*, agama, status perkawinan dan lain-lain.

## 2. *Integrity* (Integritas)

*Integrity* mengacu pada objek yang asli (original), dimana objek tidak berubah di perjalanan hingga sampai ke tujuan dari objek tersebut. Sebagai contoh, *email* yang dikirim oleh seseorang bisa di curi ditengah jalan kemudian diubah isinya dan baru dikirim ke penerima sebenarnya sehingga data yang diterima oleh penerima telah berubah dari yang diinginkan oleh pengirim. Bentuk serangan terhadap aspek *integrity* diantaranya adalah *Trojan horse*, *virus*, atau pemakai lain yang berada ditengah komunikasi. Untuk mengatasi hal tersebut, maka perlu dibuat mekanisme proteksi agar data tidak bisa diubah oleh pihak-pihak yang tak diizinkan.

## 3. *Availability* ( Ketersediaan )

*Availability* mengacu pada ketersediaan resource dengan tepat, dimana user mempunyai hak akses tepat waktu dan tidak terkendala apapun. (Syariful Ikhwan dan Ikhwana Elfitri, 2014).

### 2.5 *Wireless Network*

Jaringan lokal tanpa kabel atau *WLAN* adalah suatu jaringan area lokal tanpa kabel dimana media transmisinya menggunakan *frekuensi* radio (RF) dan *infrared* (IR), untuk memberi sebuah koneksi jaringan ke seluruh penggunadalam area disekitarnya. Area jangkauannya dapat berjarak dari ruangan kelas ke seluruh kampus atau dari kantor ke kantor yang lain dan berlainan gedung. Peranti yang umumnya digunakan untuk jaringan *WLAN* termasuk di dalamnya adalah PC, Laptop, PDA, telepon seluler, dan lain sebagainya. Teknologi *WLAN* ini memiliki kegunaan yang sangat banyak. Contohnya, pengguna mobile bisa

menggunakan telepon seluler mereka untuk mengakses e-mail. Sementara itu para pelancong dengan laptopnya bisa terhubung ke internet ketika mereka sedang di bandara, kafe, kereta api dan tempat publik lainnya. (Dedi Darmawan dan Linda Marlinda, 2015).

## **2.6 *Intrusion Prevention System (IPS)***

*Intrusion Prevention System (IPS)* adalah pendekatan yang sering digunakan sistem keamanan komputer, IPS mengkombinasikan teknik *firewall* dan metode *Intrusion Detection System (IDS)* dengan sangat baik. Teknologi ini dapat digunakan untuk mencegah serangan yang akan masuk ke jaringan lokal dengan memeriksa dan mencatat semua paket dan serta mengenali paket dengan sensor, disaat *attack* telah teridentifikasi, IPS akan menolak akses dan mencatat semua paket data yang teridentifikasi. Jadi IPS bertindak seperti layaknya *firewall* yang akan melakukan pengizinan dan penolakan paket yang berbahaya yang dikombinasikan dengan IDS yang dapat mendeteksi paket secara *detail*. (Dedit Suhartono, et al. 2015).

Secara umum IPS memiliki 4 komponen utama yaitu :

1. *Normalisasi Traffic*
2. *Detection Engine*
3. *Service Scanner*
4. *Traffic shaper*

## 2.7 Jenis Serangan Cyber

Beberapa jenis serangan yang umum terjadi pada system keamanan diantaranya *port scanning*, *sniffing*, *ICMP flood*, dan *hijacking*. *Port scanning* merupakan suatu proses untuk mencari dan membuka pada *port* komunikasi pada sebuah celah jaringan komputer. Dari hasil serangan tersebut akan didapatkan celah atau lubang kelemahan sebuah *server* yang diserang. *Packet sniffing* merupakan pengecatan data paket-paket yang mengalir pada jaringan. Dengan sebuah aplikasi yang beroperasi pada lapisan ke 2 OSI dan juga kombinasi dari NIC yang berada pada mode *promiscuous* (mode mendengar) untuk menangkap semua *traffic* yang mengalir dari dan menuju ke jaringan internet pada suatu jaringan. *ICMP flood* dilakukan oleh seorang *hacker* dengan cara melakukan eksploitasi ke *system server* dengan tujuan untuk membuat suatu target menjadi hang, yang disebabkan oleh pengiriman sejumlah paket yang besar ke arah target *server*. *Exploiting* sistem ini dilakukan dengan mengirimkan suatu *command ping* dengan tujuan *broadcast* ataupun *multicast* dimana si pengirim dibuat seolah-olah adalah target host. *Hijacking* atau yang disebut dengan *man-in-the-middle-attack* (MITM) sebuah teknik serangan yang memanfaatkan kelemahan dari *protocol TCP/IP*. Serangan dilakukan ketika terdapat diantara 2 user yang sedang berkomunikasi, tetapi terdapat seseorang yang lain yang secara aktif memonitor, *men-capture*, dan mengontrol komunikasi tersebut secara transparan. (Shah Khadafi, et al. 2017).

## 2.8 *Snort*

*Snort* merupakan suatu *tools* yang berjalan di dalam *system Linux* yang dapat digunakan untuk mendeteksi adanya penyusup (*threats*) dan mampu menganalisis paket yang melintasi jaringan secara *real time traffic* dan *logging* ke dalam *database*. *Snort* juga mampu mendeteksi berbagai serangan yang berasal dari luar jaringan. *Snort* bisa digunakan pada *platform* sistem operasi *Linux*, *Free BSD*, *Debian*, dan *Windows*. *Snort* memiliki arsitektur yang terdiri dari 4 basic komponen, yaitu *sniffer*, *preprocessor*, *detection engine*, dan *output*. (Shah Khadafi, et al. 2017).

## 2.9 *Firewall*

*Firewall* adalah suatu aturan-aturan yang mekanismenya bertujuan untuk melindungi *hardware* dan *software*. Perlindungan dapat dilakukan dengan menyaring, membatasi, atau bahkan menolak suatu atau semua hubungan/kegiatan dari suatu segmen pada jaringan pribadi dengan jaringan luar yang bukan merupakan ruang lingkungannya. Salah satu *tool firewall* yang umum digunakan pada sistem *Linux* yaitu *Iptables*. *Iptables* memungkinkan untuk seorang admin jaringan untuk merancang dan mengkonfigurasi setingan *firewall*. Selain itu juga admin juga dapat mengkonfigurasi rantai-rantai atau biasa disebut dengan *chains* dan *rules* di dalam *system Linux*. (Shah Khadafi, et al. 2017).

Menurut Marco Van Basten (2009) *Firewall* merupakan alat untuk mengimplementasikan kebijakan *security* (*security policy*). Sedangkan kebijakan *security*, dibuat berdasarkan pertimbangan antara fasilitas yang disediakan dengan



implikasi *security*-nya. Semakin ketat kebijakan *security*, semakin kompleks konfigurasi layanan informasi atau semakin sedikit fasilitas yang tersedia di jaringan. *Firewall* mempunyai beberapa tugas:


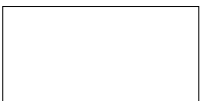
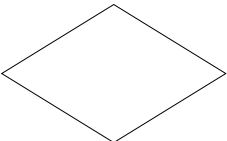

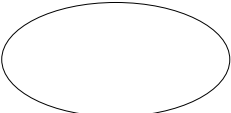
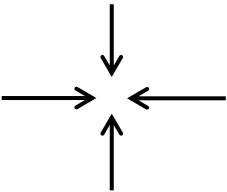

1. Harus dapat mengimplementasikan kebijakan *security* di jaringan (*site security policy*). Jika aksi tertentu tidak di perbolehkan oleh kebijakan ini, maka *firewall* harus meyakinkan bahwa semua usaha yang mewakili operasi tersebut harus gagal atau digagalkan. Dengan demikian, semua akses *illegal* antar jaringan (tidak diotoritaskan) akan ditolak.
2. Melakukan *filtering* yaitu dengan mewajibkan semua trafik yang ada untuk dilewatkan melalui *firewall* bagi semua proses pemberian dan pemanfaatan layanan informasi. Dalam konteks ini, aliran paket data dari/menjuhu *firewall* bagi semua proses pemberian dan pemanfaatan layanan informasi. Dalam nomor *port*, atau arahnya, dan disesuaikan dengan kebijakan *security*.
3. *Firewall* juga harus dapat merekam/mencatat even-even mencurigikan serta memberitahu *administrator* terhadap segala usaha-usaha menembus kebijakan *security*. ( Didit Suhartono , et al.2015)

## **2.10 Pengertian *Flowchart***

*Flowchart* adalah yang berisikan simbol-simbol untuk menentukan alur dari perancangan sistem. *Flowchart* berfungsi sebagai alat bantu dalam mempersiapkan program yang sukar dan sebagai garis alur dalam mengerjakan sistem yang kita buat, sehingga sistem tersebut dapat tersusun dengan rapi sesuai rangkaian pada alur program. Simbol-simbol yang khusus dalam pembuatan

*flowchart* untuk merangkai garis alur program yang memiliki fungsi masing-masing, sebagai berikut:

**Tabel 2.1** Simbol-simbol *Flowchart*

o	Simbol <i>Flowchart</i>	Fungsinya
		<i>Terminal</i> atau <i>Start</i> , berfungsi untuk memulai dan mengakhiri alur program.
		<i>Process</i> , adalah untuk mengolah dan mengubah data yang ada didalam komputer.
		<i>Decision</i> , digunakan untuk menentukan operasi perbandingan logika ketika masuk pada alur program.
		<i>Input</i> dan <i>Output</i> , adalah simbol yang digunakan untuk memasukan data yang biasanya berupa <i>username</i> dan <i>password</i> , dimana hasil dari proses.
		<i>Connector</i> , adalah menentukan hubungan arus proses program yang berjalan dalam halaman yang sama.
		<i>Arrow Flow</i> , adalah untuk menunjukkan alur proses program yang terdiri dari, alur atas ke bawah, kanan ke kiri dan juga sebaliknya.
		<i>Document</i> , adalah sebuah simbol untuk data atau informasi.

## 2.11 GNU/Linux

GNU merupakan singkatan *rekursif* dari “*GNU's Not Unix*” (*GNU* bukan *Unix*) serta dilafalkan ge-nuu. Proyek *GNU* diluncurkan pada tahun 1984 untuk mengembangkan sebuah sistem operasi lengkap serupa *Unix* yang berbasis perangkat lunak bebas yaitu sistem *GNU*. *Kernel GNU* tidak pernah rampung, sehingga *GNU* menggunakan *kernel Linux*. Kombinasi *GNU* dan *Linux* merupakan sistem operasi *GNU/Linux*, yang kini digunakan secara meluas. Proyek *GNU* telah mengembangkan sebuah sistem perangkat lunak bebas lengkap yaitu “*GNU*” (*GNU's Not Unix*, *GNU* bukan *Unix*) yang kompatibel dengan *Unix*. Richard Stallman menulis dokumen pertama dari proyek ini yaitu *Manifesto GNU* (31k huruf), yang telah diterjemahkan ke berbagai bahasa lain. Pengumuman pertama perihal proyek ini ditulis pada tahun 1983. Kata “bebas” di atas menyangkut pengertian kebebasan, dan bukan bebas tidak membayar. Anda mungkin perlu atau pun tidak perlu membayar, untuk mendapatkan perangkat lunak *GNU*. Dengan cara yang mana pun, setelah memiliki perangkat lunak tersebut, anda mendapatkan tiga jenis “kebebasan” dalam menggunakannya. Pertama, kebebasan untuk menggandakan program tersebut serta memberikannya ke teman atau sejawat anda. Kedua, kebebasan untuk merubah *source code* program sesuai dengan keinginan anda. Ketiga, kebebasan untuk mendistribusikan dan versi perbaikan, sehingga ikut membantu pembangunan masyarakat (Jika anda kita mendistribusikan ulang perangkat lunak *GNU*, anda dapat meminta biaya duplikasi, atau juga dapat memberikan secara cuma-cuma. (Edy Budi Harjono, 2016).

## 2.12 Topologi jaringan

Iwan Sofana, 2008:7. Topologi adalah suatu aturan/rules bagaimana menghubungkan komputer (*node*) satu sama lain secara fisik dan pola hubungan antara komponen-komponen yang berkomunikasi melalui media/peralatan jaringan, seperti : *server, workstation, hub/switch*, dan pengabelannya, sedangkan jaringan merupakan sebuah sistem yang terdiri atas komputer, perangkat komputer, tambahan dan perangkat jaringan lainnya yang saling berhubungan dengan menggunakan media tertentu dengan aturan yang sudah ditetapkan. (Satukan Halawa, 2016).

Topologi jaringan komputer adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Dalam suatu jaringan komputer jenis topologi yang dipilih akan mempengaruhi kecepatan komunikasi. Untuk itu maka perlu dicermati kelebihan/keuntungan dan kekurangan/kerugian dari masing-masing topologi berdasarkan karakteristiknya. Jenisi-Jenis Topologi: (Napoleon Lukman, 2016)

1. Topologi *Bus*
2. Topologi *Star*
3. Topologi *Ring*
4. Topologi *Mesh*

## 2.13 Router

*Router* adalah sebuah alat jaringan komputer yang mengirimkan paket data melalui sebuah jaringan atau Internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing*. Proses *routing* terjadi pada lapisan 3 (Lapisan

jaringan seperti *Internet Protocol*) dari *stack protokol*). Sebuah router mampu mengirimkan data/informasi dari satu jaringan ke jaringan lain yang berbeda. (Herlina Latipa Sari, Aji Sudarsono, dan B.Herawan Hayadi, 2013).

*Router* merupakan jaringan internal di dalam sebuah gedung atau kampus. LAN sering kali digunakan untuk menghubungkan komputerkomputer pribadi dan workstation dalam kantor suatu organisasi, perusahaan atau pabrik-pabrik untuk memakai bersama sumberdaya (misalnya *printer*, media penyimpanan/*storage*) dan saling bertukar informasi. (Napoleon Lukman, 2016).

#### **2.14 Local Area Network (LAN)**

Melwin, 2005:16. Sebuah LAN adalah jaringan yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkungan, seperti sebuah kantor pada setia gedung, atau tiap-tiap ruangan pada sebuah sekolah. Biasanya jarak antarnode tidak lebih jauh dari sekitar 200 m. (Herlina Latipa Sari, Aji Sudarsono dan B.Herawan Hayadi, 2013).

Sifat-sifat LAN selain areanya local adalah memiliki kecepatan yang sangat tinggi. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor perusahaan atau pabrik-pabrik untuk memakai bersama resource (misalnya, printer, scanner) dan saling bertukar informasi. LAN dapat dibedakan dari jenis jaringan lainnya berdasarkan tiga karakteriskomputer: ukuran, teknologi transmisi dan topologinya. (Herlina Latipa Sari, Aji Sudarsono dan B.Herawan Hayadi, 2013).

## 2.15 Web Server

*Web server* merupakan *software* yang memberikan layanan data, berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML, konsep *web server* antara lain:

1. *Web server* merupakan mesin aplikasi atau *software* yang beroperasi dalam mendistribusikan *web page ke user*, tentu saja sesuai dengan permintaan *user*.
2. Hubungan antara *web server dan browser internet* merupakan gabungan atau jaringan komputer yang berada diseluruh dunia. Setelah terhubung secara fisik, *protocol TCP/IP (networking protocol)* yang memungkinkan semua komputer dapat berkomunikasi antar satu dengan lainnya. (Muhammad Andang Novianta dan Emy Setyaningsih, 2015).

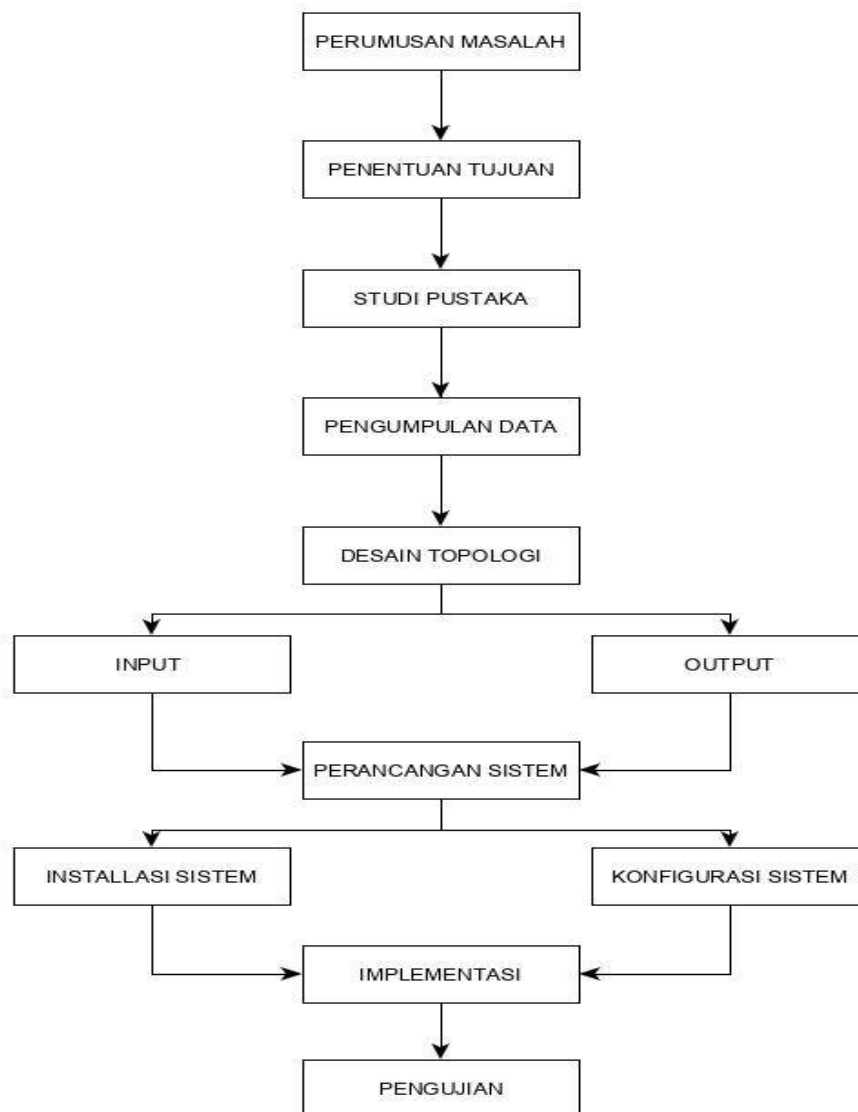
Seperti telah dijelaskan diatas, format data pada *world wide web* adalah SGML. Tapi para pengguna internet saat ini lebih banyak menggunakan format HTML (*hypertext markup language*) karena penggunaannya lebih sederhana dan mudah dipelajari. Standarisasi *web server* dalam penerapan penggunaannya antara lain dikeluarkan oleh W3C (*World Wide Web Consortium*), IETF (*Internet Engineering Task Force*), dan beberapa organisasi lainnya. Sampai saat ini, sudah lebih dari 110 spesifikasi yang dirilis oleh W3C (*W3C Recommendations*).

Contoh standarisasi *web server* antara lain : (Evy Nurmiati, 2012).

1. Spesifikasi HTML, CSS, DOM, XHTML (W3C) dan Spesifikasi *Javascript* (ECMA).

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

**3.1 Tahapan Penelitian**



**Gambar 3.1** Diagram Tahapan Penelitian

Dari diagram diatas dapat disimpulkan alurnya sebagai berikut :

1. *Perumusan masalah* adalah merumuskan sebuah masalah untuk mencari solusi yang tepat terhadap masalah tersebut.
2. *Penentuan tujuan* adalah menentukan tujuan sistem apa yg akan di bangun atas dasar masalah yang ada.
3. *Studi pustaka* adalah mencari sumber referensi yang cocok sesuai sistem yang akan di bangun.
4. *Pengumpulan data* adalah mengumpulkan data-data yang akan digunakan sebagai penunjang pembuatan sistem.
5. *Desain topologi* adalah mendesain topologi jaringan yang akan digunakan dalam pengimplementasian sistem.
6. *Input dan Output* adalah mencari masukan dari luar ataupun dalam untuk membuat sistem yang akan dibangun.
7. *Perancangan sistem* adalah merancang sistem sesuai kebutuhan yang diperlukan, yang bersumber dari masalah yang telah dirumuskan.
8. *Installasi dan konfigurasi sistem* adalah menginstall dan mengkonfigurasi sistem sesuai kinerja yang di inginkan.
9. *Implementasi* adalah mengimplemantasikan sistem yang sudah terbangun.
10. *Pengujiaan* adalah menguji coba dan menjalankan sistem secara lengkap dan rinci untuk melihat hasil secara keseluruhan.



### 3.2 Analisis Masalah

Perkembangan teknologi dari zaman ke zaman semakin meningkat, khususnya bagi para pengguna jaringan internet. Tentunya hal ini menjadi suatu masalah, karena kebutuhan manusia akan internet sudah tidak terkendali dan terpisahkan dalam kehidupan masyarakat. Dengan begitu pesatnya penggunaan *internet* di masyarakat maka perlu adanya sistem keamanan yang kuat untuk membuat para pengguna *internet* merasa aman dan nyaman.

Keamanan jaringan *internet* dari waktu ke waktu juga semakin meningkat seiring semakin banyaknya juga celah yang bisa disusupi oleh penyerang, dan tentunya hal ini menjadi suatu masalah karena serangan yang dilakukan bisa menyebabkan kerugian yang cukup besar jika tidak ada penanganan atas masalah keamanan tersebut. Begitu banyak sistem keamanan jaringan internet yang ada pada saat ini salah satunya sistem keamanan pada jaringan *wireless*, jaringan *wireless* merupakan jaringan yang menggunakan gelombang radio untuk saling terhubung satu sama lain. Oleh sebab itu dibutuhkan sebuah sistem keamanan jaringan yang kuat salah satunya yaitu *Intrusion Prevention System (IPS)*. *Intrusion Prevention System (IPS)* adalah sebuah sistem keamanan jaringan yang mengkombinasikan teknik *firewall* dan metode *Intrusion Detection System (IDS)* dengan sangat baik untuk mencegah sebuah serangan, salah satu serangan yang membahayakan jaringan *wireless* yaitu *brute force attack*.

*Bruteforce* merupakan salah satu serangan yang banyak digunakan pada jaringan yang menggunakan sistem keamanan *password* seperti *WPA*, *WPE2K*, dan *WPS*. *Bruteforce* menggunakan metode *matchmaking list* atau pencocokkan

kata hingga menemukan *password* yang dicari. Karena itu dibutuhkan keamanan jaringan yang mempuni untuk mengatasi serangan *brute force* ini yaitu *Intrusion Prevention System (IPS)*. Karena dengan *IPS* serangan *brute force* dapat dicegah atau di blok dengan mudah, sebab serangan ini sangat merugikan bagi para pengguna jaringan *wireless*.

### **3. 2.1 Perangkat yang Digunakan untuk Mengimplementasikan *Intrusion Prevention Sytem (IPS)* pada Jaringan *Wireless*.**

Dalam penerapan *IPS* sebagai sistem keamanan jaringan *wireless*, untuk mencegah serangan *brute force attack*, maka membutuhkan beberapa perangkat pendukung agar yang dibuat dapat berjalan dengan baik sesuai dengan perencanaan. Adapun perangkat yang digunakan terbagi menjadi dua bagian, sebagai berikut:

#### **1. Perangkat Keras (*Hardware*)**

- a. Laptop *Acer Aspire E 14 – 475G* dengan *processor intel core i3* (sebagai *IPS*)
- b. *Wireless router TP-LINK TL-WR840N*
- c. Sebuah laptop sebagai penyerang.

#### **2. Perangkat Lunak (*Software*)**

- a. *Linux Ubuntu 16.4* yang digunakan sebagai *Operating System* dalam menjalankan aplikasi *snort* dan *server*.
- b. *snort*, berfungsi sebagai aplikasi yang menjalankan *Intrusion Prevention System* .
- c. *Windows 10*.

- d. *Windows 10*, berfungsi sebagai penyerang, untuk mengetes tingkat keamanan *IPS*.

### 3. 2.2 Teknik Pemecahan Masalah

Pada Implementasi *Intrusion Prevention System* sebagai sistem keamanan jaringan *wireless* ini mempunyai beberapa poin teknik dalam pemecahannya, sebagai berikut:

1. Untuk langkah awal dalam perancangan sistem keamanan ini pertama perancang membangun *server* di *Ubuntu 16.4* sebagai pusat jaringan yang akan dibangun.
2. Mengkonfigurasi *server* yang telah dibangun dan menghubungkannya dengan *snort*.
3. Memasang *Intrusion Detection System* pada *snort* untuk mendeteksi dan mengawasi IP yang masuk.
4. Membangun *Intrusion Prevention System* pada *snort* di *Ubuntu 16.4*.
5. Menghubungkan *Intrusion Prevention System* / *snort* dengan *router* untuk memblokir serangan yang masuk ke dalam topologi jaringan yang dibangun.
6. Perancangan ini membutuhkan perangkat keras maupun perangkat lunak dalam membangun sistem keamanan yang berbasis *IPS* ini hingga berjalan sesuai dengan rancangan.
7. Kemudian pada proses pengujian ini akan dilakukan secara keseluruhan termasuk saat jaringan *wireless* tanpa *IPS* maupun setelah dipasang *IPS*, dimana bertujuan untuk mengetahui apakah perancangan sistem sudah sesuai dengan rencana yang sebelumnya dibuat.

### 3.3. Konsep Pengamanan Jaringan *Wireless* Menggunakan Metode *Intrusion Prevention System* .

Konsep yang digunakan untuk pengamanan jaringan *wireless* dari serangan *bruteforce attack* adalah dengan menggunakan *Intrusion Prevention System* sebagai media untuk memblokir serangan dan *intrusion detection system* yang berguna untuk mendeteksi serangan yang masuk. Metode *Intrusion Prevention System* ini menggunakan sebuah *software*, yaitu *snort*. *Software* ini membantu untuk memberikan *alert* saat ada aktifitas yang tidak sesuai dengan *rules* yang akan dibuat.

Pengamanan jaringan ini juga menggunakan *barnyard2* sebagai *output* yang digunakan pada *snort*, *rules* yang akan digunakan yaitu memberikan *alert* saat ada packet yang masuk dan IP yang mencurigakan dan untuk memonitoring aktifitas dari *alert* tersebut akan menggunakan *base* sebagai *web monitoring* dari *snort* tersebut.

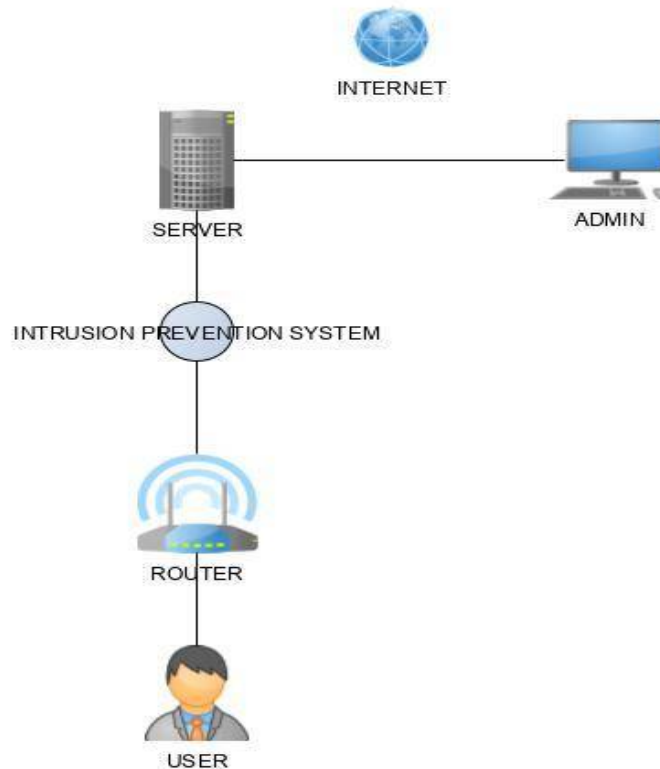
#### 3.3.1 Skema Kinerja *Intrusion Prevention System (IPS)*

*Intrusion Prevention System (IPS)* merupakan sebuah metode keamanan yang berfungsi untuk mengamankan sebuah jaringan komputer, *IPS* berjalan pada aplikasi *snort*, *IPS* bekerja dengan cara membaca serangan dan kemudian memblokir serangan yang masuk. *IPS* merupakan gabungan dari *IDS* dan *firewall* yang membuat sistem keamanan ini sangat kuat.

Untuk membangun sebuah sistem keamanan menggunakan metode *Intrusion Prevention System* , di sini saya sebagai penulis menggunakan sistem operasi *Linux Ubuntu 16.04*. Sistem operasi ini dipilih karena lebih mudah digunakan dan di dalam sistem operasi *Linux Ubuntu* sudah tersedia aplikasi *snort*, *snort* di sistem operasi Ubuntu tinggal diaktifkan saja melalui terminal karena masih berbasis *command-line* sehingga memudahkan penulis untuk membangun sistem keamanan jaringan *wireless*.

### **3.3.2 Topologi Jaringan sistem keamanan dengan metode *Intrusion Prevention System (IPS)***

Topologi jaringan adalah penyusunan jaringan yang diambil dari beberapa komponen yang ada kaitannya didalam jaringan tersebut, *Intrusion Prevention System* juga memiliki topologi jaringan sendiri yang membuat semua komponen saling terhubung dan bisa di akses oleh *user*, berikut adalah gambar dari topologi jaringan *Intrusion Prevention System (IPS)* yang dibangun:



**Gambar 3.2** Topologi Jaringan Sistem Keamanan Menggunakan *IPS*.

### 3. 4. Membangun *Network Server*

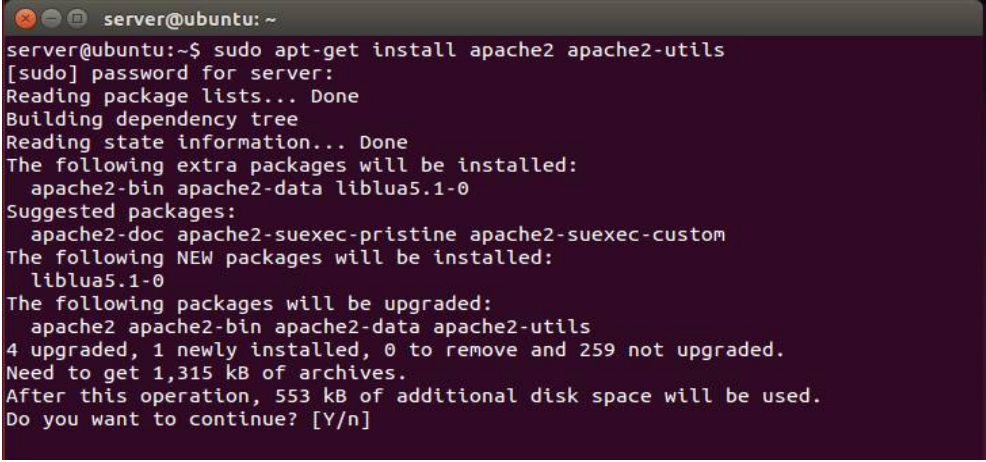
*Network server* merupakan sebuah sistem komputer yang menyediakan (*service*) tertentu dalam sebuah jaringan komputer. Pada *server* yang akan dibangun , *server* menyediakan layanan *HTTP*, *SSH*, *FTP*, *DNS*, *TELNET* untuk mendukung sistem keamanan yang akan dijalankan.

#### 3. 4.1 Konfigurasi *Server*

Untuk menjalankan sebuah *server* perlu dilakukan konfigurasi *penginstallan* layanan yang akan digunakan dalam sebuah *server*. Berikut layanan yang harus di *install* pada *server*.

## 1. *Instalasi HTTP*

Buka terminal pada Ubuntu 16.04 lalu *install HTTP* dengan perintah *sudo apt-get install apache2 apache2-utils*. Kemudian masukan password seperti yang diminta, lalu akan muncul pertanyaan *Do you want continue? [Y/n]* pilih Y.



```
server@ubuntu: ~
server@ubuntu:~$ sudo apt-get install apache2 apache2-utils
[sudo] password for server:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data liblua5.1-0
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom
The following NEW packages will be installed:
  liblua5.1-0
The following packages will be upgraded:
  apache2 apache2-bin apache2-data apache2-utils
4 upgraded, 1 newly installed, 0 to remove and 259 not upgraded.
Need to get 1,315 kB of archives.
After this operation, 553 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

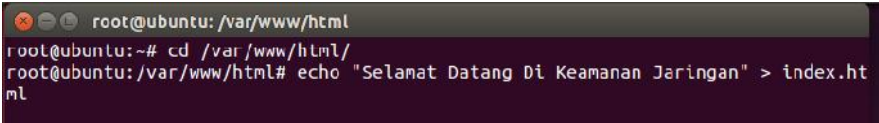
**Gambar 3.3** *Instalasi HTTP*

Di *HTTP* ada penambahan yaitu mengaktifkan *HTTP Basic Authentication*, dan juga membuat *HTTP credentials* (kita akan menggunakan username = cahyo dan password = 123cahyo)

Adapun langkah – langkahnya :

- a. Ubah tampilan dari *Web* sesuai keinginan.

Untuk mengubah tampilan *Web server* masuk ke dalam *system root* dan masukan perintah *cd /var/www/html/* dan masukan perintah *echo "selamat datang di keamanan jaringan" > index.html*. Berikut penerapannya :



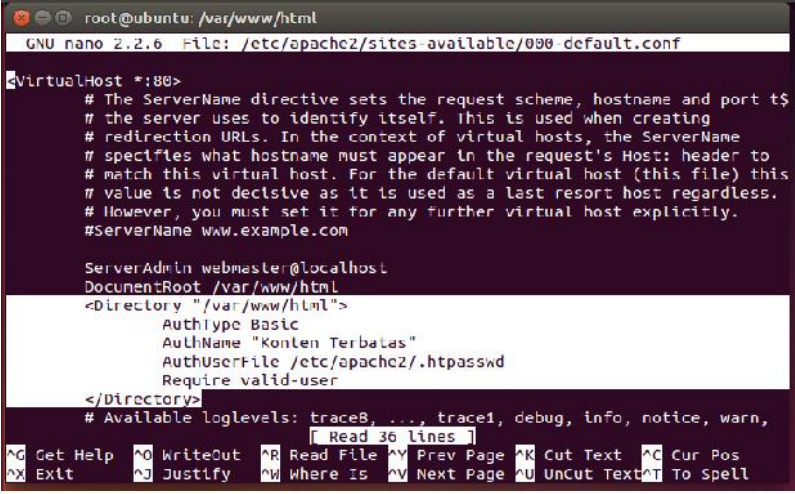
```
root@ubuntu: /var/www/html
root@ubuntu:~# cd /var/www/html/
root@ubuntu:/var/www/html# echo "Selamat Datang Di Keamanan Jaringan" > index.html
```

**Gambar 3.4** *Penambahan Sintaks Untuk Tampilan Web Server*

- b. Ubah konfigurasi *sites apache2*.

Masukan perintah untuk merubah konfigurasi *sites pache2* : `nano /etc/apache2/sites-available/000-default.conf`.

Kemudian tambahkan isi konfigurasi tersebut dengan *rules/sintaks* sesuai dengan *screenshot block* berwarna putih.



```

root@ubuntu: /var/www/html
GNU nano 2.2.6 file: /etc/apache2/sites-available/000-default.conf
VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

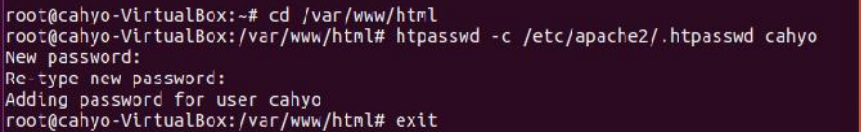
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
<Directory "/var/www/html">
    AuthType Basic
    AuthName "KonLen Terbatas"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Directory>
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# Read 36 Lines
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U Uncut Text ^T To Spell

```

**Gambar 3.5** Konfigurasi *Rules Apache2*.

- c. Buat *username* dan *password* untuk *HTTP*

Setelah itu buat *username* dan *password* untuk *HTTP*, dengan memasukan perintah `cd /var/www/html` untuk masuk kedalam *directory*. Kemudian lanjut keperintah `htpasswd -c /etc/apache2/.htpasswd cahyo` kemudian masukan *password* dan selesai.



```

root@cahyo-VirtualBox:~# cd /var/www/html
root@cahyo-VirtualBox:/var/www/html# htpasswd -c /etc/apache2/.htpasswd cahyo
New password:
Re-type new password:
Adding password for user cahyo
root@cahyo-VirtualBox:/var/www/html# exit

```

**Gambar 3.6** Konfigurasi *Username Or Password Apache2*.



- d. Buka *web browser* dan jalankan *HTTP*

Untuk melihat hasil dari *web server* yang telah dibuat tinggal buka *browser* dan kemudian masukan alamat *ip address server* yaitu *192.168.137.2*.



**Gambar 3.7** Tampilan *Web Server*

## 2. *Instalasi SSH*

*Sudo apt-get install ssh-server*. Merupakan sintaks untuk menginstall *ssh-server*.

```
server@ubuntu:~$ sudo apt-get install ssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package ssh-server is a virtual package provided by:
  openssh-server:1386 1:6.6p1-2ubuntu2.8
  dropbear 2013.60-1ubuntu2.1
  openssh-server 1:6.6p1-2ubuntu2.8
  lsh-server 2.1-1ubuntu1
You should explicitly select one to install.
```

**Gambar 3.8** Penginstallan *SSH*

## 3. *Instalasi FTP*

*Sudo apt-get install vsftpd*. Merupakan sintaks untuk menginstall *FTP-server*.

```
server@ubuntu:~$ sudo apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
vsftpd is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 263 not upgraded.
server@ubuntu:~$
```

**Gambar 3.9** Penginstallan *FTP*

#### 4. *Instalasi DNS*

*Sudo apt-get install bind9.* Merupakan sintaks untuk menginstall *DNS-server*.

```
server@ubuntu:~$ sudo apt-get install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  bind9-host bind9utils dnsutils libbind9-90 libdns100 libisc95 libisccc90
  libisccfg90 liblwres90
Suggested packages:
  bind9-doc rblcheck
The following NEW packages will be installed:
  bind9 bind9utils
The following packages will be upgraded:
  bind9-host dnsutils libbind9-90 libdns100 libisc95 libisccc90 libisccfg90
  liblwres90
8 upgraded, 2 newly installed, 0 to remove and 255 not upgraded.
Need to get 1,475 kB of archives.
After this operation, 1,637 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

**Gambar 3.10** Penginstallan *DNS*

#### 5. *Instalasi TELNET*

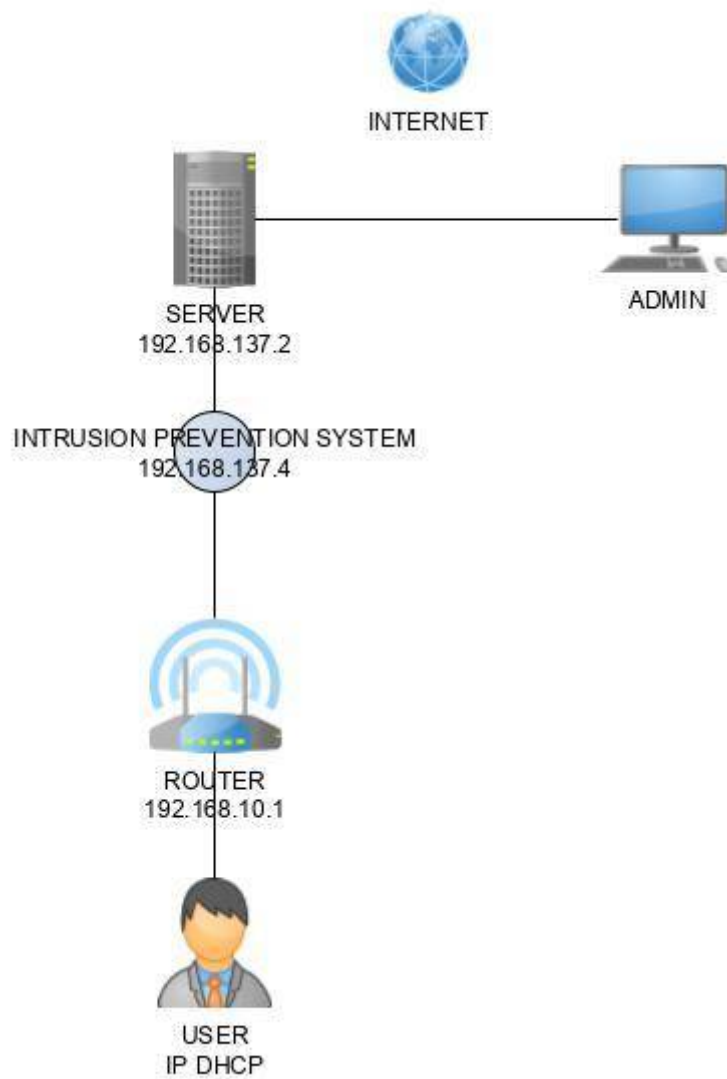
*Sudo apt-get install telnetd.* Merupakan *sintaks* untuk *menginstall* *TELNET* pada *server*.

```
server@ubuntu:~$ sudo apt-get install telnetd
Reading package lists... Done
Building dependency tree
Reading state information... Done
telnetd is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 263 not upgraded.
server@ubuntu:~$
```

**Gambar 3.11** Penginstallan *TELNET*

### 3. 4.2 Topologi Jaringan *Server*

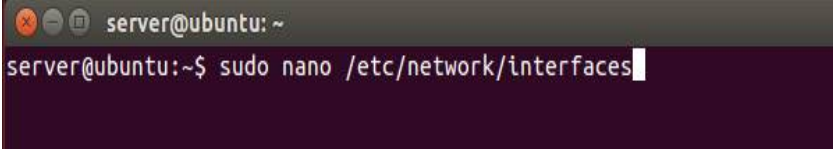
Topologi jaringan adalah penyusunan jaringan yang diambil dari komponen-komponen yang terkait didalam jaringan tersebut, *server* juga memiliki topologi jaringan sendiri yang membuat semua komponen saling terhubung dan bisa di akses oleh *user*, berikut adalah gambar dari topologi jaringan *server* yang dibangun:



**Gambar 3.12** Topologi Jaringan Server

Setelah kebutuhan layanan yang ingin dibangun telah di *install* kedalam *server*, maka selanjutnya dilakukan pengalamatan *IP address* untuk *server* supaya bisa saling terhubung kedalam 1 jaringan yang menggunakan sistem keamanan *IPS*. Untuk mengkonfigurasi *IP address* pada *server* diperlukan perintah sebagai berikut :

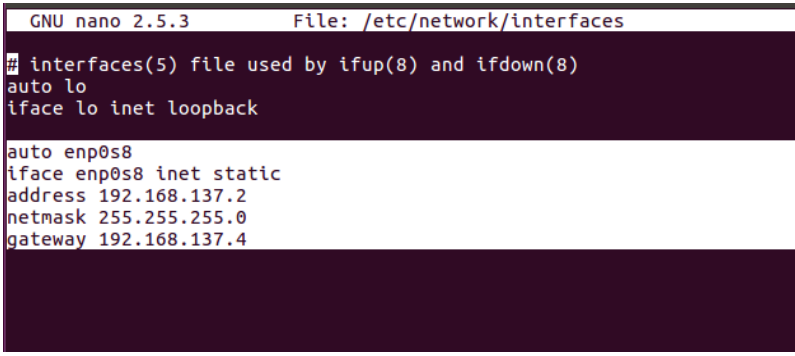
1. *Sudo nano /etc/network/interfaces*



```
server@ubuntu: ~
server@ubuntu:~$ sudo nano /etc/network/interfaces
```

**Gambar 3.13** Konfigurasi IP Address

2. Kemudian masukan *rules* digambar yang di *block* putih



```
GNU nano 2.5.3      File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto enp0s8
iface enp0s8 inet static
address 192.168.137.2
netmask 255.255.255.0
gateway 192.168.137.4
```

**Gambar 3.14** Konfigurasi IP Address

Pada gambar diatas *server* menggunakan *IP address* 192.168.137.2 , *netmask* 255.255.255.0 dan *gateway* 192.168.137.4. *Gateway* disitu merupakan *IP address* dari *snort*.

### 3.5 Membangun *Intrusion Prevention System (IPS)*

Dalam pembangunan *IPS* digunakan *software* tambahan yaitu *snort*, untuk mengaktifkan *IPS* itu sendiri maka akan ditambahkan *rules* baru pada perintah */etc/snort/snort.conf* pada *snort*. *Snort* sendiri sudah ada dalam repository sistem operasi Ubuntu 16.04, karena itu *snort* dapat di install dengan mudah.

Sebelum menginstall *snort* diperlukan *library* pendukung untuk menjalankan *snort* nantinya. Yaitu bisa dengan menggunakan perintah “*apt-get*

*install -y flex bison build-essential checkinstall libpcap-dev libnet1-dev libnetfilter-queue-dev libpcr3-dev libmysqlclient-dev iptables-dev libnet-dev”*

```

snort@ubuntu:~$ sudo apt-get install flex bison build-essential checkinstall libpcap-dev libnet1-dev libpcr3-dev libmysqlclient-dev libnetfilter-queue-dev iptables-dev libnet-dev
[sudo] password for snort:
Reading package lists... Done
Building dependency tree
Reading state information... Done
bison is already the newest version.
build-essential is already the newest version.
iptables-dev is already the newest version.
libnet1-dev is already the newest version.
libpcap-dev is already the newest version.
checkinstall is already the newest version.
libnet-dev is already the newest version.
libnetfilter-queue-dev is already the newest version.
libmysqlclient-dev is already the newest version.
libpcr3-dev is already the newest version.
The following extra packages will be installed:
  libfl-dev
The following packages will be upgraded:
  flex libfl-dev
2 upgraded, 0 newly installed, 0 to remove and 247 not upgraded.
1 not fully installed or removed.
Need to get 266 kB of archives.

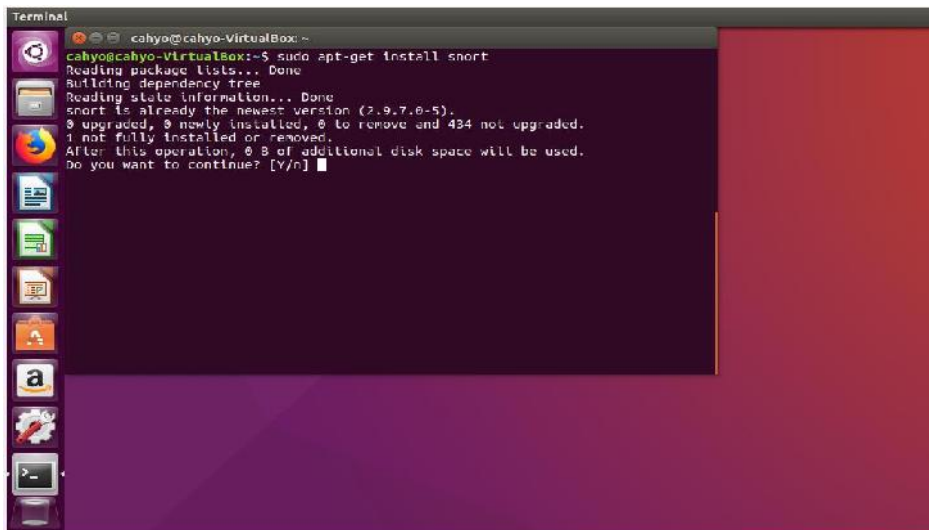
```

**Gambar 3.15** Peningstallan Library Pendukung

### 3.5.1 *Instalasi Snort*

Ada beberapa langkah yang harus dilakukan untuk menginstall *snort* di sistem operasi Linux Ubuntu 16.04, yaitu sebagai berikut :

1. Buka terminal pada *Ubuntu 16.04* lalu *install snort* dengan perintah *sudo apt-get install snort*. Kemudian masukan password seperti yang diminta, lalu akan muncul pertanyaan *Do you want continue? [Y/n]* pilih Y.



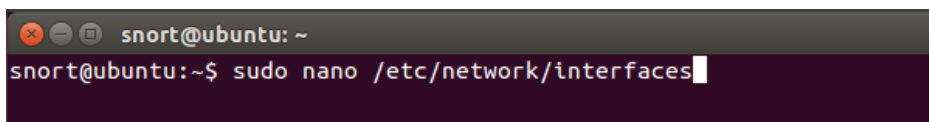
```

Terminal
cahyo@cahyo-VirtualBox:~$ sudo apt-get install snort
Reading package lists... Done
Building dependency tree
Reading state information... Done
snort is already the newest version (2.9.7.0-5).
0 upgraded, 0 newly installed, 0 to remove and 434 not upgraded.
1 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n]

```

Gambar 3.16 penginstalan snort

2. Kemudian konfigurasi *network interfaces* pada *snort*.



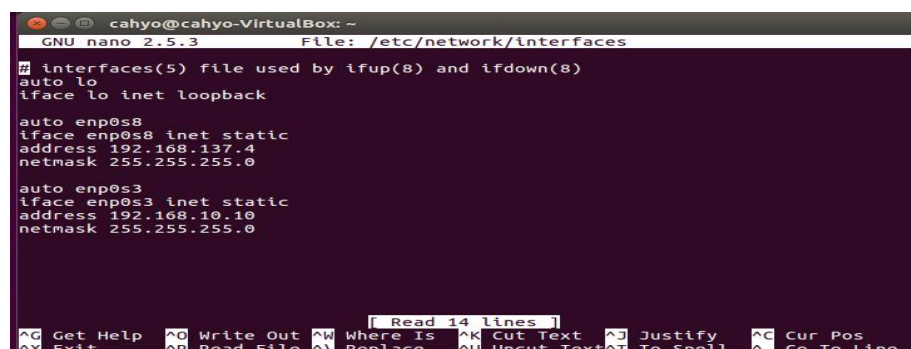
```

snort@ubuntu:~$ sudo nano /etc/network/interfaces

```

Gambar 3.17 Konfigurasi Jaringan Snort

Kemudian akan muncul jendela *interfaces* pada *snort*. Pilih interface yang digunakan, atur *IP static* sesuai dengan keinginan, dan juga pastikan *IP netmask* dan *gateway* berada dalam 1 jaringan dengan *server* dan *router* karena *snort* memakai 2 *adapter* yang akan saling terhubung 1 dengan lainnya. Seperti gambar berikut :



```

GNU nano 2.5.3 File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto enp0s8
iface enp0s8 inet static
address 192.168.137.4
netmask 255.255.255.0

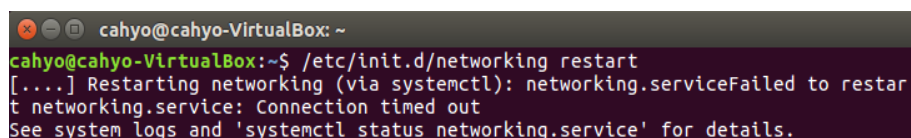
auto enp0s3
iface enp0s3 inet static
address 192.168.10.10
netmask 255.255.255.0

```

Gambar 3.18 Konfigurasi Jaringan Snort

Pada gambar diatas *interfaces enp0s8* merupakan *interfaces* yang terhubung ke *server* dengan *IP Address 192.168.137.4* dan *interfaces enp0s3* merupakan *interfaces* yang terhubung ke *router TP-LINK*.

Setelah *menkonfigurasi network interfaces*, kemudian *restart interfaces* tersebut dengan perintah `/etc/init.d/networking restart`.

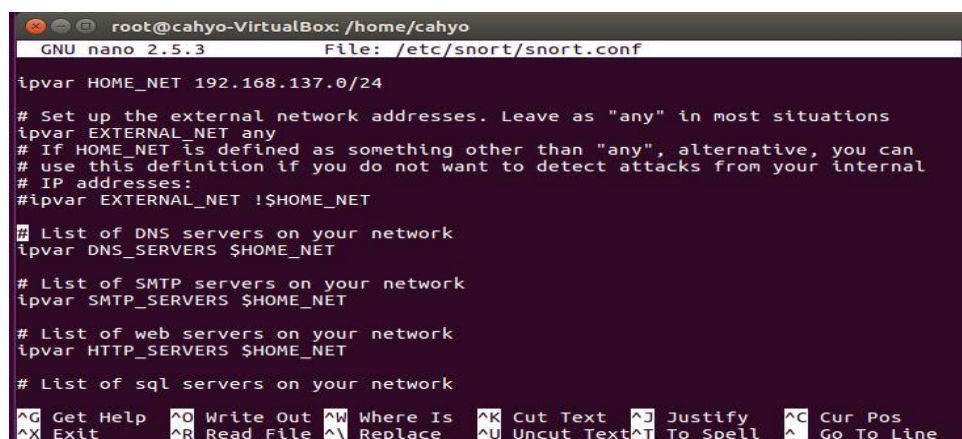


```
cahyo@cahyo-VirtualBox: ~
cahyo@cahyo-VirtualBox:~$ /etc/init.d/networking restart
[...] Restarting networking (via systemctl): networking.serviceFailed to restart
networking.service: Connection timed out
See system logs and 'systemctl status networking.service' for details.
```

**Gambar 3.19** Restart Jaringan Snort

### 3.5.2 Konfigurasi snort

Untuk konfigurasi *snort*, pertama perlu mengubah pengaturannya dibagian `/etc/snort/snort.conf` untuk dan mengubah *ipvar HOME\_NET any* menjadi *ipvar HOME\_NET 192.168.137.0/24* dibagian itu diubah untuk membuat *network address* yang ingin dilindungi. Tetapi dibagian itu bisa saja dibiarkan *any* untuk menyesuaikan dengan situasi yang ada. Seperti dibawah ini:



```
root@cahyo-VirtualBox: /home/cahyo
GNU nano 2.5.3 File: /etc/snort/snort.conf
ipvar HOME_NET 192.168.137.0/24

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

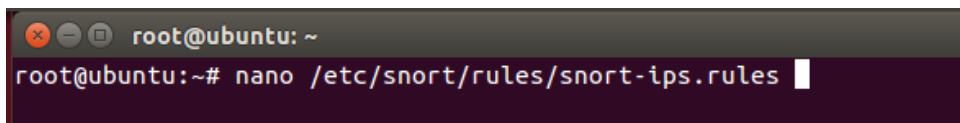
# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

# List of sql servers on your network

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

**Gambar 3.20** Konfigurasi Snort IPS

Kemudian kita juga perlu mengatur *rules snort* agar bisa berjalan sesuai dengan yang diinginkan, untuk konfigurasi *file rulesnya* berada di */etc/snort/rules/local.rules*.



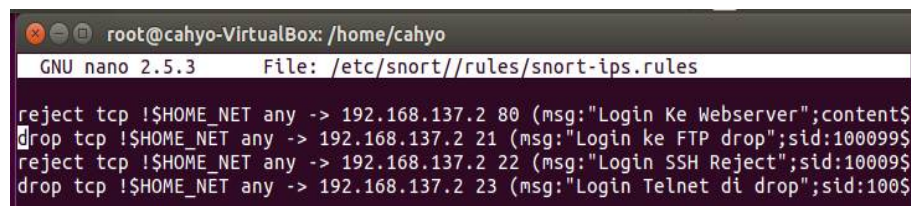
```

root@ubuntu: ~
root@ubuntu:~# nano /etc/snort/rules/snort-ips.rules

```

**Gambar 3.21** Konfigurasi Rules Intrusion Prevention System

Disana bisa ditambahkan beberapa *rules* seperti beberapa *rules* dibawah ini:



```

root@cahyo-VirtualBox: /home/cahyo
GNU nano 2.5.3 File: /etc/snort//rules/snort-ips.rules
reject tcp !$HOME_NET any -> 192.168.137.2 80 (msg:"Login Ke Webserver";content$
drop tcp !$HOME_NET any -> 192.168.137.2 21 (msg:"Login ke FTP drop";sid:100099$
reject tcp !$HOME_NET any -> 192.168.137.2 22 (msg:"Login SSH Reject";sid:10009$
drop tcp !$HOME_NET any -> 192.168.137.2 23 (msg:"Login Telnet di drop";sid:100$

```

**Gambar 3.22** Konfigurasi Rules Intrusion Prevention System

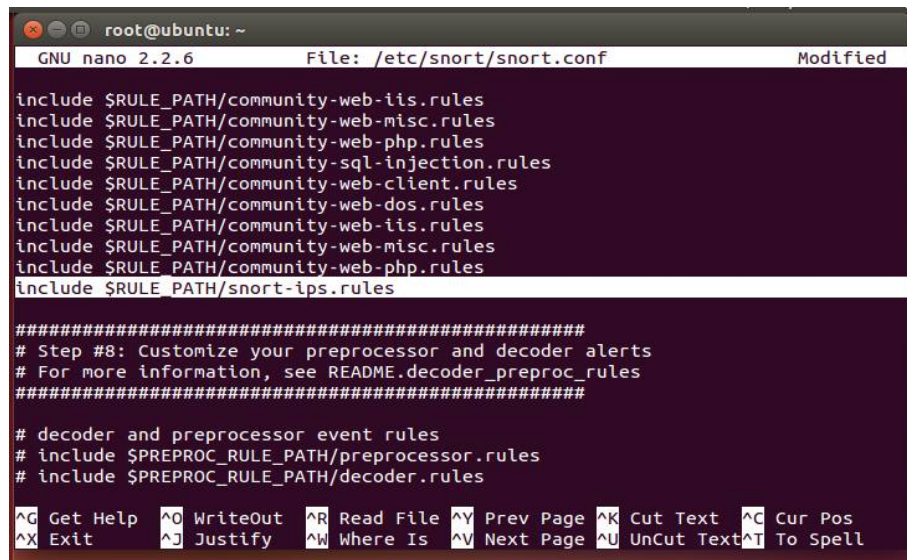
Dari setiap kata yang dituliskan *dirules* tersebut memiliki perintah sendiri mulai dari *rule header*, yaitu:

1. *alert* yaitu *rule action*, *Snort* akan memberikan peringatan ketika kondisi yang diatur sesuai.
2. *Any* yaitu *Source IP*, *Snort* akan melihat semua dari ip yang masuk.
3. *Any* yaitu *Sourc eport*, *Snort* akan melihat dari semua *port* yang masuk
4. *\$HOME\_NET* yaitu *Destination IP*, ini adalah *value* dari *filesnort.conf*
5. *any* yaitu *Destination port*, *Snort* akan mendeteksi semua *port* yang dilindunginya.

Kemudian *file snort-IPS.rules* ini di masukkan kedalam *file konfigurasi snort* dalam perintah *nano /etc/snort/snort.conf* , supaya disaat *snort* dijalankan,



maka *action* isi *file* ini ikut dijalankan. Ketika ada intruksi yang terdeteksi maka akan menjalankan isi *file rules* tersebut. Seperti gambar dibawah ini:



```

root@ubuntu: ~
GNU nano 2.2.6 File: /etc/snort/snort.conf Modified

include $RULE_PATH/community-web-iis.rules
include $RULE_PATH/community-web-misc.rules
include $RULE_PATH/community-web-php.rules
include $RULE_PATH/community-sql-injection.rules
include $RULE_PATH/community-web-client.rules
include $RULE_PATH/community-web-dos.rules
include $RULE_PATH/community-web-iis.rules
include $RULE_PATH/community-web-misc.rules
include $RULE_PATH/community-web-php.rules
include $RULE_PATH/snort-ips.rules

#####
# Step #8: Customize your preprocessor and decoder alerts
# For more information, see README.decoder_preproc_rules
#####

# decoder and preprocessor event rules
# include $PREPROC_RULE_PATH/preprocessor.rules
# include $PREPROC_RULE_PATH/decoder.rules

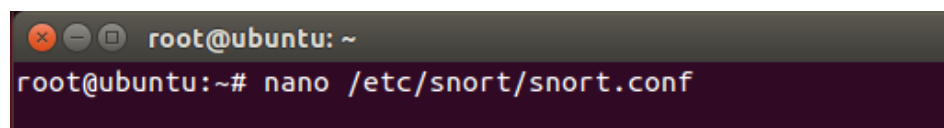
^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit          ^J Justify      ^W Where Is    ^V Next Page    ^U UnCut Text  ^T To Spell

```

**Gambar 3.23** Konfigurasi *Rules Intrusion Prevention System*

### 3. 5.3 Konfigurasi *Snort Inline* Menggunakan Metode *Afpacket*

*Snort inline* adalah sebuah model pengamanan yang menggunakan model pengaman satu jalur, dimana antara *server*, *snort*, dan *client* saling terhubung dalam satu sistem keamanan. Untuk merubah *snort* menjadi mode *IPS*, *snort* harus dijalankan pada *mode inline* dengan *data acquisition (DAQ)*. Untuk menambahkannya masuk kedalam perintah *nano /etc/snort/snort.conf*.



```

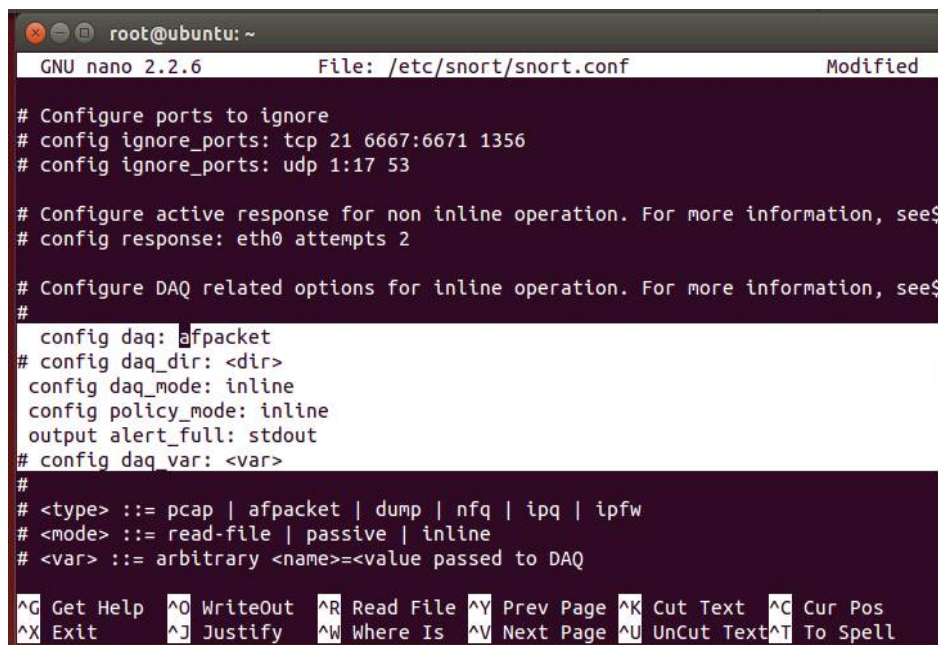
root@ubuntu: ~
root@ubuntu:~# nano /etc/snort/snort.conf

```

**Gambar 3.24** Konfigurasi *Snort Inline* Mode *Afpacket*

*DAQ* sendiri memiliki banyak tipe diantaranya *NFQ*, *IPQ*, *AFPACKET* dan *IPFW*. Masing – masing tipe ini merupakan skema penangkapan paket,

misalnya *NFQ* menggunakan *Queue* yang menggunakan antrian dan *rule iptables*, *AFPACKET* menggunakan skema *forward* paket dari satu *interface* ke *interface* lain (membutuhkan 2 *interface*). Disini saya menggunakan *AFPACKET* karena tidak perlu konfigurasi tambahan pada *iptables* tapi syaratnya harus menggunakan 2 *interface/ethernet card*. Lalu cari line code / baris sesuai dengan screenshot yang dibawah :



```

root@ubuntu: ~
GNU nano 2.2.6      File: /etc/snort/snort.conf      Modified

# Configure ports to ignore
# config ignore_ports: tcp 21 6667:6671 1356
# config ignore_ports: udp 1:17 53

# Configure active response for non inline operation. For more information, see$
# config response: eth0 attempts 2

# Configure DAQ related options for inline operation. For more information, see$
#
config daq: afpacket
# config daq_dir: <dir>
config daq_mode: inline
config policy_mode: inline
output alert_full: stdout
# config daq_var: <var>
#
# <type> ::= pcap | afpacket | dump | nfq | ipq | ipfw
# <mode> ::= read-file | passive | inline
# <var> ::= arbitrary <name>=<value passed to DAQ>

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell

```

**Gambar 3.25** Konfigurasi *Snort Inline Mode Afpacket*

Kemudian tambahkan perintah atau *rules* baru dengan *code* seperti diatas yang di *block* berwarna putih sebagai berikut :

*Config daq: afpacket*

*# config daq\_dir: <dir>*

*Config daq\_mode: inline*

*Config policy\_mode: inline*

*Output alert\_full: stdout*

*# config daq\_var: <var>*

Setelah perintah diatas telah ditambahkan kedalam *nano /etc/snort/snort.conf*, Konfigurasi *snort* sudah selesai dan *mode snort* sudah menjadi *IPS Inline mode*, sebelum *snort* dijalankan ketikan perintah *snort --daq-list*, dan pastikan *daq afpacket* ada seperti *screenshot* dibawah ini:

```

root@ubuntu: ~
root@ubuntu:~# 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@ubuntu:~#

```

**Gambar 3.26** Konfigurasi *Snort Inline Mode Afpacket*

Kemudian *snort* siap dijalankan. Untuk menjalankan *snort* dengan *mode afpacket inline mode* maka sintaks nya seperti berikut :

```

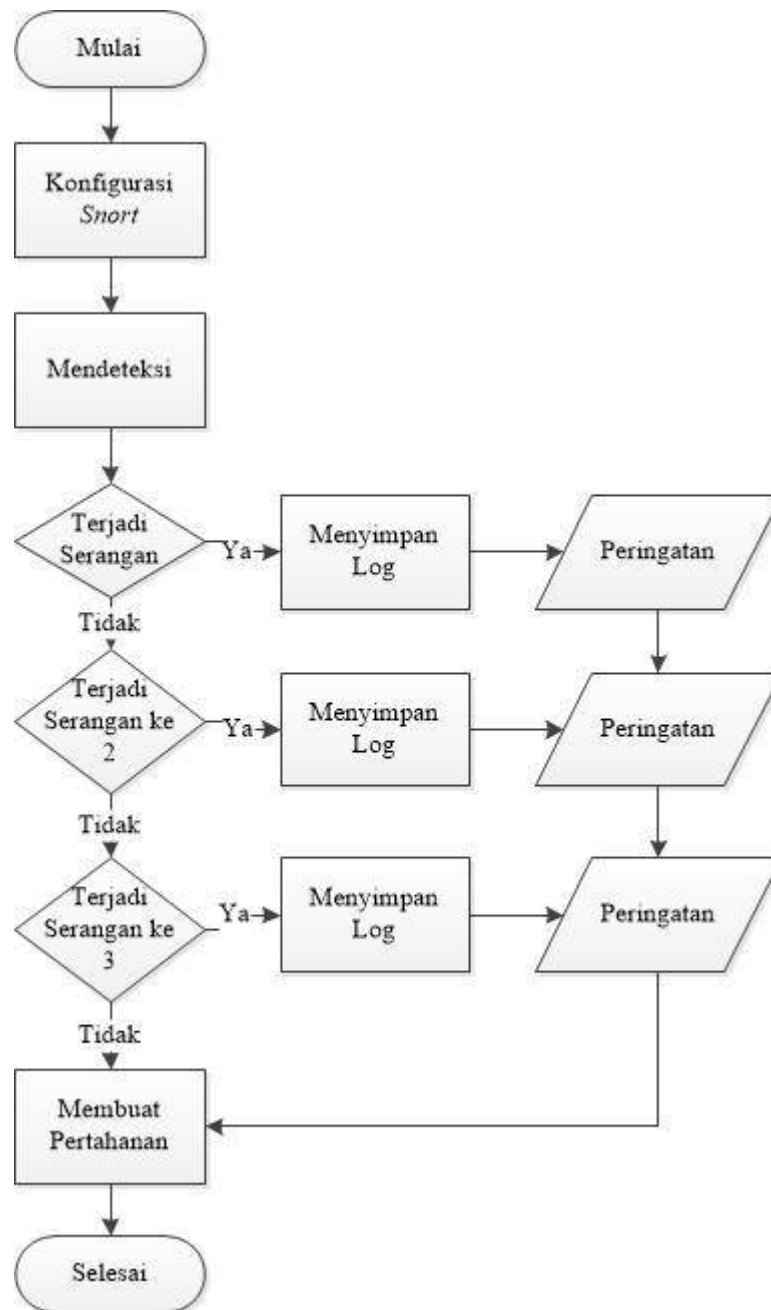
root@cahyo-VirtualBox: /home/cahyo
root@cahyo-VirtualBox:/home/cahyo# snort -c /etc/snort/snort.conf -i enp0s3:enp0s8 -Q > /var/log/snort/snort.log &

```

**Gambar 3.27** Menjalankan *Snort Inline Mode Afpacket*

*Snort inline* dengan menggunakan *mode Afpacket* merupakan *Intrusion Prevention System* , *Intrusion Prevention System* merupakan sistem keamanan yang menggunakan 2 *interfaces* yang dapat melindungi dari serangan yang ditujukan ke *server*.

### 3. 5.4 Flowchart Kinerja *Intrusion Prevention System*



**Gambar 3.28** Flowchart *Intrusion Prevention System*

### 3.6 Mesin Penyerang

Mesin penyerang atau sering disebut dengan *attacker* dalam sebuah jaringan merupakan sebuah ancaman yang nyata dalam sebuah jaringan komputer yang akan dibangun. Untuk mengantisipasi sebuah serangan yang akan mengancam sebuah topologi jaringan, maka diperlukan sebuah sistem keamanan yang sangat kuat untuk memblokir serangan yang masuk. Sistem keamanan dalam sebuah jaringan komputer berfungsi untuk mengamankan *server* dan *client* sehingga keduanya dapat menggunakan jaringan dengan aman dan nyaman.

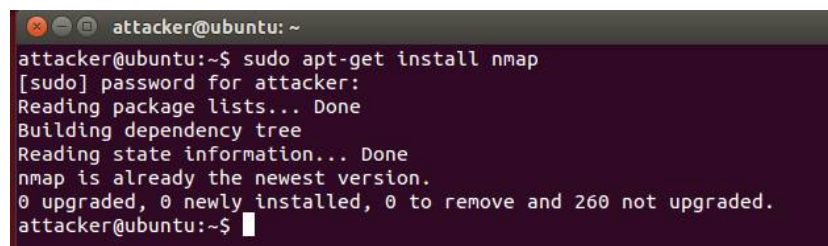
Sistem keamanan *Intrusion Prevention System* merupakan sistem keamanan yang mampu, membaca dan memblokir sebuah ancaman hanya dengan satu perintah. Oleh sebab itu, untuk menguji coba kemampuan sistem keamanan dengan menggunakan metode *IPS* diperlukan mesin penyerang. Mesin penyerang ini akan di *installasi* pada *operating system UBUNTU 16.0*. Dalam mesin penyerang akan dimasukkan jenis serangan yaitu *Brute Force Attack* dan *NMAP* serangan ini merupakan serangan yang bekerja dengan cara melakukan pengacakan *password* pada *server* sehingga akan merusak kinerja *server* itu sendiri. Berikut cara *installasi* mesin penyerang.

#### 3.6.1 Installasi Mesin Penyerang

Untuk *menginstall* mesin penyerang ini langkahnya hampir sama dengan cara *menginstall snort* seperti diatas, namun dalam *penginstallan* ini akan di tambah beberapa *tools hacking* yang diambil langsung dari *repository Ubuntu* itu sendiri. Seperti *brute force* dan *nmap* yang akan kita tambahkan. Caranya sebagai berikut

### 1. *Instalasi NMAP*

Masukan perintah pada *repository Ubuntu*, *sudo apt-get install nmap*. Perintah tersebut digunakan untuk *menginstall nmap* kedalam mesin penyerang. Contohnya sebagai berikut:

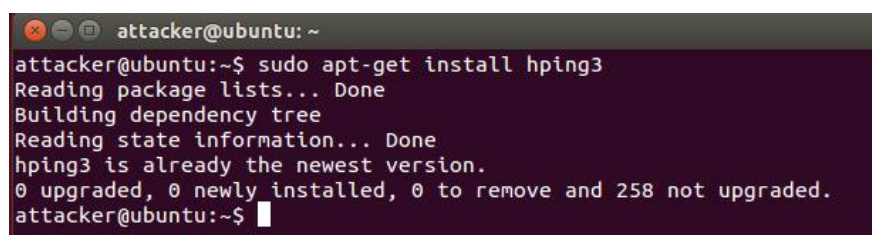
A terminal window with a dark purple background. The prompt is 'attacker@ubuntu: ~'. The user enters 'sudo apt-get install nmap'. The terminal shows the following output: '[sudo] password for attacker:', 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done', 'nmap is already the newest version.', and '0 upgraded, 0 newly installed, 0 to remove and 260 not upgraded.' The prompt returns to 'attacker@ubuntu:~\$' with a cursor.

```
attacker@ubuntu:~$ sudo apt-get install nmap
[sudo] password for attacker:
Reading package lists... Done
Building dependency tree
Reading state information... Done
nmap is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 260 not upgraded.
attacker@ubuntu:~$
```

**Gambar 3.29** *Instalasi NMAP*

### 2. *Instalasi Brute Force*

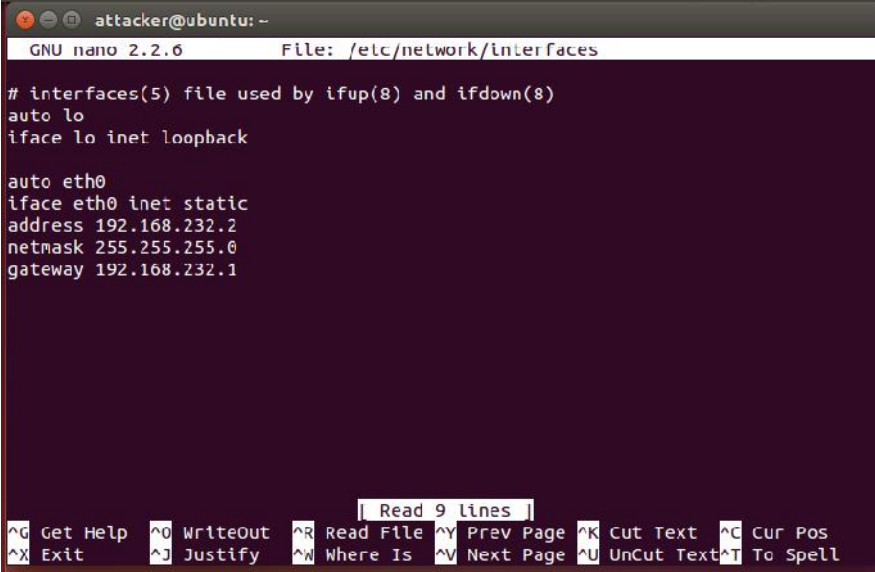
Masukan perintah pada *repository Ubuntu*, *sudo apt-get install hping3*. Perintah tersebut digunakan untuk *menginstall bruteforce* kedalam mesin penyerang. Contohnya sebagai berikut:

A terminal window with a dark purple background. The prompt is 'attacker@ubuntu: ~'. The user enters 'sudo apt-get install hping3'. The terminal shows the following output: 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done', 'hping3 is already the newest version.', and '0 upgraded, 0 newly installed, 0 to remove and 258 not upgraded.' The prompt returns to 'attacker@ubuntu:~\$' with a cursor.

```
attacker@ubuntu:~$ sudo apt-get install hping3
Reading package lists... Done
Building dependency tree
Reading state information... Done
hping3 is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 258 not upgraded.
attacker@ubuntu:~$
```

**Gambar 3.30** *Instalasi Bruteforce*

Setelah semua *tools hacking terinstall* kemudian masuk kedalam *nano /etc/network/interface* kemudian atur *IP Address* sesuai dengan alamat yang ingin di serang. Contohnya seperti berikut :



```

attacker@ubuntu: ~
GNU nano 2.2.6 File: /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.232.2
netmask 255.255.255.0
gateway 192.168.232.1

Read 9 lines |
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

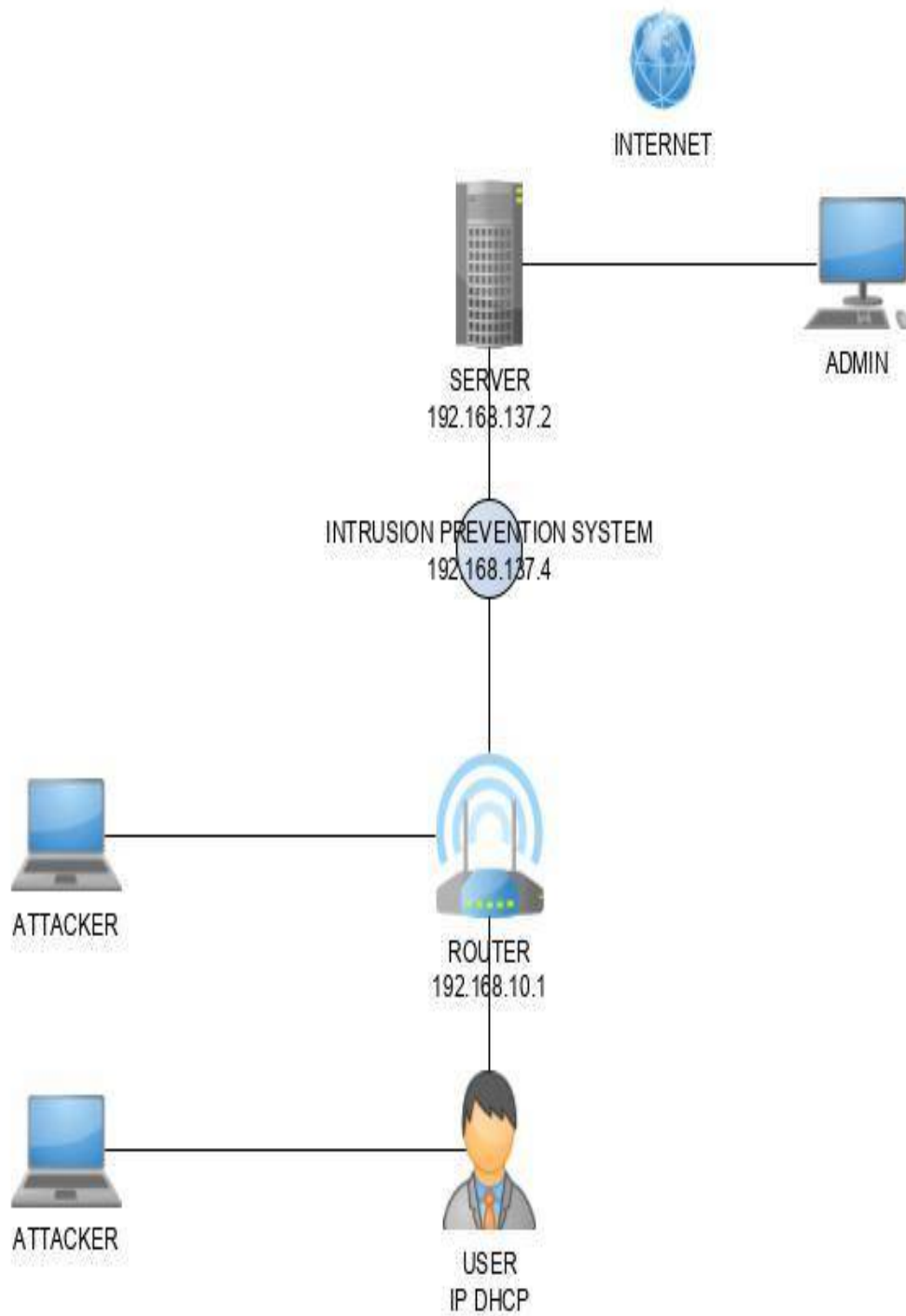
```

**Gambar 3.31** Konfigurasi *IP Address Attacker*

### 3. 6.2 Topologi Jaringan Penyerang

Percobaan ini di lakukan dengan 2 serangan atau *attacker* yang mencoba menyerang melalui *user* dan langsung melalui *router*. *PC1* menyerang melalui *router* dengan jenis serangan *DDOS attack* dan *PC2* menyerang melalui *user* menggunakan jenis serangan *brute force*.

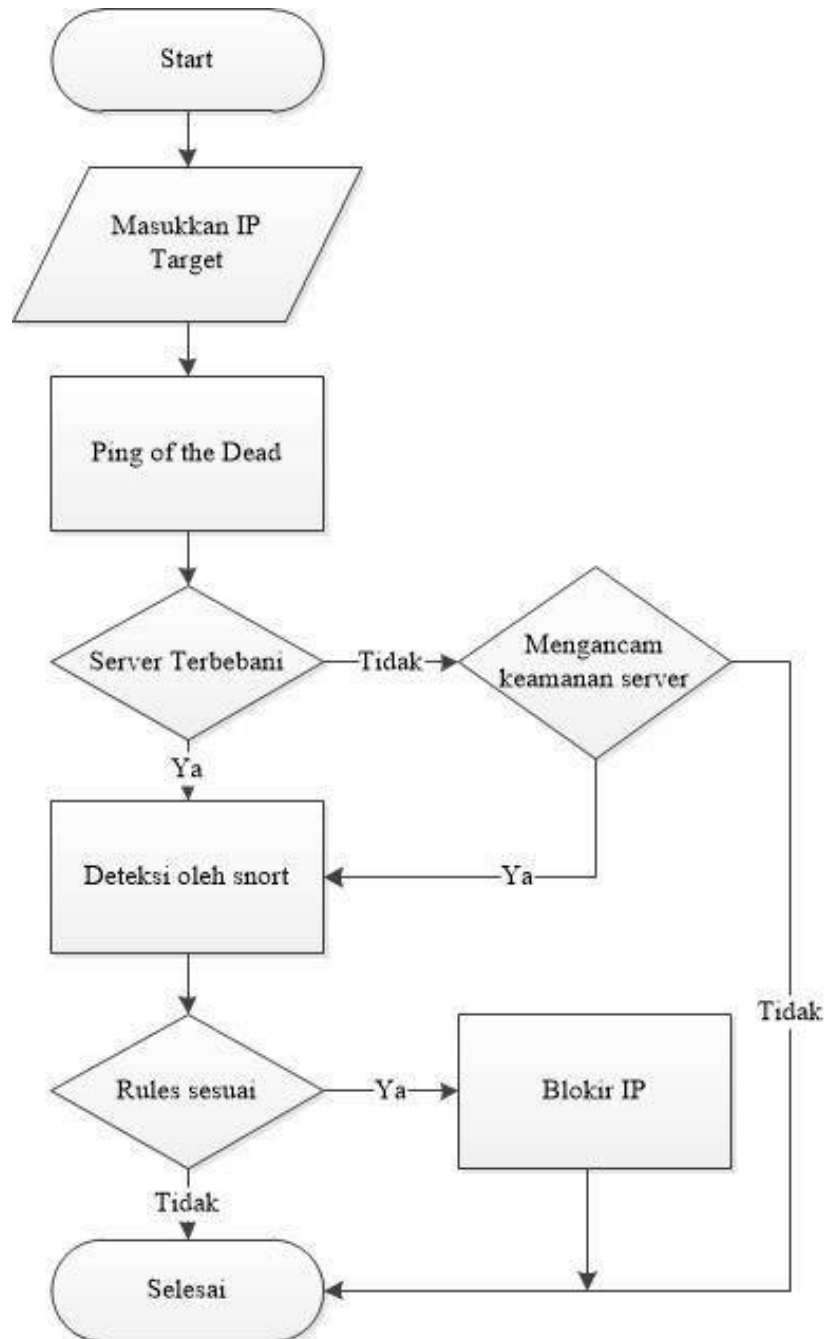
Masing-masing serangan memiliki kegunaannya tersendiri. *DDOS* digunakan untuk melemahkan sistem keamanan, sehingga *server* bisa diakses dengan mudah. Sedangkan *brute force* digunakan untuk mendapatkan *username* dan *password admin*, untuk bisa masuk kedalam *server* melalui *user client* yang telah terhubung kedalam jaringan yang sama.



**Gambar 3.32** Topologi Jaringan Penyerang



### 3. 6.3 Flowchart Kinerja Attacker



Gambar 3.33 Flowchart Penyerang

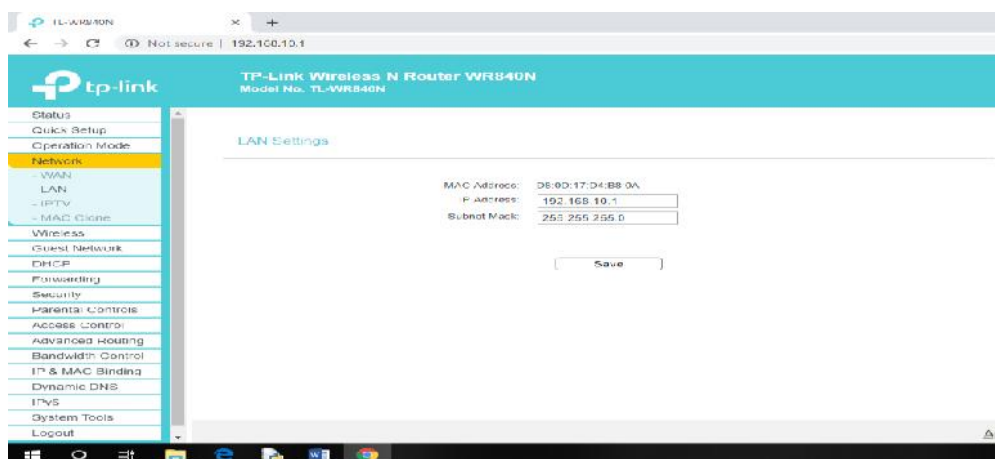
### 3.7 Konfigurasi Router TP-LINK WR840

*Router* merupakan sebuah perangkat yang bekerja sebagai pembagi paket-paket jaringan, atau disebut juga sebagai jembatan jaringan. *Router* membagikan jaringan dari *server* ke *client*, dan pada *router* yang saya gunakan ini pembagian jaringan akan dilakukan secara *wireless* atau sering disebut *jaringan nirkabel*.

Untuk mengkonfigurasi *router* ini, harus melalui *browser* yang tersedia dan memasukan *ip router* tersebut. *Ip router* ialah *192.168.10.1*. Berikut langkah-langkah konfigurasi *router TP-LINK WR840N*.

#### 3.7.1 Konfigurasi IP Address Router

Untuk menghubungkan sebuah jaringan harus memiliki sebuah kesatuan atau menghubungkan satu dengan yang lainnya. Karena itu *router* harus di *setting* supaya bisa terkoneksi dengan *server* dan dapat membagikannya ke *client*. Pertama *setting IP address pada LAN, dengan IP address 192.168.10.1 dan subnet mask 255.255.255.0*.

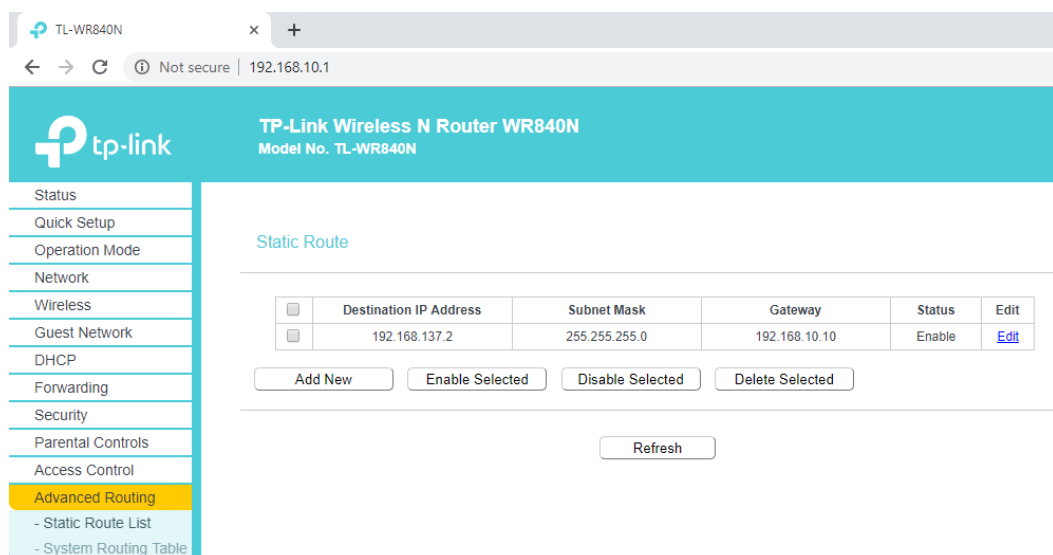


Gambar 3.34 Konfigurasi Router

Kemudian setelah IP address LAN diatur dengan IP 192.168.10.1 dan subnet mask 255.255.255.0, kemudian save.

Selanjutnya *setting ip* yang akan menghubungkan antara *router, snort* dan *server*. Untuk saling menghubungkan *ip* tersebut, kita perlu *menyetting* pada *router* dengan masuk kedalam menu *advanced routing*, dan pilih *static route list*.

1. Pilih *add new*
2. Masukkan *destination ip address 192.168.137.2*, *ip* tersebut merupakan *ip server*
3. Masukkan *subnet mask 255.255.255.0*
4. Masukkan *ip gateway 192.168.10.10*, *ip* tersebut merupakan *ip pada snort*.
5. Kemudian *save* dan *klik enable selected* untuk mengenablekan jaringan tersebut.



**Gambar 3.35** Konfigurasi Router

Setelah semua disetting maka konfigurasi *router TP-LINK WR840N* telah selesai.

### 3.8 Rincian Biaya Penelitian

Dalam sebuah penelitian ada beberapa alat yang dibutuhkan untuk menunjang penelitian ini. Seperti Laptop, Smartphone, dan Router, maka dari itu berikut rincian alat dan biaya yang dibutuhkan :

**Tabel 3.1** Rincian Harga Barang Yang Digunakan Dalam Penelitian

<b>NO</b>	<b>BAHAN</b>	<b>JUMLAH</b>	<b>HARGA (Rp)</b>
<b>1</b>	Laptop Acer aspire E 14 E5-475G-33DM	1	4.500.000
<b>2</b>	Smartphone OPPO F3	1	4.000.000
<b>3</b>	Laptop Lenovo	1	3.500.000
<b>4</b>	Router TP-LINK WR840-N	1	300.000
<b>TOTAL</b>		<b>4</b>	<b>Rp.12.300.000</b>

## BAB IV

### IMPLEMENTASI DAN HASIL

#### 4.1 Serangan *Brute Force Attack*

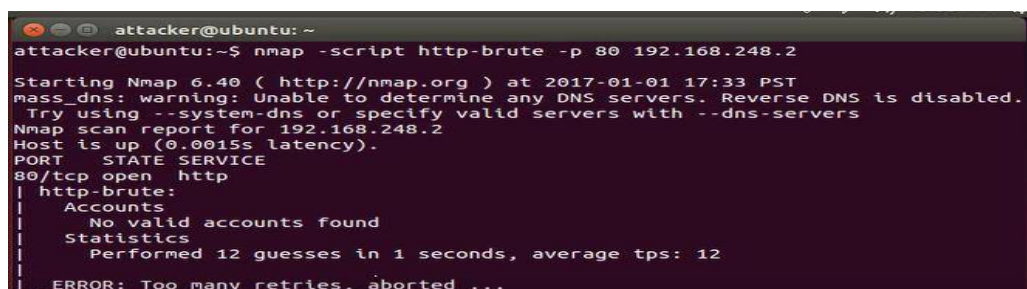
Serangan *bruteforce* bekerja dengan cara mengacak *username* dan *password server* secara sistematis, sehingga apa bila serangan ini berhasil maka *username* dan *password* akan dimiliki. Apabila *username* dan *password* telah dimiliki seorang penyerang atau *hacker* maka dia akan dengan mudah masuk kedalam *server* dan akan dengan mudah merusak dan mencuri isi dan data pribadi dalam *server* tersebut.

ini bisa dilakukan dengan beberapa cara, dan disini saya akan menggunakan *software* yang bernama *Brutus Aet2* dan *script/perintah* di terminal untuk menyerang *server* tersebut.

##### 4.1.1 *Script* untuk membuat serangan *Brute force*

Berikut adalah salah satu cara yang digunakan untuk menyerang *server* yang menggunakan jaringan *wireless* yaitu dengan membuat *script* untuk memberikan *bruteforce* ke target serangan:

```
Nmap -script http-brute -p 80 192.168.137.2
```



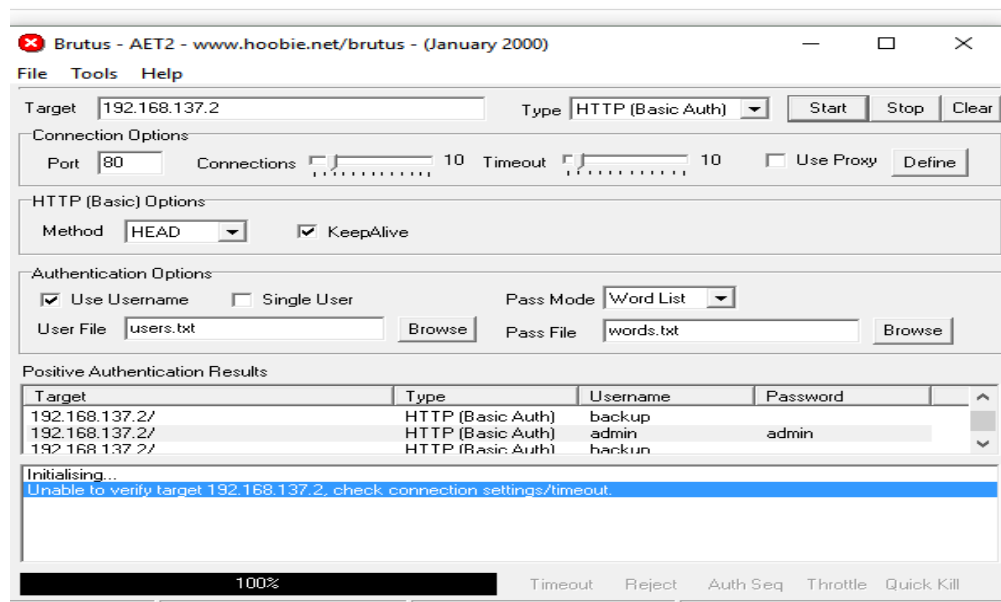
```
attacker@ubuntu: ~
attacker@ubuntu:~$ nmap -script http-brute -p 80 192.168.248.2
Starting Nmap 6.40 ( http://nmap.org ) at 2017-01-01 17:33 PST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.248.2
Host is up (0.0015s latency).
PORT      STATE SERVICE
80/tcp    open  http
| http-brute:
| Accounts
| No valid accounts found
| Statistics
| Performed 12 guesses in 1 seconds, average tps: 12
|_ ERROR: Too many retries, aborted ...
```

**Gambar 4.1** Uji Coba Serangan *Bruteforce* Menggunakan *Script*

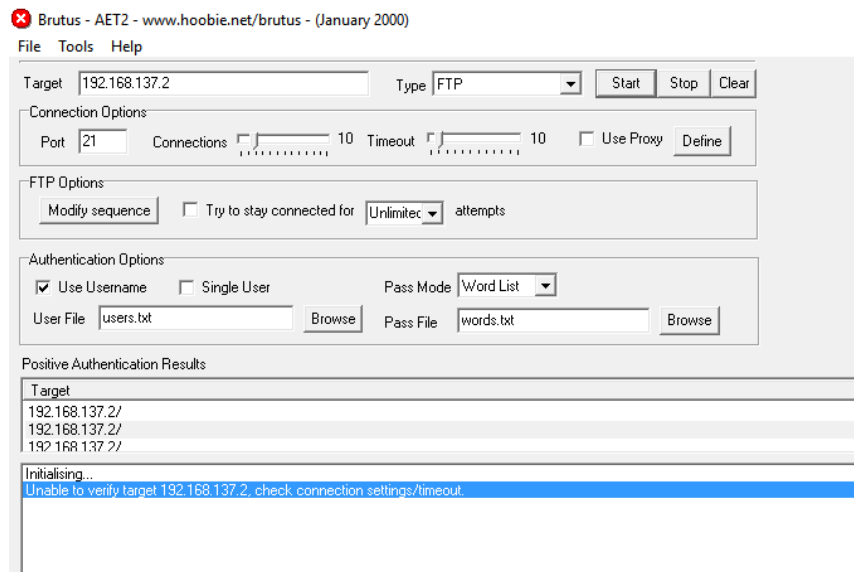
#### 4.1.2 Menyerang *Server* Dengan *Software*

Menyerang sebuah *server* dengan *bruteforce* juga bisa dilakukan menggunakan *software* yang bernama *Brutus Aet2* hanya dengan memasukkan alamat tujuan yang ingin diserang dan kemudian *software* ini akan mengidentifikasi alamat *IP address* dan bisa langsung menyerang *server* dengan mengacak dan mencari username dan password untuk melakukan *login Software* ini bukan satu-satunya alat jahat untuk menjatuhkan sebuah *server* dan merugikan pemilik jika disalah gunakan.

Tetapi sebenarnya *software* ini diperuntukkan untuk penggunaan percobaan keamanan dari serangan *bruteforce* dan *FTP*. Tapi banyak juga orang yang menyalah gunakan *software* ini untuk melakukan tindakan kejahatan. Berikut gambar serangan yang menggunakan *Brutus Aet2* :



**Gambar 4.2** Serangan *Bruteforce* Menggunakan *Brutus Aet2*.



**Gambar 4.3** Serangan *FTP* menggunakan *Brutus Aet2*.

Kesimpulan gambar :

1. Pada gambar 4.2, *brutus* menyerang menggunakan *bruteforce* dengan *ip address 192.168.137.2* dengan *port* yang dituju yaitu 80.
2. Namun bisa dilihat bahwasanya serangan langsung di blok dengan ips dengan ditandai kalimat *unable to verify target 192.168.137.2*.
3. Pada gambar 4.3, *brutus* melakukan serangan dengan menggunakan metode *FTP* dengan *ip address* sama *192.168.137.2*. dengan *port* yang berbeda yaitu *port 21*.
4. Namun bisa dilihat bahwasanya serangan juga langsung di blok dengan IPS dengan ditandai kalimat *unable to verify target 192.168.137.2*.

Jadi *Intusion Prevention System* bekerja secara langsung dengan memblok jaringan yang masuk melalui *port-port* yang sudah di konfigurasi dalam rules.

#### 4.2 Pengukuran Kinerja Keamanan

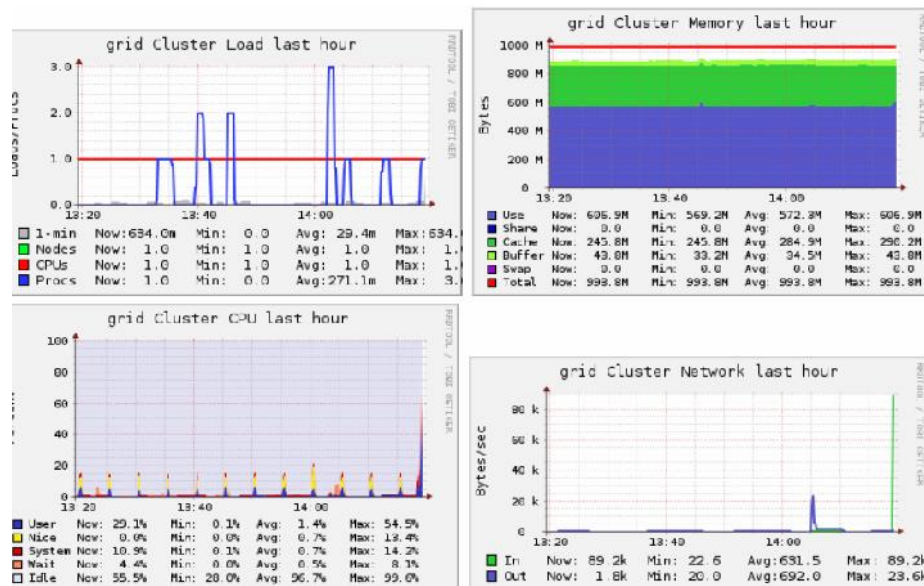
Kinerja sistem keamanan jaringan yang telah dibangun akan diuji coba dengan menyerang *server* tersebut dengan *bruteforce* yaitu jenis serangan berbahaya karena akan mengancam kinerja *server* tersebut.

Melihat kinerja sistem keamanan yang terpasang pada *server* ini bisa menggunakan *software ganglia*, yaitu *software* yang mendukung untuk memantau kinerja *server* melalui *browser* dengan *user interface GUI*, sehingga memudahkan untuk memantau kinerja dengan tampilan laporan kinerja yang lebih *friendly*.

Pengukuran dimulai dari membuat serangan terhadap *server* tersebut dan melihat bagaimana kinerja tersebut dalam bentuk grafik, dimana grafik akan menampilkan beberapa hal yang bekerja yaitu *processor*, *memory*, *network*, dan *load* yang akan ditampilkan secara bersamaan didalam satu gambar.

Kinerja *memori* dalam satu jam terakhir pada *server* tersebut terlihat datar dan tidak ada berubah karena tidak ada *client* yang masuk kedalam jaringan tersebut, dan membuat kinerja *server* ini menjadi maksimal dan bisa merespon permintaan dari *client* dengan baik. Dalam keadaan normal *memory* pada *server* terpakai sekitar 50% dari kapasitas yang diberikan.





Gambar 4.4 Keadaan normal pada server

#### 4.2.1 Kinerja Snort

Jika dijalankan dalam mode *console* maka *snort* akan menampilkan aktifitas yang masuk kedalam *host* secara *real time* dan jika ada segala aktifitas maka *snort* akan memasukkannya kedalam *log file*. Untuk mengetahui apakah *snort* tetap berjalan dengan baik ketika dilakukan penyerangan maka bisa di ping, untuk melihat apakah *server* dan *router* masih berjalan normal setelah diserang. Seperti berikut:

```

cahyo@cahyo-VirtualBox:~$ ping 192.168.137.2
PING 192.168.137.2 (192.168.137.2) 56(84) bytes of data:
64 bytes from 192.168.137.2: icmp_seq=1 ttl=64 time=0.324 ms
64 bytes from 192.168.137.2: icmp_seq=2 ttl=64 time=1.00 ms
64 bytes from 192.168.137.2: icmp_seq=3 ttl=64 time=0.477 ms
64 bytes from 192.168.137.2: icmp_seq=4 ttl=64 time=0.412 ms
64 bytes from 192.168.137.2: icmp_seq=5 ttl=64 time=0.673 ms
64 bytes from 192.168.137.2: icmp_seq=6 ttl=64 time=0.528 ms
64 bytes from 192.168.137.2: icmp_seq=7 ttl=64 time=0.391 ms
64 bytes from 192.168.137.2: icmp_seq=8 ttl=64 time=0.540 ms
^Z
[1]+  Stopped                  ping 192.168.137.2
cahyo@cahyo-VirtualBox:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data:
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=1.37 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=1.36 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=1.40 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=1.05 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=64 time=1.37 ms
64 bytes from 192.168.10.1: icmp_seq=6 ttl=64 time=1.18 ms
64 bytes from 192.168.10.1: icmp_seq=7 ttl=64 time=1.24 ms
64 bytes from 192.168.10.1: icmp_seq=8 ttl=64 time=1.33 ms
64 bytes from 192.168.10.1: icmp_seq=9 ttl=64 time=1.33 ms
64 bytes from 192.168.10.1: icmp_seq=10 ttl=64 time=1.31 ms
^Z
[2]+  Stopped                  ping 192.168.10.1
cahyo@cahyo-VirtualBox:~$
  
```

Gambar 4.5 Ping Kinerja Snort IPS

Pada gambar di atas bisa dilihat *IP 192.168.137.2* yang merupakan *IP server* tetap bisa berjalan dengan baik dan *IP 192.168.10.1* yang merupakan *IP router* juga berjalan dengan baik.

#### 4.2.2 Kinerja router

*Router* sebagai pembagi jaringan juga harus di uji coba, karena apa bila *router* gagal mendistribusikan koneksi maka program akan gagal. Sebab jaringan tidak bisa terdistribusikan. Berikut adalah gambar *ping router* yang berjalan pada sistem:

```

C:\Users\Windows 10>ping 192.168.137.2
Pinging 192.168.137.2 with 32 bytes of data:
Reply from 192.168.137.2: bytes=32 time=1ms TTL=63
Reply from 192.168.137.2: bytes=32 time=2ms TTL=63
Reply from 192.168.137.2: bytes=32 time=1ms TTL=63
Ping statistics for 192.168.137.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
C:\Users\Windows 10>ping 192.168.137.4
Pinging 192.168.137.4 with 32 bytes of data:
Reply from 192.168.137.4: bytes=32 time<1ms TTL=64
Reply from 192.168.137.4: bytes=32 time=1ms TTL=64
Reply from 192.168.137.4: bytes=32 time=1ms TTL=64
Reply from 192.168.137.4: bytes=32 time=1ms TTL=64
Ping statistics for 192.168.137.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\Users\Windows 10>ping 192.168.10.10
Pinging 192.168.10.10 with 32 bytes of data:
Reply from 192.168.10.10: bytes=32 time<1ms TTL=64
Reply from 192.168.10.10: bytes=32 time<1ms TTL=64
Reply from 192.168.10.10: bytes=32 time<1ms TTL=64
Reply from 192.168.10.10: bytes=32 time<1ms TTL=64
Ping statistics for 192.168.10.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\Windows 10>

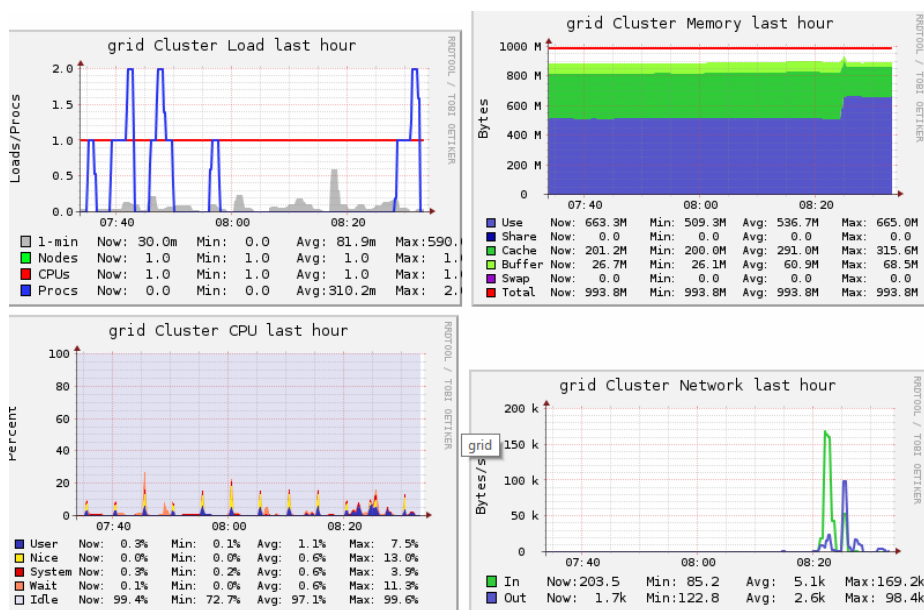
```

**Gambar 4.6** Ping Kinerja Router

*Port* tersebut adalah *port* yang sedang digunakan untuk konfigurasi *server* dan sedang terbuka sehingga dapat diakses.

### 4.3 Penyerangan dengan 1 Penyerang

Serangan *bruteforce* akan dilakukan dengan menggunakan 1 penyerang yaitu dengan menggunakan alamat IP yang menuju kedalam *server* tersebut, percobaan ini dilakukan dengan menggunakan 1 komputer dan *software Brutus Aet2* yang menggunakan jenis serangan berbeda secara bergantian dan mulai dari 1 penyerang. Berikut adalah grafik kinerja *server* dengan serangan 1 penyerang.



**Gambar 4.7** Kinerja serangan dengan 1 penyerang

Dari grafik diatas diketahui ada perubahan yang cukup membuat *server* menjadi lebih berat karena kinerja dari *server* tersebut meningkat, kinerja dari *server* ini lebih baik lagi, dan dengan membangun *IPS* bisa membuat keamanan *server* tersebut menjadi lebih aman dengan tambahan *inline mode affpacket* yang telah diaktifkan dan beberapa konfigurasi yang telah dibuat.

### 4.3.1 Hasil Pengukuran Kinerja IPS pada Server

Adapun hasil pengukuran kinerja *Intrusion Prevention System* pada server dapat dilihat pada tabel berikut:

**Tabel 4.1.** Hasil Pengukuran Kinerja IPS

No.	Jumlah <i>host</i> penyerang	Pengukuran	
		<i>Memory</i>	<i>CPU</i>
1	1 <i>host</i>	650 MB	54.5%

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Dari pembahasan di atas tentang “Implementasi Kinerja *Intrusion Prevention System (IPS)* Sebagai Sistem Keamanan Pada Jaringan *Wireless*”, penulis dapat menarik beberapa kesimpulan yang mana nantinya dapat berguna bagi para pembaca dan juga masyarakat umum lainnya. Beberapa kesimpulan dapat dilihat sebagai berikut:

1. Perancangan sistem ini dibangun dari awal dengan tujuan untuk mengamankan server yang menggunakan jaringan *nirkabel* atau *wireless*, dari berbagai macam serangan salah satunya *bruteforce*. Sehingga data yang ada tidak dapat di curi dan di hack orang yang tidak bertanggung jawab.
2. Pada perancangan IPS ini mampu mengamankan *server* dari serangan *bruteforce* dan *FTP*.
3. Implementasi IPS ini terdapat 4 pilihan proses pengamanan, namun disini saya menggunakan *inline mode afpacket*, yaitu proses pengamanan yang menggunakan 2 *interfaces* sebagai penghubung antara *snort* dan *server*.
4. IPS ini berjalan pada *software* yang bernama *snort*, dan *snort* itu sendiri berada pada sistem operasi *linux Ubuntu 16.04*.
5. Sistem keamanan ini bersifat gratis sehingga mudah untuk mengimplementasikannya di salam sistem keamanan sederhana.

## 5.2 Saran

Dari hasil perancangan sistem keamanan yang berbasis *Intrusion Prevention System* ini bahwa terdapat saran yang ditujukan pada para pengguna untuk pengembangan selanjutnya, sebagai berikut:

1. Perancangan sistem keamanan ini dari segi tampilan maupun yang lainnya masih terdapat kekurangan, sehingga para pengguna dapat mengembangkan tampilan tersebut maupun yang lainnya menjadi lebih bagus.
2. Dalam sistem keamanan ini masih terbatas pada penanganan beberapa jenis serangan saja, sehingga dibutuhkan pengembangan selajutnya agar sistem keamanan ini dapat mengamankan dari serangan yang lainnya dan dapat mencakup lingkungan yang lebih luas.

## DAFTAR PUSTAKA

- Andrian, Yudhi, and Purwa Hasan Putra. "Analisis Penambahan Momentum Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network." Seminar Nasional Informatika (SNIf). Vol. 1. No. 1. 2017.
- Arief, M.R. (2007). Teknologi Jaringan Tanpa Kabel (*Wireless*). *Seminar nasional teknologi 2007*, ISSN : 1978 – 9777.
- Aryza, S., Irwanto, M., Lubis, Z., Siahaan, A. P. U., Rahim, R., & Furqan, M. (2018). A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012067). IOP Publishing.
- Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." *IT Journal Research and Development* 2.1 (2017): 1-11.
- Darmawan, D, Marlinda, L (2015). Impelementasi Jaringan Wireless Outdoor Menggunakan NaniBridge. *Jurnal teknik informatika*, Vol.1 No.12. ISSN : 2442-2436
- Fachri, B. (2018, September). Aplikasi Perbaikan Citra Efek Noise Salt & Papper Menggunakan Metode Contraharmonic Mean Filter. In Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 87-92).
- Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. *Int. J. Recent Trends Eng. Res*, 3(8), 58-64.
- Hadiyanti, R. (2013). Implementasi Peraturan Pemerintah Nomor 8 Tahun 2003 Tentang Pedoman Organisasi Perangkat Daerah Pemerintah Kota Samarinda. *E-journal pemerintahan*, 1 (3), 985 – 997, Diakses dari [ejournal.ip.fisip.unmul.ac.id](http://ejournal.ip.fisip.unmul.ac.id)
- Hafni, Layla, and Rismawati Rismawati. "Analisis Faktor-Faktor Internal Yang Mempengaruhi Nilai Perusahaan Pada Perusahaan Manufaktur Yang Terdaftar Di BEI 2011-2015." *Bilancia: Jurnal Ilmiah Akuntansi* 1.3 (2017): 371-382.

- Halawa, S. (2016). Perancangan Aplikasi Pembelajaran Topologi Jaringan Komputer Untuk Sekolah Menengah Kejuruan (Smk) Teknik Komputer Dan Jaringan (Tkj) Dengan Metode Computer Based Instruction. *Jurnal Riset Komputer (JURIKOM)*, Volume : 3, Nomor: 1. ISSN : 2407-389X.
- Harjono, E.B. (2016). Analisa Dan Implentasi Dalam Membangun Sistem Operasi *Linux* Menggunakan Metode LSF Dan REMASTER. *Jurnal teknik informatika*, Vol.1 No.1. ISSN : 2541-2019
- Ikhwan, S., Elfitri, I. (2014). Analisa Delay Yang Terjadi Pada Penerapan Demilitarized Zone (Dmz) Terhadap Server Universitas Andalas. *Jurnal Nasional Teknik Elektro*, Vol: 3 No. 2. ISSN: 2302 – 2949.
- INDRA PERMANA, A. M. I. N. U. D. D. I. N. "Sistem Pakar Mendeteksi Hama Dan Penyakit Tanaman Kelapa Sawit Pada Pt. Moeis Kebun Sipare-Pare Kabupaten Batubara." (2013).  
*Jurnal Sistem Informasi*, 5 (2), 1-17, ISSN: 1979-0767.
- Khadafi, S. (2017). Sistem Keamanan Open Cloud Computing Menggunakan IDS (*Intrusion Detection System*) Dan IPS (*Intrusion Prevention System*). *Jurnal Iptek*, Vol.21 No.2. ISSN : 1411-7010
- Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." *Int. J. Recent Trends Eng. Res* 2.12 (2016): 140-151.
- Novianta, M.A., Setyaningsih, E. (2015). Sistem Informasi Monitoring Kereta Api Berbasis Web Server Menggunakan Layanan GPRS. *Jurnal Momentum*, Vol.17 No.2. ISSN : 1693-752X
- Nurmiati, E. (2012). Analisis Dan Perancangan Web Server Pada Handphone.
- Permana, A. I., and Z. Tulus. "Combination of One Time Pad Cryptography Algorithm with Generate Random Keys and Vigenere Cipher with EM2B KEY." (2020).
- Permana, Aminuddin Indra. "Kombinasi Algoritma Kriptografi One Time Pad dengan Generate Random Keys dan Vigenere Cipher dengan Kunci EM2B." (2019).
- Preayogo, F.A., (2017). Perancangan Sistem Pencegahan Serangan *Bruteforce* pada Jaringan *Wireless*. *Artikel ilmiah*
- Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di



- Sumatera Utara." Seminar Nasional Teknologi Informasi Dan Multimedia, ISSN. 2015.
- Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. *Int. J. Secur. Its Appl*, 10(8), 173-180.
- Putra, Randi Rian, and Cendra Wadisman. "Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K Means." *INTECOMS: Journal of Information Technology and Computer Science* 1.1 (2018): 72-77.
- Sari, H.L., Sudarsono, A., Hayadi, B.H. (2013). Pengembangan Jaringan Local Area Network Menggunakan Sistem Operasi Linux Redhat 9. *Jurnal Media Infotama*, Vol.9, No.1. ISSN : 1858 – 2680.
- Suhartono, D., Riyanto, A.D., Astomo, Y.W. (2015). Intrusion Detection Prevention System (Idps) pada Local Area Network (Lan). *Jurnal Telematika*, Vol 8 No. 1. ISSN : 1979 – 925X e-ISSN : 2442 – 4528.
- Syahputra, Rizki, and Hafni Hafni. "Analisis Kinerja Jaringan Switching Clos Tanpa Buffer." *Journal Of Science And Social Research* 1.2 (2018): 109-115.
- Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." *Jurnal Abdi Ilmu* 10.2 (2018): 1899-1902.