PERANCANGAN DAN IMPLEMENTASI *ADAPTIVE INTRUSION PREVENTION SYSTEM* (IPS) SNORT MENGGUNAKAN LOGIKA FUZZY UNTUK MENCEGAH SERANGAN PADA ARSITEKTUR *SOFTWARE-DEFINED NETWORK* (SDN)

DESIGN AND IMPLEMENTATION ADAPTIVE INTRUSION PREVENTION SYSTEM (IPS) SNORT USING FUZZY LOGIC TO PREVENT ATTACK ON SOFTWARE-DEFINED NETWORK (SDN)

Dicky Rahadian[1], Sofia Naning Hertiana[2], Fardan[3]
[1,2,3] Universitas Telkom, Bandung
dickyrahadian@student.telkomuniversity.ac.id[1], sofiananing@telkomuniversity.ac.id[2], fardanfnn@telkomuniversity.ac.id[3]

Abstrak

Intrusion Prevention System (IPS) merupakan sebuah *tools* yang digunakan untuk mendeteksi dan mencegah masuknya paket berbahaya ke dalam. Adapun IDS yang dapat digunakan pada arsitektur SDN ini salah satunya adalah Snort. Namun, IDS Snort ini sendiri memiliki sebuah kelemahan, yang dimana durasi blokir yang ditentukan akan selalu sama. Pada penelitian ini, penulis merancang suatu sistem yang dapat merubah IDS Snort menjadi *adaptive* IPS. Adapun logika yang digunakan untuk membangun sistem ini adalah logika fuzzy. Sistem keanggotaan ini yang akan menentukan, apakah frekuensi paket yang datang tersebut dikategorikan sebagai serangan yang sering atau tidak, yang kemudian hasil penentuan tersebut yang akan menentukan apakah waktu blokir untuk *host* sumber akan terus bertambah atau tidak, apabila dalam masa blokir ternyata *host* sumber masih melakukan serangan, maka waktu blokir untuk *host* terebut akan terus bertambah. Sistem fuzzy yang dibuat diuji dengan serangan *Host Discovery* dan serangan *Distributed Denial of Service*(DDoS).Hasil yang diperoleh dari penelitian Tugas Akhir ini adalah, sistem fuzzy yang telah dirancang dapat menjadikan IDS Snort menjadi *adaptive* IPS, yang dimana IPS ini dapat beradaptasi dengan frekuensi serangan yang datang. Selain itu juga, serangan *Host Discovery* dan serangan DDoS yang digunakan untuk pengujian ini dapat diblokir oleh sistem yang telah dirancang.

Kata kunci : IPS, IDS, logika fuzzy, SDN, *Host Discovery*, DDoS.

Abstract

Intrusion Prevention System (IPS) is a tools which uses to detect and prevent a dangerous packet from entering the system which is sent by a host. One of the IDS that can be used in SDN architecture is Snort. But, this IDS Snort alone has some weakness, where, the blocking duration which already decided, will always the same. Therefore, a system that can support the IDS Snort to become adaptive is needed. In this research, a system will be designed to trasform the IDS Snort into adaptive IPS. As for the algorithm to be used to support this design is, fuzzy logic. If during blocking time, the attacker is or are still attacking, then the blocking duration for the attacker will be increased based on how frequent the attack is.The Fuzzy system which is made is tested with some cyber attack method, which are, Host Discovery and Distributed Denial of Service attack.The result obtained from this research is, the Fuzzy system which was made can transform the IDS Snort into adaptive IPS. The attack method that are used to test the system, which are Host Discovery and DDoS attack, are able to be blocked as well

Keywords: IPS, IDS, Fuzzy logic, SDN, Host Discovery, DDoS.

1. Pendahuluan

Dengan semakin banyaknya teknologi yang terhubung kedalam jaringan internet, kompleksitas jaringan menjadi meningkat. Peningkatan kompleksitas ini dapat menyebabkan kesulitan operator untuk melakukan pemeliharaan jaringan. Masalah yang ada ini disolusikan oleh Open Networking Foundation dengan menciptakan sebuah konsep struktur jaringan baru yaitu *Software Defined Network* (SDN). Perbedaan struktur jaringan SDN dengan yang tradisional adalah, apabila pada

struktur jaringan tradisional *control plane* dan *data plane* disatukan pada perangkat jaringan (*e.g., router* dan *switch)*, sedangkan pada struktur jaringan SDN, *control plane* dan *data plane* dipisah, dengan menjadikan *control plane* sebagai pusat dari seluruh struktur jaringan tersebut. Hal ini memberikan kemudahan dalam pemeliharaan dan pengelolaan jaringan secara keseluruhan[1].

Akan tetapi, dengan meningkatnya kompleksitas lalu lintas, muncul pula isu keamanan jaringan yang tidak bisa diabaikan. Semakin tinggi lalu lintas yang ada pada suatu jaringan, maka semakin tinggi tingkat ancaman keamanan pada jaringan tersebut. Salah satu serangan yang umum dilakukan adalah serangan *Denial of Service* (DoS) atau serangan *Distributed Denial of Service* (DDoS). tipe serangan DoS atau DDoS yang dilakukan juga bervariasi, seperti *ICMP Flood*, *SYN Flood*, *PING Flood*, *UDP Flood*, dan masih banyak lagi. Seperti yang dibahas di penilitian pada jurnal [1], yang dimana pada jurnal tersebut mengatakan bahwa pada aristekur SDN, serangan-serangan tersebut dapat terjadi di setiap bagian arsitektur SDN. *S*alah satu cara untuk mencegah terjadinya serangan-serangan tersebut adalah dengan menggunakan *Intrusion Prevention System* (IPS), yang dimana IPS ini dapat menghentikan paket data yang dapat membahayakan dan memblokir akses *host* sumber serangan tersebut. Perangkat lunak yang dapat digunakan pada jaringan SDN ini adalah Snort[2].

Akan tetapi, Snort memiliki suatu kendala, yaitu Snort tidak bisa beradaptasi dengan frekuensi serangan yang datang[3]. Sehingga durasi waktu pemblokiran selalu statis dengan waktu yang telah ditentukan.

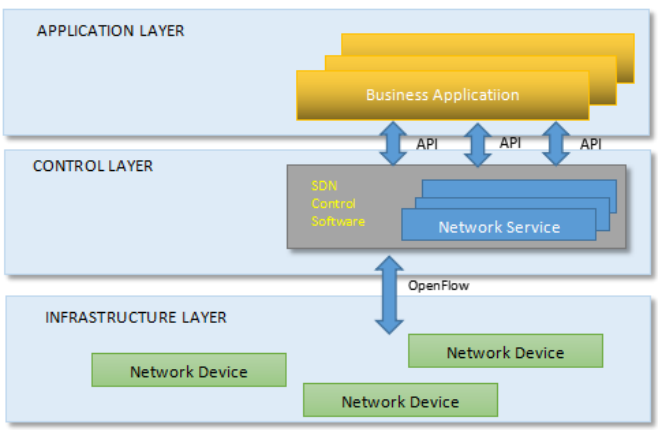
Solusi yang sesuai dengan permasalahan tersebut adalah dengan menciptakan *Adaptive Intrusion Prevention System* (*Adaptive* IPS) yang tidak hanya dapat mencegah serangan di arsitektur SDN, akan tetapi juga bisa beradaptasi dalam durasi waktu pemblokiran *host* sumber serangan berdasarkan frekuensi serangan yang datang. Logika yang bisa dimanfaatkan untuk menciptakan *Adaptive* IPS ini adalah, dengan menggunakan *fuzzy logic*[4]. Dimana data yang didapat akan dicatat ke dalam *log* Snort, lalu data *log* tersebut akan diproses oleh sistem *fuzzy* yang telah dirancang.

Pada penelitian sebelumnya yang telah dilakukan oleh peneliti sebelumnya [5],sistem *fuzzy* yang dibuat dapat beradaptasi terhadap frekuensi serangan DoS yang datang, akan tetapi belum diuji apakah dapat beradaptasi dengan frekuensi serangan DDoS yang memiliki jumlah *host attacker* lebih dari satu. Oleh karena itu, pada tugas akhir ini, akan diuji dengan serangan DDoS dan akan dikembangkan apabila sistem sebelumnya tidak dapat beradaptasi dengan frekuensi serangan dengan sumber serangan lebih dari satu.

2. Dasar Teori

2.1 *Software-Defined Network*

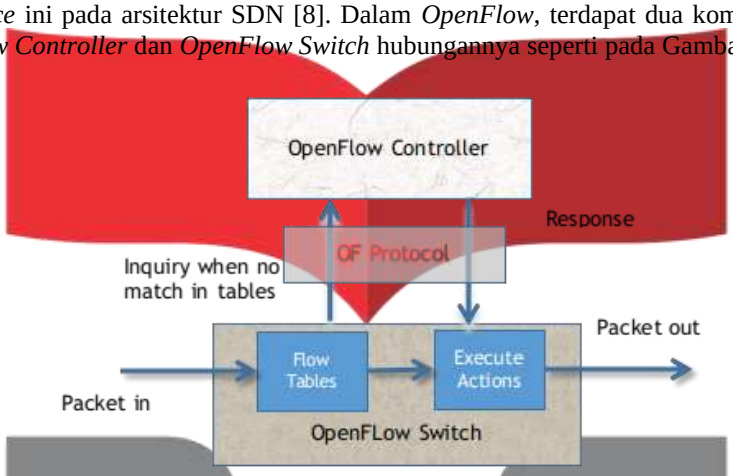
Software Defined Network (SDN) merupakan suatu konsep pendekatan baru dalam arsitektur jaringan. Protokol yang digunakan dalam konsep SDN adalah protokol OpenFlow, dengan konsep dimana Control Plane dan Data Plane dipisahkan[6]. Pada Gambar 1 dapat dilihat gambaran pandangan logis dari arsitektur SDN. Jaringan terpusat pada SDN merupakan perankat lunak, yang bertugas memelihara keseluruhan jaringan. Karena SDN mendukung Application Programming Interfaces (APIs), layanan umum jaringan, seperti routing, multicast, keamanan, kontrol akses, bandwidth management, rekayasa lalu lintas, kualitas layanan, optimalisasi prosesor dan penyimpanan, penggunaan energi dan semua bentuk manajemen kebijakan dapat diimplementasikan untuk memenuhi kebutuhan bisnis[7].



Gambar 1 Arsitektur SDN

2.2 *OpenFlow*

OpenFlow merupakan salah satu dari jenis APIs yang ada dalam jaringan SDN yang digunakan untuk mengatur dan mengontrol *traffic flows* pada *switch* dalam sebuah jaringan. Artinya *Control Plane* berkomunikasi dengan *Data Plane* melalui *OpenFLow*. *OpenFlow* itu sendiri merupakan *southband interface*, komunikasi antara layer *controller* dan *Data Plane* dapat terjadi karena adanya *interface* ini pada arsitektur SDN [8]. Dalam *OpenFlow*, terdapat dua komponen penting yaitu *OpenFlow Controller* dan *OpenFlow Switch* hubungannya seperti pada Gambar 2 [9].



Gambar 2 *OpenFLow Workflow* [9]

2.3 Ryu *Controller*

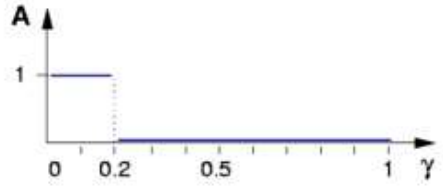
Salah satu *Controller* yang bisa digunakan dalam rangka kerja SDN adalah Ryu[10]. Perangkat lunak dengan API yang disediakan oleh Ryu didefinisikan dengan baik yang membuatnya mudah untuk dikembangkan dalam membuat jaringan baru dan mudah untuk dikontrol. Ryu mendukung berbagai protokol seperti Netconf, OF-Config, dan juga OpenFlow. Bahasa pemograman yang digunakan oleh Ryu adalah bahasa Python dengan lisensi Apache 2.0.

2.4 *Denial of Service* (DoS)

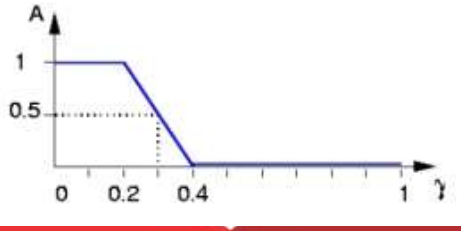
DoS[11] merupakan salah satu bentuk serangan yang bertujuan untuk membuat suatu sistem menjadi tidak dapat melakukan pelayanan kepada *client* untuk diakses. Serangan DoS dapat dilihat dari beberapa upaya untuk membuat suatu sistem menjadi terganggu dalam sementara waktu atau permanen dengan cara membanjiri data dengan jumlah yang banyak. Serangan DoS yang dilakukan oleh lebih dari satu *host* disebut dengan *Distributed Denial of Service* (DDoS).

2.5 *Fuzzy Logic*

Fuzzy logic diperkenalkan oleh Lotfi A. Zadeh, seorang profesor bidang ilmu komputer di *University of California* di Barkeley pada tahun 1965[12]. Pada dasarnya, *fuzzy logic* ini merupakan logika dengan *multi value*, yang berarti logikan ini memungkinkan sebua keadaan memiliki banyak *state value*, tidak seperti logika *boolean* yang dimana hanya terdapat dua kemungkinan yaitu *true or false*, satu atau nol, dan sebagainya. Perbedaan lainnya dapat dilihat dari grafik, yaitu *fuzzy logic* bersifat kontinyu, sedangkan logika *boolean* bersifat diskrit.

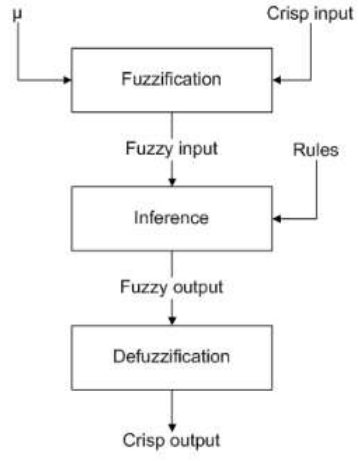


Gambar 3 Grafik yang merepresentasikan logika boolean[12]



Gambar 4 Grafik yang merepresentasikan *fuzzy logic*[12]

Pada Gambar 3, terdapat dua nilai pada sumbu A, yaitu A bernilai 1 jika nilai γ berada di rentang 0 hinga 0,2 dan A bernilai 0 apabila γ berada di rentang lebih besar dari 0,2. Sedangkan pada Gambar 4, terdapat area diantara 0,2 hingga 0,4. ini berarti bahwa pada rentang ini, sumbu A terdapat kondisi yang bernilai antara 0 hingga 1. Adapun diagram algoritma *fuzzy* dapat diilustrasikan pada Gambar 5.

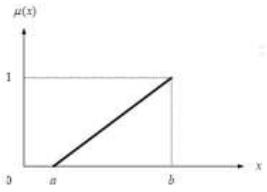


Gambar 5 Diagram Algoritma *fuzzy*[12]

Fuzzy logic terdiri dari *fuzzification, inference*, dan *defuzzification.* Secara garis besar, terdapat dua komponen utama yang dibutuhkan untuk mengembangkan *fuzzy logic*[12]. Pertama, yaitu merancang fungsi keanggotaan untuk setiap parameter yang menjadi masukan atau keluaran dan yang kedua, mendesain *fuzzy rules.* Fungsi keanggotaan adalah representasi grafis dari besarnya kesertaan dari tiap masukan. *Rules* dari *fuzzy logic* menggunakan nilai keanggotaan masukan sebagai faktor pembobotan untuk menentukan pengaruhnya terhadap keluaran.

2.5.1 *Fuzzification*

Fuzzification merupakan langkah pertama pada logika *fuzzy*, dimana *crisp value* diubah menjadi *fuzzy input* yang prosesnya bergantuk pada fungsi keanggotaan. Selain bentuk trapesium yang digunakan pada contoh ini, beberapa fungsi keanggotaan lainnya seperti linear, segitiga, dan sigmorid. Rumus yang digunakan untuk mendapatkan *fuzzy input* untuk tiap jenis fungsi keanggotaan adalah sebagai berikut[13].



$$LinearNaik(x,a,b) = f(x) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a < x \le b \end{cases}$$

$$Sigmoid(x,a,b,c) = \begin{cases} 0, & x < a \\ 2\left(\frac{x-a}{c-a}\right)^2, & a < x \le b \end{cases}$$

Gambar 6 Jenis fungsi keanggotaan dan rumus-rumusnya[13]

2.5.2 *Inference*

Langkah selanjutnya adalah *inference*, yang merupakan proses untuk memperoleh *fuzzy output* berdasarkan *rules*, yang dimana masukan untuk proses ini adalah hasil keluaran dari *fuzzification*. *Rules* merupakan rancangan relasi *if-then* antara seluruh *fuzzy set* antara variabel linguistik yang didapatkan dari hasil *fuzzification* [13].

2.5.3 *Deffuzification*

Hasil dari proses sebelumnya akan diolah pada proses ini, yang dimana hasil akhir dari proses ini adalah *crisp output*. Metode penentuan yang digunakan untuk mendapat *crisp output* adalah metode *weight avearage.*

2.6 *Intrusion Detection System*

IDS berfungsi untuk melakukan deteksi serangan, yang kemudian serangan yang terdektsi tersebut akan ditampung ke dalam berkas *log* lalu memeriksa paket mana saja yang seharusnya dan yang tidak seharusnya berada di dalam jaringan[14]. Hal ini dilakukan dengan cara mencari anomali paket data secara manual. Namun cara ini sangatlah tidak efisien dan memakan sumber daya yang besar, sehingga dikembangkan sistem yang dapat membaca *log* secara otomatis. Untuk melakukan pendeteksian paket data yang berbahaya atau tidak, terdapat dua metode yang digunakan, yaitu *signature-based* dan *anomaly-based*.

2.7 Mininet

Mininet merupakan perangkat lunak simulator jaringan yang digunakan untuk membuat sebuah jaringan virtual realistik, menjalankan kernel, dan berbagai perangkat jaringan lainnya (*switch* atau *router*) di satu mesin (*virtual machine, cloud*, atau juga *native*). Mininet juga mendukung protokol OpenFlow sehingga dapat mensimulasikan arsitektur jaringan SDN[16].

2.8 Snort

Snort merupakan perangkat lunak *network-based* IPS yang bersifat *open source* dengan kemampuan untuk menganalisisi lalu lintas data secara *real-time* dengan menggunakan teknik

signature-based[3]. Snort juga dapat menangani berbagai jenis serangan, diantara lain, yaitu *Denial of Service, port scanning, ARP spoofing,* dsb.

3. Perancangan

Dalam pengerjaan tugas akhir ini, sistem yang akan dibangun adalah sistem keamanaan jaringan yang akan diimplementasikan pada arsitektur jaringan SDN. *Tools* keamanan yang digunakan adalah *network-based* IPS Snort, dengan jenis deteksi *signature-based* yang berfungsi untuk mendeteksi paket berbahaya dalam jaringan. Sedangkan SDN itu sendiri merupakan konsep arsitektur jaringan, yang dimana *control plane* ditempatkan terpisah dengan *data plane*, dan terpusat.

Dengan menggunakan arsitektur SDN, paket yang datang akan diproses oleh *controller* Ryu, yang kemudian akan diperiksa oleh IPS Snort dan dianalisis apakah data tersebut sesuai dengan *rules* yang ada atau tidak. Apabila Snort tidak mendeteksi bahaya dalam paket tersebut, maka paket tersebut akan diteruskan ke Ryu yang kemudian akan dikirim ke alamat tujuan. Namun, jika terdeteksi bahaya dalam paket tersebut, maka proses pengamanan akan dijalankan. Proses pengamanan ini yaitu dengan memberi peringatan dan memblokir sumber dengan waktu pemblokiran yang telah ditentukan. Dengan proses pengamanan ini, paket berbahaya tersebut tidak akan sampai ke *host* tujuan, serta *host* pengirim akan diblokir dalam waktu yang berbanding lurus dengan frekuensi serangan yang dilakukan *host* pengirim.

Sistem yang dibangun ini akan menggunakan *fuzzy logic* untuk menentukan durasi waktu blokir. Apabila *host* penyerang melakukan serangan untuk pertama kalinya, *host* penyerang akan diblokir dalam waktu 10 menit. Akan tetapi, apabila *host* tersebut melakukan serangan lagi, maka waktu pemblokiran akan bertambah sesuai dengan frekuensi serangan yang dilakukan.

3.1 Analisis Kebutuhan Sistem

Sebagai tahap awal sebelum melakukan pengujian, akan dilakukan analisis kebutuhan sistem terlebih dahulu yang akan mensimulasikan jaringan di arsitektur SDN.

3.1.1 Spesifikasi Perangkat Keras

Adapun spesifikasi perangkat keras untuk menjalankan perancangan dan pengujian sistem adalah sebagai berikut:

(Asus X450C)

Sistem Operasi	: Ubuntu 20.04 LTS (Focal)
CPU	: Intel Core i3-3217U @ 1.8 GHz
RAM	: 4 GB
HDD	: 500 GB

3.1.2 Spesifikasi Perangkat Lunak

Berikut ini merupakan spesifikasi perangkat lunak yang digunakan pada tugas akhir ini:

Perangkat Lunak Utama :

1. Linux Ubuntu 20.04 LTS (Focal) sebagai sistem operasi yang akan menjalankan Mininet, Ryu dan Snort.
2. Mininet sebagai emulator jaringan.
3. Ryu sebagai *controller.*
4. Snort sebagai IPS.

Perangkat Lunak Pendukung :

1. Nmap sebagai alat pemindai jaringan atau serangan *Host Discovery*.
2. Hping3 untuk menjalankan serangan DDoS.
3. Python3 sebagai bahasa pemrograman utama dan algoritma *fuzzy*.
4. BASH sebagai *command language* untuk *script* pemblokiran.

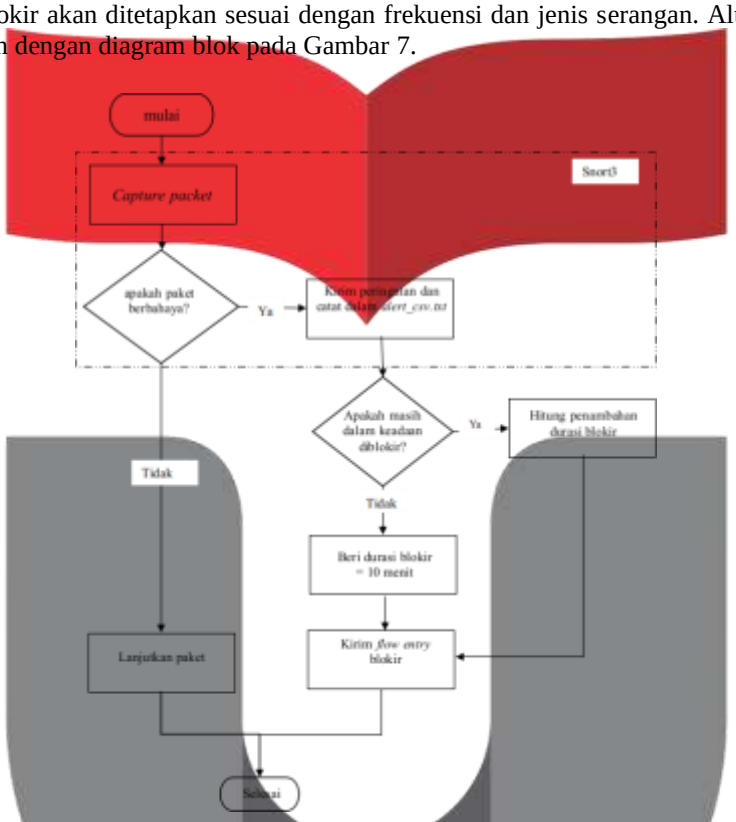
3.2 Rancangan Sistem

Pada sub-bab ini akan dijelaskan mengenai alur kerja sistem, rancangan topologi, serta rancangan *fuzzy* yang digunakan.

3.2.1 Alur Kerja Sistem

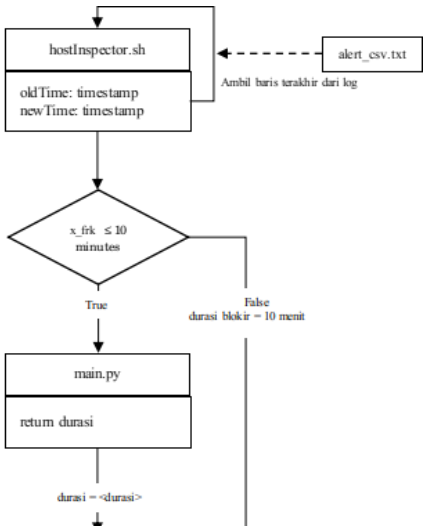
IPS yang akan diinstalasi pada *application plane* adalah Snort dan pada *control plane* akan digunakan Ryu sebagai *controller*. Paket-paket yang datang akan melewati Ryu terlebih dahulu, dan akan diperiksa oleh IPS Snort berdasarkan pencocokan dengan *rules* yang terdapat pada konfigurasi Snort.

Apabila paket yang diperiksa dinyatakan aman, maka Ryu akan langsung meneruskan paket ke alamat tujuan. Akan tetapi, apabila paket yang diperiksa dinyatakan berbahaya, maka Snort akan mengirim peringatan serta akan mengatur durasi waktu pemblokiran *attacker host* yang menjadi sumber paket berbahaya tersebut dengan durasi awal adalah 10 menit. Apabila *attacker host* tersebut masih melakukan serangan dalam kurun waktu kurang dari atau sama dengan 10 menit, maka waktu blokir akan ditetapkan sesuai dengan frekuensi dan jenis serangan. Alur kerja sistem dapat dijelaskan dengan diagram blok pada Gambar 7.



Gambar 7 *Flowchart* alur kerja keseluruhan sistem.

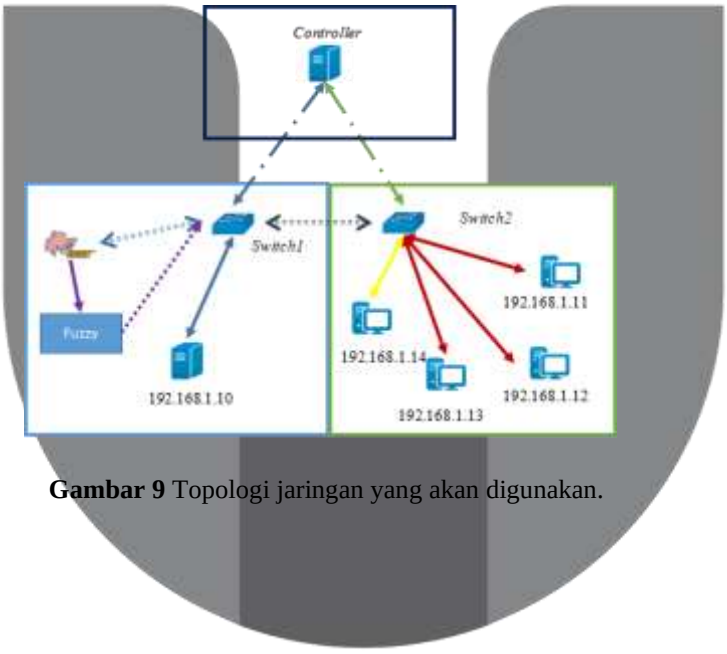
Sedangkan pada Gambar 8, adalah *block diagram* yang menjelaskan komponen-komponen yang dirancang dengan alur prosesnya. Diawali dari berkas *alert_csv.txt* yang adalah hasil keluaran dari IPS Snort yang isinya adalah *tiemstamp*, sid, alamat IP sumber, alamat IP tujuan, protokol yang digunakan, serta pesan. Oleh *hostInspector.sh* akan mengambil baris terakhir dari *alert_csv.txt* dan mencari selisih waktu antara serangan terakhir dengan serangan sebelumnya dari *host* sumber. File *main.py* akan digunakan hanya jika selisih waktu (interval) penyerangan lebih kecil atau sama dengan dari 10 menit. Hasil keluaran dari *main.py* adalah durasi waktu blokir dalam satuan detik yang nantinya akan berfungsi untuk menjeda proses. File *block.sh* berfungsi untuk mengirimkan *flow entry* pemblokiran ke Ryu melalui REST API. File *unblock.sh* akan diaktifkan setelah durasi blokir mencapai nol detik dan akan menghapus *flow entry* pemblokiran.



Gambar 8 *Flowchart* alur kerja sistem fuzzy.

3.2.2 Rancangan Topologi Jaringan

Mininet yang digunakan pada tugas akhir ini akan membangkitkan dua virtual *switch* yang telah mendukung protokol OpenFlow versi 1.3, *switch* pertama akan terhubung dengan *server* serta *switch* kedua, dan untuk *switch* kedua akan terhubung dengan 4 *host* dan *switch* pertama. Adapun penggambaran umum rancangan arsitektur jaringan SDN yang akan dibangun dapat dilihat di Gambar 9, yang dimana akan ada tiga *attacker* serta satu *server* dan satu *victim(client)*.



Gambar 9 Topologi jaringan yang akan digunakan.

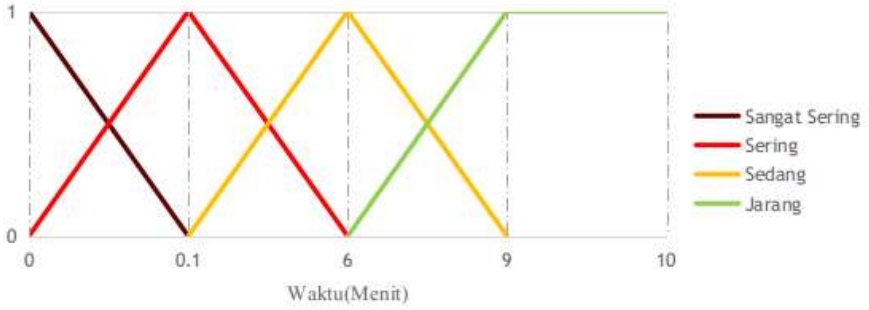
3.2.3 Rancangan Fuzzy

Sistem yang akan dibangun menggunakan *fuzzy logic* untuk menentukan durasi waktu pemblokiran suatu *host* yang melakukan penyerangan. Ada tiga pembagian utama dalam rancangan ini, yaitu *fuzzification, inference,* dan *deffuzification.*

3.2.3.1 *Fuzzification*

Tahap ini merupakan tahap pertama dari tiga tahap pada metode *fuzzy*. Proses yang terjadi pada tahap ini yaitu mengubah *crisp input* menjadi *fuzzy input* dengan menggunakan fungsi keanggotaan.

Dalam sistem ini, akan digunakan dua variabel linguistik yang merupakan hasil keluaran dari Snort3 yang berupa *log* dengan nama *log file* adalah *alert_csv.txt*. Untuk penjelasan fungsi keanggotaan daripada dua variable linguisitik yang digunakan, dapat dilihat sebagai berikut.



Gambar 10 Fungsi keanggotaan frekuensi serangan[5].

Adapun parameter dari masing-masing fungsi keanggotaan frekuensi adalah sebagai berikut:

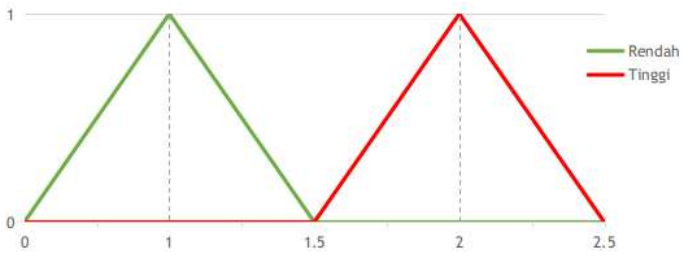
$$\mu \text{ Sangat Sering (x)} = \begin{cases} 1, & x = 0 \\ \frac{0,1 - x}{0,1 - 0}, & 0 < x \leq 0,1 \\ 0, & x \geq 0,1 \end{cases} \quad (1)$$

$$\mu \text{ Sering (x)} = \begin{cases} 1, & x = 0,1 \\ \frac{x - 0}{0,1 - 0}, & 0 < x \leq 0,1 \\ \frac{6 - x}{6 - 0,1}, & 0,1 < x \leq 6 \\ 0, & x \geq 6 \text{ atau } x \leq 0 \end{cases} \quad (2)$$

$$\mu \text{ Sedang (x)} = \begin{cases} 1, & x = 6 \\ \frac{x - 0,1}{6 - 0,1}, & 0,1 < x \leq 6 \\ \frac{9 - x}{9 - 6}, & 6 < x \leq 9 \\ 0, & x \geq 9 \text{ atau } x \leq 0,1 \end{cases} \quad (3)$$

$$\mu \text{ Jarang (x)} = \begin{cases} 1, & 9 \leq x \leq 10 \\ \frac{x - 6}{9 - 6}, & 6 < x \leq 9 \\ 0, & x \leq 6 \end{cases} \quad (4)$$

Gambar 10 diatas merupakan grafik fungsi keanggotaan frekuensi serangan yang akan digunakan, adapun untuk fuzzy set yang ada adalah, Sangat Sering, Sering, Sedang dan Jarang.



Gambar 11 Fungsi keanggotaan tingkat ancaman.

Adapun parameterdari masing-masing fungsi tingkat ancaman adalah sebagai berikut:

$$\mu \text{ Rendah (x)} = \begin{cases} 1, & x = 1 \\ \frac{x-0}{1-0}, & 0 < x \leq 1 \\ \frac{1{,}5-x}{1{,}5-1}, & 1 < x \leq 1{,}5 \\ 0, & x \geq 1{,}5 \end{cases} \quad , x \leq 0 \tag{5}$$

$$\mu \text{ Tinggi (x)} = \begin{cases} 1, & x = 2 \\ \frac{x-1{,}5}{2-1{,}5}, & 1{,}5 < x \leq 2 \\ \frac{2{,}5-x}{2{,}5-2}, & 2 < x \leq 2{,}5 \\ 0, & x \leq 1{,}5 \end{cases} \quad , x \geq 2{,}5 \tag{6}$$

Pada penelitian ini, terdapat dua jenis serangan yang akan dijadikan untuk pengujian, yaitu serangan *Host Discovery* dan serangan DDoS, dan kedua serangan ini masing-masing memiliki tingkat serangan yang berbeda. Adapun untuk fuzzy set, dibagi menjadi dua, yaitu Rendah dan Tinggi. Untuk tingkat ancaman dari masing-masing serangan, *Host Discovery* berada di tingkat rendah, karena hanya merupakan serangan yang bersifat pasif, sedangkan untuk serangan DDoS berada di tingkat tinggi, dikarenakan serangan DDoS merupakan serangan yang bersifat aktif, dan dapat dilancarkan dengan banyak teknik, seperti SYN *flood*, UDP *flood*, bahkan dapat digunakan sebagai *masking* atau *smokescreen* yang disebut dengan *Dark* DDoS, akan tetapi pada penelitian ini, hanya akan digunakan SYN *flood*.

3.2.3.2 *Inference*

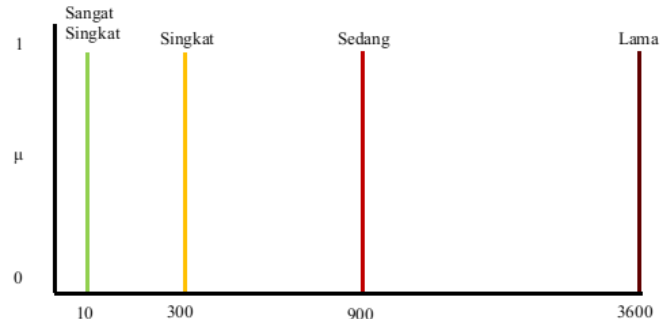
Nilai *fuzzy input* yang didapat dari tahap *fuzzification* akan menjadi masukan bagi tahap kedua ini, yang dimana selanjutnya akan diubah menjadi *fuzzy output* yang hasilnya bergantung pada *fuzzy rules* dengan cara mencocokan hasil keluaran dari proses sebelunnya dengan *rules* yang sudah ditentukan. Adapun untuk *rules* yang ditentukan, ada empat kemungkinan, yaitu Sangat Singkat, Singkat, Sedang, dan Lama. Pada proses *inference* ini, digunakan metode aturan *if-then* dengan model Sugeno.

Adapun *rules* yang akan digunakan dapat dilihat pada tabel 1 dibawah ini:

Tabel 1 Fungsi keanggotaan tingkat ancaman.

no	Frekuensi	Tingkat Ancaman	Waktu Blokir
1	Jarang	Rendah	Sangat Singkat
		Tinggi	Singkat
2	Sedang	Rendah	Singkat
		Tinggi	Sedang
3	Sering	Rendah	Sedang
		Tinggi	Lama
4	Sangat Sering	Rendah	Sedang
		Tinggi	Lama

Dikarenakan pada proses *inference* digunakan model sugeno, maka fungsi keanggotaan keluaran dari proses ini bersifat linear. Adapun parameter keluaran yang digunakan pada sistem ini dapat dilihat pada gambar 12 dibawah ini.



Gambar 12 Parameter keluaran fuzzy.

3.2.3.3 *Deffuzification*

Pada proses ini, hasil keluaran dari proses *inference* akan diproses, hasil dari proses ini adalah *crisp output* yang berupa nilai durasi waktu blokir yang akan diberikan. Metode yang digunakan untuk mengolah nilai keluaran *inference* menjadi *crisp output* adalah dengan metode *weight avearge*, yang dimana, akan diambil nilai rata-rata dengan menggunakan pembobotan derajat keanggotaan, adapun formulasi yang digunakan adalah sebagai berikut:

$$z = \sum \frac{\mu(z)z}{\mu(z)} \qquad (7)$$

3.3 Skenario Pengujian

Dalam pengujian ini, akan digunakan tiga *host* sebagai penyerang, ketiga penyerang ini akan mengirimkan paket berbahaya ke *server*, paket-paket berbahaya yang dikirimkan adalah paket yang sudah dikenali oleh Snort, hal ini dilakukan karena teknik deteksi yang digunakan Snort pada pengujian ini adalah *signature-based*. Pengujian dianggap berhasil apabila sistem yang telah dibangun dapat memberikan waktu blokir sesuai dengan frekuensi serangan dan jenis serangan yang dikirimkan oleh *host* penyerang.

4. Hasil Pengujian dan Analisis

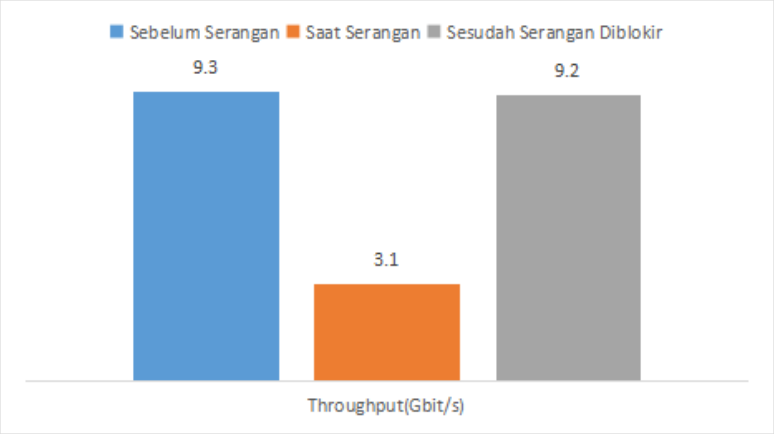
Hasil yang didapat dari pengujian pada penelitian ini dibagi menjadi dua, yaitu hasil pengujian dengan serangan *Host Discovery* dan serangan DDoS, yang dimana pada pengujian yang dilakukan, terdapat tiga *host* penyerang, satu *host client*, serta satu *server*, adapun untuk topologi yang digunakan dapat dilihat pada gambar 9.

Dapat dilihat gambar 13 dibawah ini, waktu blokir yang diberikan sesuai dengan *rules* yang telah ditentukan, sehingga waktu blokir akan sesuai dengan frekuensi serangan serta jenis serangan yang dikirimkan, dari hasil ini dapat disimpulkan bahwa sistem fuzzy yang telah dirancang, dapat membangun *adaptive* IPS dengan baik.

Jarak antar serangan	0	1	2	3	4	5	6	7	8	9	10
DDoS	60	53	45	38	30	23	15	12	8	5	5
Host Discovery	15	14	12	10	8	7	5	3	2	0.18	0.18

Gambar 13 Grafik hasil sistem fuzzy untuk kedua serangan.

Adapun perbandingan kondisi jaringan ketika serangan DDoS terjadi pada saat sebelum serangan, saat serangan dan ketika serangan berhasil di blokir adalah sebagai berikut.



Gambar 14 Grafik perbandingan *throughtput* pada saat serangan DDoS.

Dapat dilihat dari gambar 14, ketika terjadi serangan, *throughput* dari *host client* ke *server* mengalami penurunan dikarenakan adanya serangan DDoS, pada saat serangan berhasil diblokir, *throughtput* kembali normal, hal ini menunjukan bahwa sistem yang telah dirancang dapat menjaga *server* di arsitektur jaringan SDN dengan baik. Pengujian ini dilakukan dengan menggunakan *tools* iperf di *server* dan *host client*

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil analisis dari pengujian sistem yang telah dilakukan untuk mencegah serangan *Host Discovery* dan *Distributed Denial of Service,* didapat beberapa kesimpulan sebagai berikut.

1. IDS Snort3 dapat dikoneksikan dengan salah satu switch yang akan dijaga pada jaringan SDN. *Switch* yang dijaga ini merupakan *switch* yang terhubung dengan *server* langsung dan dijadikan sebagai *firewall*.

2. Pada penelitian ini, sistem Fuzzy yang sudah dirancang dapat bekerja dengan baik, serta pada penelitian ini, digunakan rancangan sistem Fuzzy yang sudah dirancang pada penelitian sebelumnya[5], namun telah dilakukan beberapa perubahan agar dapat bekerja dengan lebih baik dan menghasilkan hasil yang diharapkan, terutama dalam menghadapi serangan DDoS.

3. Sistem yang telah dirancang pada penelitian ini, dapat terhubung dengan IDS Snort dan *adaptive* IPS berhasil dirancang.

4. Berdasarkan hasil dari penelitian yang telah dilakukan, rancangan *adaptive* IPS yang telah dirancang ini dapat beradaptasi dengan frekuensi serangan *Host Discovery* dan serangan DDoS yang datang dengan baik.

5.2 Saran

Saran penulis yang dapat dipertimbankan untuk penelitian rancangan sistem ini selanjutnya adalah sebagai berikut:

1. Melakukan pengujian menggunakan *device* yang sesungguhnya, bukan menggunakan *emulator*, dikarenakan, untuk menguji serangan yang bersifat *volume based attack* seperti serangan DDoS, membutuhkan *resource* yang besar, sehingga, jika dilakukan dalam bentuk *emulasi*, hasil yang didapat belum terlalu maksimal.

2. Menggunakan teknik serangan yang lain. terkhususnya pada serangan DDoS, seperti teknik serangan DDoS menggunakan UDP *flood*, *ping of death*, HTTP *flood*, dll

Referensi

[1] G. Garg dan R. Garg, "Review On Architecture & Security Issues of SDN", *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, pp. 6519-6524, 2014.

[2] X. Tianyi, H. Dijiang dan X. Le, "SnortFlow: A OpenFlow-based Intruison Prevention", *GENI Research and Educational Experiment Workshop*, pp. 89-92,2013.

[3] P. Maheshwari dan A. Garg, "Performance Analysis of Snort-based Intrusion", *International Conference on Advanced Computing and Communication System*, 2016.

[4] W. El-Hajj, F. Aloul dan Z. Trabelsi, "On Detecting Port Scanning using Fuzzy Based intrusion Detection System".

[5] U. Telkom and R. F. Pratama, "Perancangan dan Implementasi Adaptive Intrusion Prevention System (IPS) untuk Pencegahan Penyerangan pada Arsitektur Software-Defined Network (SDN)", 2017

[6] D. Rawat, S. R,- environment, and undefined 2017, Software defined networking architecture, security and energy efficiency: A survey, ieeeplore.ieee.Org, vol 19, no. 1, pp. 32346, 2017

[7] O. N. F. W. Paper, Software-Defined Networking: The New Norm For Networks, 2012.

[8] T. R. Hapsari, Berkenalan dengan OpenFlow, Medium, 2018. [Online]. Available:https://medium.com/core-network-laboratory-techpage/berkenalan-dengan-openflow-3caca9194e51 [Accessed : 06-Aug-2020].

[9] Admin, OpenSwitch vs OpenFlow: What Are They, Whats Their Relationship, Fiber Transceiver, 2018. [Online]. Available:http://www.fiber-optic-transceiver-module.com/openvswitch-vs-openflow-what-are-they-whats-their-relationship.html [Accessed:06-Aug-2020].

[10] "About Ryu", [Online] Available:https://osrg.github.io/ryu-book/en/html/ [Accessed:06-Aug-2020].

[11] R. Das, A. Karabde dan G. Tuna, "Common Network Attack Types and Defense Mechanism", 2015.

[12] Hellman, "Fuzzy Logic Introduction".

[13] Suyanto, Slide Presentasi Kuliah Kecerdasan Buatan: Lesson 2 Reasoning Semester VI 2016.

[14] Vijayarani dan M. Sylviaa, "Intrusion Detection System - A Study". *International Journal of Security, Privacy and Trus Management*,vol. 4, pp. 31-44, 2015.