MENGIMPLEMENTASIKAN SISTEM KEMANAN JARINGAN *INTRUSION PREVENTION SYSTEM* BERBASIS *SNORT* PADA ARSITEKTUR *SOFTWARE DEFINED NETWORK*

IMPLEMENTING SNORT BASED INTRUSION PREVENTION SYSTEM AS NETWORK SECURITY IN SOFTWARE DEFINED NETWORK

Mochamad Bagus Firmansyah[1], Ridha Muldina Negara[2], Danu Dwi Sanjoyo [3]

Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

[1]mbfirmansyah4896@gmail.com [2]ridhanegara@telkomuniversity.ac.id

[3]danudwj@telkomuniversity.ac.id

Abstrak

***Software Defined Network* (SDN) adalah sebuah arsitektur jaringan yang memisahkan antara *control plane* dan *data plane*. SDN bisa juga disebut sebagai sentralisasi jaringan karena keseluruhan jaringan dapat diatur hanya dari satu titik yang disebut *controller* menjadikan SDN memiliki fleksibilitas yang lebih tinggi dibandingkan jaringan konvensional. Dibalik kelebihannya, keamanan masih menjadi perhatian utama di dalam arsitektur ini. Maka dari itu pada Tugas Akhir ini akan dibuat sistem kemanan jaringan *Intrusion Prevention System* (IPS) untuk melindungi jaringan dari serangan. IPS dilakukan dengan cara melakukan integrasi fungsi *Intrusion Detection System* (IDS) pada *Snort*. IDS akan mendeteksi paket yang dianggap serangan sesuai rules yang sudah ditentukan. Kemudian dari rules tersebut akan diambil data seperti *packet source, packet destination* dan protokol paket untuk bisa melakukan blokir terhadap serangan. Serangan yang akan disimulasikan bervariasi mulai dari menggunakan banyaknya paket sampai ukuran paket yang dikirim sebagai parameter.**

Kata Kunci: SDN, IDS, IPS, snort.

Abstract

***Software Defined Network* (SDN) is a network architecture which make control plane and data plane being separated. SDN also known as centralized network because a network administrator can handle a whole network by a thing called controller. This makes SDN has more flexibility than conventional network. At the same time, security invulnerabilities of this technology are still the biggest concern of researches for adapting this technology. This Final Project will make an IPS security system. IPS will made by integrating the Snort IDS function with Ryu's rest_firewall module. IDS will capture packet and detect if it is an attack or not by the rules created before. If it is an attack, system will take packet source, paket destination and packet protocol data for triggering rest_firewall module to block the packet. In this Final Project will use packet interval and packet size as parameter for blocking.**

Key word: SDN, IDS, IPS, snort.

1. Pendahuluan

Pertumbuhan jaringan komputer bergerak dengan sangat cepat seiring dengan pertumbuhan penggunanya. Oleh karena itu dibutuhkan sebuah inovasi yang memungkinkan untuk meningkatkan efisiensi dalam mendisain, mengelola serta menerapkan sebuah jaringan komputer. Adapun SDN dapat dipilih sebagai solusi untuk hal tersebut.

SDN adalah sebuah arsitektur jaringan yang memisahkan antara *control plane* dan *data plane*. *Control plane* berfungsi untuk mengatur jaringan secara keseluruhan. Sedangkan *data plane* hanya memiliki fungsi *forwarding*. *Control plane* terdapat pada titik yang biasa disebut *controller* sedangkan *data plane terdapat* pada masing-masing perangkat jaringan seperti *switch*, *router*, dsb.

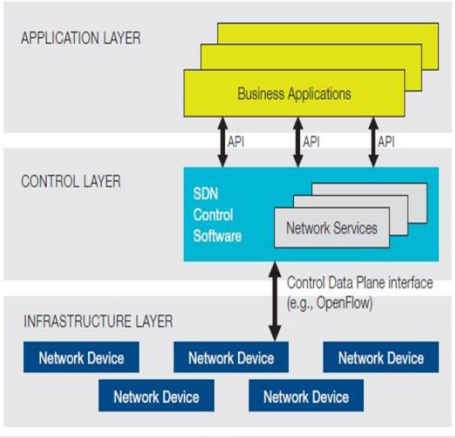
Pada Tugas Akhir ini akan dilakukan penerapan sistem kemanan jaringan IPS dimana *Snort* akan diintegrasikan dengan modul *rest_firewall* pada *Ryu Controller*. Setelah dilakukan penerapan sistem kemanan jaringan akan dilakukan simulasi terhadap sistem kemanan yang telah dibuat dengan paket serangan ICMP *Flood* dan *Ping of Death*.

Tujuan dari Tugas Akhir ini adalah untuk menciptakan sistem kemanan jaringan yang tidak hanya terfokus kepada salah satu parameter seperti ukuran paket dan interval pengiriman paket.

2. Dasar Teori

2.1 *Software Defined Network* (SDN)

SDN adalah inovasi dalam arsitektur jaringan. Pada arsitektur ini *control plane* dan *data plane* sudah tidak terletak di dalam satu elemen jaringan. SDN bisa disebut sebagai jaringan tersentralisasi. Perangkat *control plane* berfungsi untuk mengatur jaringan sedangkan *data plane* hanya berfungsi untuk *forwarding*.



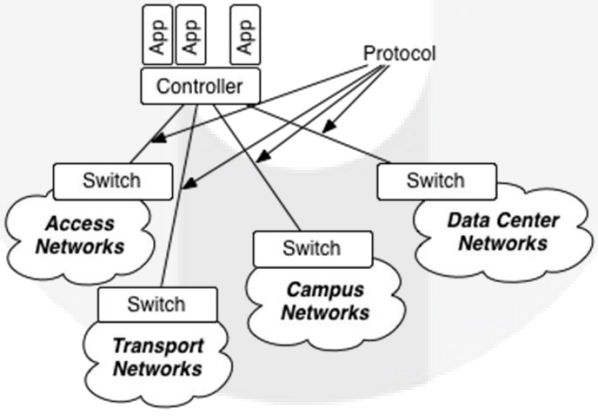
Gambar 1 Arsitektur SDN

Pada **Gambar 1** dapat dilihat bahwa SDN terdiri dari 3 layer yang memiliki fungsi antara lain:

1. *Application Layer*
   Pada *layer* ini sebuah jaringan dapat dikembangkan melalui modul-modul aplikasi baik yang sudah tersedia ataupun yang dikembangkan sendiri.
2. *Control Layer*
   Layer ini bertugas untuk memberi regulasi serta mengelola perangkat *forwarding*.
3. *Infrastructure Layer*
   Layer yang berfungsi untuk *forwarding* melalui perangkat seperti *swtich* ataupun *router*.

2.2 *OpenFlow*

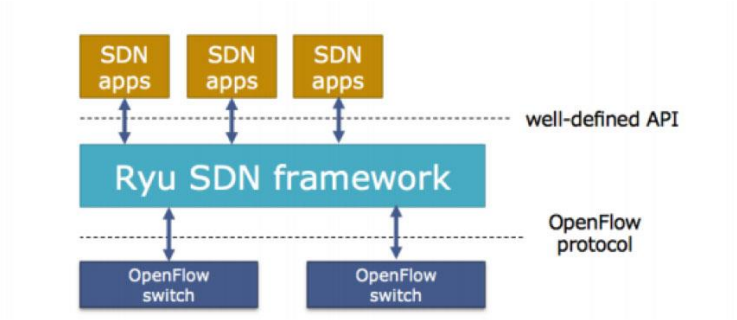
OpenFlow adalah salah satu protokol yang berada di antara *control layer* dan *infrastrucutre layer*. *OpenFlow* memungkinkan *control plane* untuk melakukan kontrol langusng terhadap perangkat *forwarding* untuk meregulasi bagaimana pergerakan paket atau *flow* di dalam sebuah jaringan. *OpenFlow* juga memungkinkan sebuah *controller* untuk berinteraksi dengan *switch* yang memiliki tipe yang berbeda-beda.



Gambar 2 Protokol *OpenFlow*

2.4 *Ryu Controller*

Ryu Controller adalah salah satu contoh *controller* yang dapat digunakan dalam menerapkan jaringan dengan arsitektur SDN. *Ryu* menyediakan berbagai macam *Application Interface* (API) [6]. Salah satunya adalah *rest_firewall*. *Ryu Controller* dapat dikembangkan dengan menggunakan Bahasa Pemrograman Python.



Gambar 3 Arsitektur SDN dengan Ryu sebagai *controller* [7]

2.5 *Snort*

Snort adalah sebuah perangkat lunak *open source* untuk aplikasi IDS [8]. *Snort* memiliki kemampuan untuk melakukan *real-time traffic analysis* dan *packet logging*. Untuk Tugas Akhir ini *Snort* akan dikonfigurasi sebagai *Network Intrusion Detection* dimana *Snort* akan melakukan pemindaian trafik di dalam sebuah jaringan, menganalisa paket tersebut dan dicocokkan dengan *rules* yang sudah ditentukan yang dapat dikeluarkan sebagai *alert* ataupun dalam bentuk *log*.

2.6 *Intrusion Detection System* (IDS) & *Intrusion Prevention System* (IPS)

IDS dan IPS adalah sebuah sistem yang dirancang untuk meningkatkan kemanan sebuah jaringan [21]. IDS dan IPS sama-sama merupakan bagian dari infrastruktur sebuah jaringan, lebih tepatnya dibagian kemanan. Perbedaan utama keduanya adalah, IDS merupakan sebuah *monitoring system* sedangkan IPS adalah *control system* [22].

Gambar 4 Perbedaan IDS dan IPS

2.7 Denial of Service (DoS)

Ada tiga pilar utama dalam kemanan jaringan, salah satunya adalah *availability*. *Availability* memungkinkan sesama *client* atau *server-client* untuk terhubung dan melakukan pertukaran data. Untuk Tugas Akhir ini akan dilakukan simulasi DoS berupa:

2.7.1 ICMP *Flood*

ICMP *Flood* adalah sebuah serangan DoS dimana penyerang membanjiri korban dengan ICMP *echo-request* yang memakan bandwidth korban dan membuat korban tidak dapat terhubung ke jaringan secara normal [11].

2.7.2 *Ping of Death*

Ping of Death adalah sebuah serangan DoS dimana penyerang mengirimkan ICMP *echo-request* dengan ukuran yang tidak wajar sehingga membuat sistem korban menjadi *freeze* atau *crash* [12].

2.8 *Bandwidth*

Bandwidth atau lebar pita adalah transfer data rate, yaitu jumlah data yang dapat dibawa dalam satu kali pengiriman dari suatu titik ke titik lain dalam satuan waktu tertentu, bandwidth atau transfer data rate biasanya dihitung dalam satuan detik seperti bit per second (bps), kilobit per second (kbps), mengabit per second (mbps) dan seterusnya.

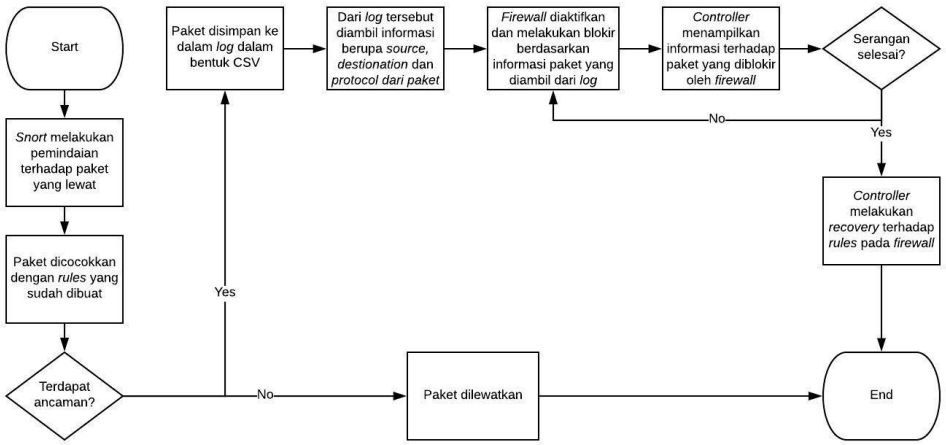
3 Perancangan Sistem

3.1 Desain Sistem

Sistem yang dibuat adalah sebuah jaringan menggunakan arsitektur SDN dengan IPS dengan cara mengintegrasikan modul firewall dari Ryu Controller dengan IDS dari Snort. Lalu sistem tersebut akan diuji menggunakan serangan, ketika paket serangan masuk ke dalam jaringan, Snort akan mendeteksi serangan tersebut, jika sesuai rules yang telah ditentukan maka paket tersebut akan disimpan ke dalam log, selanjutnya modul firewall akan otomatis memblokir paket serangan berdasarkan data pengirim, penerima dan protokol paket.

3.1.1 Alur Kerja Sistem

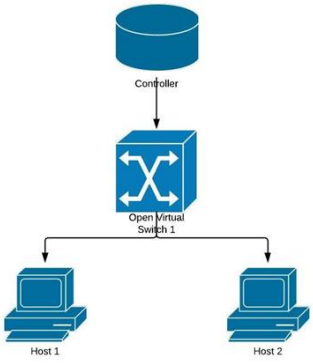
Pertama-tama akan dilakukan montoring terhadap paket yang lewat menggunakan fungsi IDS dari *Snort* setelah itu jika terdapat paket yang sesuai dengan *rules* maka paket tersebut akan dimasukkan ke dalam *log*, setelah itu dari *log* akan dibaca data berupa *packet source*, *packet destination* dan *packet protocol*. Dari data-data yang diambil dari *log* akan dilakukan pemblokiran melalui *rules* dari modul *rest_firewall* yang ada pada *Ryu Controller*.



Gambar 5 Diagram alur kerja sistem

3.1.2 Rancangan Topologi Sistem

Topologi yang akan dilakukan untuk simulasi adalah sebagai berikut:



Gambar 6 Topologi jaringan

4. Pengujian dan Analisis

4.1 Skenario Uji Sistem

Akan dilakukan uji sistem dengan cara melakukan serangan kombinasi antara ICMP *F*lood dan *Ping of Death*. Dari sistem yang diserang oleh serangan tersebut secara bersamaan dapat dilihat bahwa ada bandwidth yang terbuang sia-sia akibat serangan seperti gambar di bawah ini:

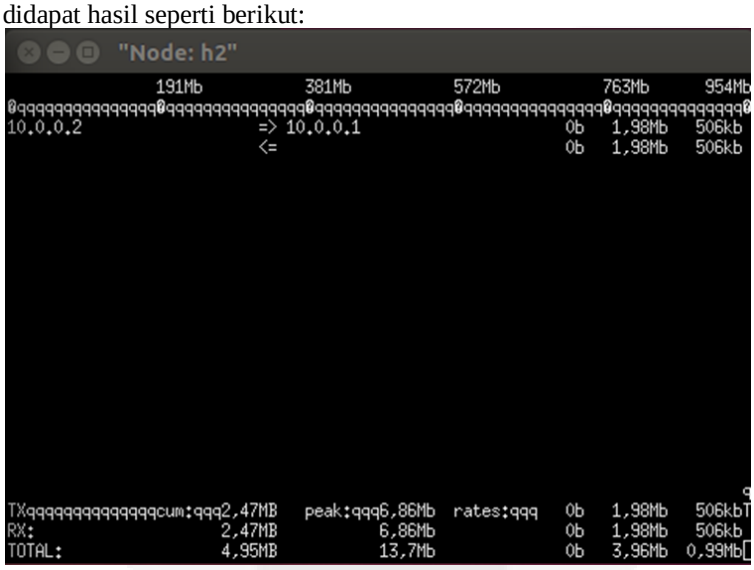
Gambar 8 *Bandwidth* terbuang akibat serangan

Untuk menanggulangi hal tersebut maka dilakukan integrasi *Snort* IDS dengan modul *rest_firewall* dari *Ryu Controller*. Dari serangan tersebut maka didapatkan *alert* sebagai berikut:

```
01/03-02:39:02.708282 ,"PING OF DEATH DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708286 ,"ICMP FLOOD DETECTED",10.0.0.2,10.0.0.1,ICMP
01/03-02:39:02.708291 ,"ICMP FLOOD DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708291 ,"PING OF DEATH DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708295 ,"ICMP FLOOD DETECTED",10.0.0.2,10.0.0.1,ICMP
01/03-02:39:02.708299 ,"ICMP FLOOD DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708299 ,"PING OF DEATH DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708303 ,"ICMP FLOOD DETECTED",10.0.0.2,10.0.0.1,ICMP
01/03-02:39:02.708307 ,"ICMP FLOOD DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708307 ,"PING OF DEATH DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708311 ,"ICMP FLOOD DETECTED",10.0.0.2,10.0.0.1,ICMP
01/03-02:39:02.708316 ,"ICMP FLOOD DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708316 ,"PING OF DEATH DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708320 ,"ICMP FLOOD DETECTED",10.0.0.2,10.0.0.1,ICMP
01/03-02:39:02.708324 ,"ICMP FLOOD DETECTED",10.0.0.1,10.0.0.2,ICMP
01/03-02:39:02.708324 ,"PING OF DEATH DETECTED",10.0.0.1,10.0.0.2,ICMP
```

Gambar 9 *Alert* akibat serangan

Untuk memastikan bahwa paket benar-benar terblock, dapat dilakukan dengan membuka *bandwidth monitoring tool*, dan didapat hasil seperti berikut:



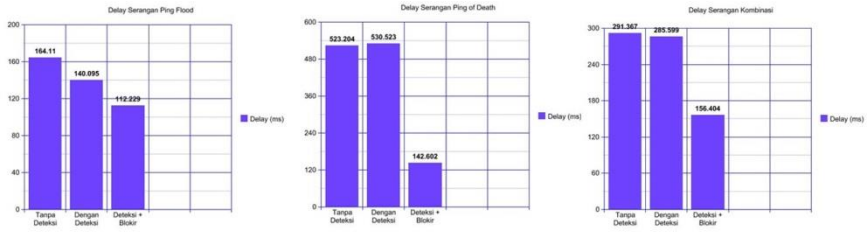
Gambar 10 *Bandwidth* setelah dilakukan *blocking*

Dari **Gambar 9** dan **Gambar 10** dapat diambil kesimpulan bahwa *Snort* sebagai IDS dapat mendeteksi serangan walaupun hasil kombinasi antara paket dengan ukuran yang besar dan interval kedatangan yang cepat. Lalu *script blocking* juga berhasil mengambil data yang dibutuhkan untuk kemudian menjalankan mekanika pemblokiran.

4.2 Uji Performansi

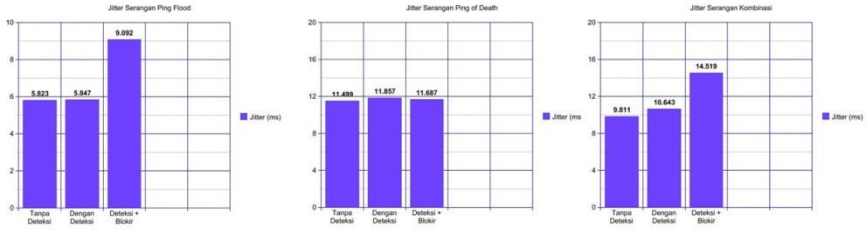
Uji performansi dilakukan untuk mengetahui performa jaringan sebelum dilakukan deteksi, saat dilakukan deteksi dan dilakukan pemblokiran dengan mengaktifkan modul *rest_firewall* dari *Ryu Controller*. Uji performansi akan dilakukan saat jaringan tidak melakukan deteksi, saat dilakukan deteksi dan dilakukan pemblokiran terhadap serangan, adapun parameter uji performansi yang digunakan antara lain:

4.2.1 *Delay*



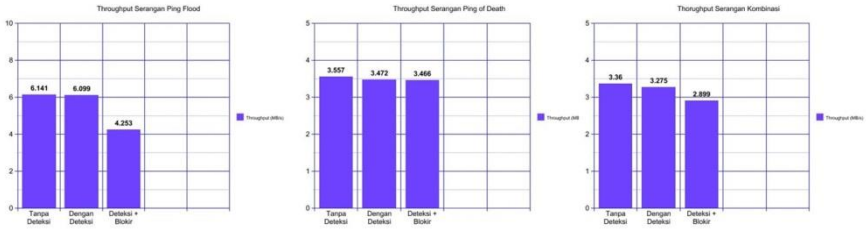
Gambar 11 Uji performansi *delay*

4.2.2 *Jitter*



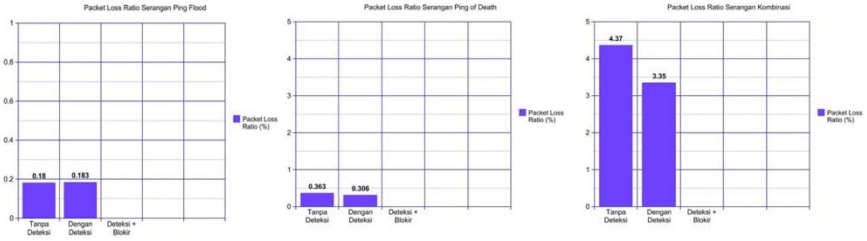
Gambar 12 Uji performansi *jitter*

4.2.3 *Throughput*



Gambar 13 Uji performansi *throughput*

4.2.4 *Packet Loss Ratio*



Gambar 14 Uji performansi *packet loss ratio*

5 Kesimpulan

Berdasarkan analisis hasil dari pengujian pada sistem yang telah dibangun dapat diambil kesimpulan sebagai berikut:

1. Arsitektur SDN dengan *Ryu* sebagai *controller* dapat diintegrasikan dengan *Snort* untuk menerapkan sistem kemanan jaringan.
2. *Snort* dapat mendeteksi serangan dengan parameter serangan ukuran paket yang dikirim dan interval kedatangan paket yang cepat.
3. Sistem blokir dapat dilakukan dengan cara mengambil data *packet source*, *packet destination* dan *packet protocol* dari *log* file *alert* yang sudah dimodifikasi ke dalam bentuk csv.

Daftar Pustaka:

[1] N., & Sood, M. 2014. *Software Defined Network – Architectures.* Parallel, Distributed and Grid Computing (PDGC).

[2] J, Dixon. 2016. *Software Defined Networking: What it is and what you shoud know!*. Infsecwriters.

[3] Abdelkarim, Amhad Abdallah & Nasereddin, Hebah H.O. 2011. *Intrusion Prevention System.* International Journal of Academic Research: Vol. 3 No.1 January, 2011, Part II.

[4] SDXCentral. 2014. *What is Software Defined Networking (SDN)? Definition*. https://www.sdxcentral.com/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/. Diakses pada tanggal 16 November 2017.
[5] Flowgrammable. 2014. *OpenFlow*. http://flowgrammable.org/sdn/openflow/. Diakses pada tanggal 16 November 2017.
[6] Ryu Development Team. 2018. *Ryu Documentation*.
[7] SDxCentral. 2014. *What is Ryu Controller?*. https://www.sdxcentral.com/sdn/definitions/sdn-controllers/open-source-sdn-controllers/what-is-ryu-controller/. Diakses pada tanggal 2 Januari 2019.
[8] Carr, Jefrey. 2007. *Snort: Open Source Network Intrusion Prevention*. eSecurity Planet. https://www.esecurityplanet.com/network-security/Snort-Open-Source-Network-Intrusion-Prevention-3681296.html. Diakses pada tanggal 18 November 2017.
[9] S. Karen & M. Peter. 2007. *Guide to Intrusion Detection and Prevention System (IDPS)*. NIST: SP-800-94
[10] Juniper. 2016. *What is IDS/IPS?*. https://www.juniper.net/us/en/products-services/what-is/ids-ips/. Diakses pada tanggal 18 November 2017.
[11] Cloudflare. 2017. *What is a Ping (ICMP) flood attack?*. https://www.cloudflare.com/learning/ddos/ping-icmp-flood-ddos-attack/. Diakses pada tanggal 2 Januari 2019.
[12] Cloudflare. 2017. *What is a Ping of Death attack?*. https://www.cloudflare.com/learning/ddos/ping-of-death-ddos-attack/. Diakses pada tanggal 2 Januari 2019.
[13] Trabelsi, Zouheir & Alketbi, Latifa. 2013. *Using network packet generators and snort rule for teaching denial of service attacks*. Conference Paper DOI: 10.1145/2462476.2465580.
[14] Dietrich, Noah. 2017. *Snort 2.9.9.x on Ubuntu 14 and 16*.
[15] Fauzi, Taufik Nur. 2016. *Integrasi Intrusion Detection System pada Software Defined Network*. Skripsi. Bandung: Universitas Telkom.
[16] Pratama, Rifqi Fauzan. 2017. *Perancangan dan Implementasi Adaptive Intrusion Prevention System (IPS) untuk Pencegahan Penyerangan pada Arsitektur Software-Defined Network (SDN)*. Skripsi. Bandung: Universitas Telkom.
[17] Farhan, Aulia. 2017. *Implementasi Intrusion Prevention System (IPS) pada Arsitektur Jaringan Software-Defined Network (SDN)*. Skripsi. Bandung: Universitas Telkom.
[18] Saputra, Pande Putu Kika Adi. 2018. *Integrasi Intrusion Prevention System dan Analisa Performansi P ada Software Defined Network*. Skripsi. Bandung: Universitas Telkom.
[19] IP WITH EASE. *Difference between IPS and IDS in Network Security*. https://ipwithease.com/difference-between-ips-and-ids-in-network-security/. Diakses pada tanggal 4 Januari 2019.
[20] Mininet. *Mininet Overview*. http://mininet.org/overview/. Diakses pada tanggal 4 Januari 2019.
[21] Petters, Jeff. 2018. *IDS vs IPS: What is the Difference?*. https://www.varonis.com/blog/ids-vs-ips/. Diakses pada tanggal 13 Januari 2019.
[22] Stump, Patrick. 2017. *IDS vs IPS : What's the difference?*. https://www.rokacom.com/ids-vs-ips/. Diakses pada tanggal 13 Januari 2019.