IMPLEMENTASI ADAPTIVE BLOCKING INTRUSION DETECTION SYSTEM PADA JARINGAN INTERNET OF THINGS BERBASIS SOFTWARE DEFINED NETWORK

IMPLEMENTATION OF ADAPTIVE BLOCKING INTRUSION DETECTION SYSTEM ON INTERNET OF THINGS NETWORK BASED ON SOFTWARE DEFINED NETWORK

Yohanes Armenian Putra[1], Ridha Muldina Negara[2], Rahmat Yasirandi[3]
[1,2,3]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[1]**yohanesarmenian@student.telkomuniversity.ac.id**, [2]**ridhanegara@telkomuniversity.co.id**,
[3]**batanganhitam@telkomuniversity.ac.id**

Abstrak

***Internet of Things* (IoT) merupakan sebuah konsep yang dapat membantu manusia, karena dengan IoT manusia dapat memantau suatu keadaan maupun objek dan dapat melakukan pengontrolan terhadap objek tersebut dari jarak jauh. Dengan munculnya *Software Defined Network* (SDN) sebuah konsep dalam jaringan yang melakukan pemisahan antara data plane dan control plane sehingga jaringan menjadi lebih fleksible untuk dikelola. Kebutuhan kecepatan komunikasi data dan ketersediaan yang tinggi untuk melayani perangkat IoT dibutuhkan jaringan yang kuat, stabil dan dapat di desain sesuai kebutuhan seperti SDN. Saat ini SDN masih memiliki kekurangan pada sisi keamanan, seperti serangan *Denial of Service* (DoS) yang menyerang ketersediaan dari jaringan sehingga jaringan tersebut tidak dapat melayani permintaan atau dengan kata lain disebut *down*.**

Dengan penggunaan *Intrusion Detection System* (IDS) deteksi serangan pada jaringan dapat dilakukan, tetapi IDS masih memiliki kekurangan yaitu tidak dapat melakukan blokir pada *host* penyerang. Pemanfaatan algoritma *fuzzy* digunakan untuk melakukan bloking kepada *host* penyerang dan melakukan pengaturan waktu *blocking* dengan melihat banyak-nya frekuensi serangan dengan interval tertentu dan jenis serangan yang terdeteksi sehingga waktu *block host* akan disesuaikan dengan tingkat banyaknya frekuensi serangan dan jenis serangan yang terdeteksi.

Dari hasil pengujian yang telah dilakukan sebanyak 5 kali percobaan serangan dengan interval serangan 0.1 menit sampai 10 menit didaptkan waktu rata- rata blokir dan mendapatkan hasil akurasi algoritma *fuzzy* sebesar 50 % untuk serangan DoS dan 75% untuk *host discovery*. Perbandingan hasil pengujian *Quality of Service* (QoS) dengan menggunakan sub-sistem kemanan dan tidak menggunakan sub-sistem keamanan yang memberikan hasil parameter *jitter*, *delay*, *throughput* dan paket *loss*, pada serangan DoS didapatkan hasil penurunan *delay* sebesar 5,3471 detik dan 0 % paket loss dengan menerapkan sub-sistem *adaptive* IDS pada jaringan IoT berbasis SDN.

Kata Kunci: ***Internet of Things, Software Defined Network, Denial of Service, Intrusion Detection System, Fuzzy***

Abstract

Internet of Things (IoT) is a concept that can help humans, because with IoT humans can monitor a state or object and can control these objects remotely. With the emergence of a Software Defined Network (SDN), a concept in a network that separates the data plane and the control plane so that the network becomes more flexible to manage. The need for data communication speed and high availability to serve IoT devices requires a network that is strong, stable and can be designed according to needs such as SDN. Currently SDN still has shortcomings on the security side, such as Denial of Service (DoS) attacks which attack the availability of the network so that the network cannot serve requests or in other words it is called down

Using Intrusion Detection System (IDS) attack detection on the network can be done, but IDS still has disadvantages, namely it cannot block the attacking host, IDS only detection the attack and gift notification to the network administrator. The use of fuzzy algorithms is used to block attacking hosts and time blocking by looking at the frequency of attacks at

certain intervals and the types of attacks detected so that the host block time will be adjusted according to the level of the frequency of attacks and the types of attacks detected.

From the test results that have been carried out 5 times attack trials with an attack interval of 0.1 minutes to 10 minutes, the average blocking time is obtained and the accuracy of the fuzzy algorithm is 50% for DoS attacks and 75% for host discovery. Comparison of the results of the Quality of Service (QoS) test using the security sub-system and not using the security sub-system which gives the results of the parameters of jitter, delay, throughput and packet loss, the DoS attack results in a reduction in delay of 5.3471 seconds and 0% packet loss by implementing the adaptive IDS sub-system on an SDN-based IoT network.

Keywords: Internet of Things, Software Defined Network, Denial of Service, Intrusion Detection System, Fuzzy

1. Pendahuluan

Kemajuan teknologi dalam industri saat ini mulai berkembang cepat dan mulai memasuki revolusi industri 4.0. Dengan revolusi industri 4.0 menjadikan *Internet of Things* (IoT) sebagai salah satu elemen dasar didalamnya. IoT memanfaatkan jaringan internet untuk melakukan komunikasi antar perangkat yang terhubung. Untuk memantau dan mengkontrol keadaan sekitar manusia [2]. IoT merupakan sebuah konsep komunikasi yang dapat menghubungkan apa saja dengan siapa saja, dengan tidak dibatasi jarak dan waktu. Perkembangan IoT yang semakin massif dan heterogen yang membuat IoT menjadi riskan pada sisi keamanannya [3]. Pada penelitian yang dilakukan pada penelitian [4] serangan yang dapat terjadi pada IoT adalah *Denial of Service* (DoS) yang menyerang *availability*, serangan atau ancaman tersebut terjadi karena masalah pada skalabilitas dan heterogeneity dari IoT [4]. IoT memiliki tiga layer, yaitu *Preception layer, Network layer* dan *Application layer*, serangan DoS yang terjadi pada jaringan IoT dapat terjadi pada *layer network* [5].

Software Defined Network (SDN) adalah sebuah konsep dalam jaringan yang melakukan pemisahan antara control plane dan data plane dengan menggunakan protokol OpenFlow, konsep tersebut mendukung inovasi untuk operator jaringan dapat membangun bentuk arsitektur sesuai dengan sumber daya dan kebutuhan yang ada, dengan SDN pemrograman pada jaringan dapat dilakukan [1]. Pada penelitian [6] salah satu acaman yang sering dan dapat terjadi adalah DoS pada perangkat forwading atau yang biasa di sebut data plane, DoS yang terjadi menyerang availability pada jaringan SDN. Pada penelitian [7] melakukan riset mengenai pengamanan jaringan IoT menggunakan SDN dengan menggunakan *Network Intrusion Detection System* (NIDS) yang melakukan pemantauan traffic IoT dengan melihat anomaly detection. Implementasi SDN pada jaringan IoT diharapkan dapat memberikan solusi kelemahan dari jaringan IoT yang sudah ada. Penelitian tentang serangan *Distributed Denial of Service* (DDoS) pada jaringan SDN yang dijelaskan pada [8]. DoS melakukan serangan tidak hanya pada host target namun juga *link network* pada jaringan tersebut. Sehingga jaringan tidak dapat memberikan layanan kepada pengguna. Terdapat beberapa cara yang dapat digunakan untuk mengatasi DDoS maupun *Denial of Service* (DoS) seperti pada penelitian [9] dan [10] dengan menggunakan snort yang berbasis *Intrusion Prevention System* (IPS) dan Intrusion Detection System (IDS) yang diaplikasikan pada jaringan SDN. Pada penelitian [7] dilakukan riset mengenai penggunaan algoritma *fuzzy* dan IDS untuk melakukan pengamanan pada jaringan, hasil yang didapatkan penggunaan algoritma fuzzy memberikan effisiensi pada pengurangan 20% hingga 30% baris kode dibandingkan dengan menggunakan keamanan SDN konvensional [7].

Pada Tugas Akhir sebelumnya berjudul "Implementasi Pengontrolan Terpusat Gateway Long Range (LORA) *Internet of Things* berbasis *Software Defined Network*" berfokus pada penggunaan jaringan SDN untuk traffic IoT. Dalam penelitian lanjutan ini penulis berfokus pada pencegahan DoS dengan melakukan blokir pada *host* penyerang dengan jenis serangan DoS yaitu *synchronization Attack* (SYN *Attack*), namun kekurangan dari snort IPS adalah waktu blokir yang statis tidak mempedulikan frekuensi serangan dari suatu node pada jaringan, sehingga akan digunakan logika *fuzzy* seperti pada penelitian [11] dan [12] dengan melakukan pengambilan data log dari snort IDS lalu data akan diteruskan pada sistem fuzzy yang akan menentukan waktu blokir dan akan di block dari jaringan SDN oleh *controller*.

2. Dasar Teori /Material dan Metodologi/perancangan

2.1 *Internet of Things*

IoT merupakan jaringan yang menghubungkan berbagai perangkat yang saling bertukar informasi tanpa adanya interfrensi dari campur tangan manusia atau disebut *machine to machine communication* (M2M), jaringan IoT merupakan jaringan yang dapat dibangun dengan arsitektur *wireless* maupun *wired*. Komunikasi antara perangkat IoT diawali dari perkembangan perangkat komunikasi yang semakin banyak dan kebutuhan manusia untuk melakukan berbagai pekerjaan dalam satu waktu, IoT dapat membantu dalam menyambungkan komunikasi antara smartphone, laptop dan perangkat lain dengan perangkat IoT yang dipasang pada lingkungan *e-learning*, rumah, kantor, bidang kesehatan, industri dan yang lainnya [13].

2.2 *Software Defined Network* (SDN)

Software Defined Network merupakan sebuah konsep yang melibatkan perangkat – perangkat jaringan seperti *switch*, *router* serta perangkat lainnya yang melakukan pemisahan cara kerja dalam jaringan, dua bagian pemisah tersebut yaitu *control plane* dan *data plane*. *Flow data* pada SDN dapat diprogram sesuai dengan kebutuhan dari jaringan itu sendiri [10].

2.3 *Intrusion Detection Sytem* (IDS)

IDS merupakan sebuah perangkat lunak yang dapat melakukan *monitoring* pada trafik jaringan. IDS akan memberikan peringatan kepada *administrator* jaringan bila menemukan kejangalan pada aktifitas jaringan yang dianggap berbahaya atau mencurigakan. Namun IDS hanya melakukan deteksi tanpa adanya tindakan pencegahan yang dilakukan [19].

2.4 *Snort*

Snort merupakan sebuah perangkat lunak yang berbasis open-source pada IPS dan IDS selain itu *snort* dapat digunakan untuk melakukan *sniffer*, *packet logger* maupun data analisis [20]. Snort memiliki kemampuan yang sangat baik untuk mendeteksi bila ada paket yang berbahaya maupun mencurigakan karena berbasis pada kemanan jaringan IPS dan IDS, pada *snort* IDS hanya melakukan deteksi paket tanpa men-*drop* paket bila paket tersebut berbahaya maupun mencurigakan, sedangkan pada snort IPS terdapat kombinasi dari snort dan *Iptables* sehingga bila ada paket yang masuk kedalam sistem dan terdeteksi berbahaya maupun mencurigakan akan langsung di *drop* [22]. Dari kelebihan dan kekurangan pada kedua jenis snort IPS dan IDS tersebut mulai dikembang pada NIDS yang dapat mengabungkan kelebihan dari kedua *snort* tersebut [23].

2.5 *Logika Fuzzy*

Fuzzy merupakan logika yang diperkenalkan oleh seorang professor dibidang ilmu komputer University of California yaitu Lotfi A. Zadeh pada tahun 1965 [24]. Logika *fuzzy* merupakan logika yang dibuat agar komputer dapat membuat keputusan dari sebuah masalah dengan formula matematika untuk mengambil keputusan seperti manusia dalam sebuah program komputer, tidak seperti logika tradisional (*Boolean*) yang hanya mengenal benar atau salah, iya atau tidak dan tinggi atau rendah. Fuzzy terdiri dari tiga proses yaitu *fuzzification*, *inference* dan *defuzzification*.

2.6 *Denial of Service* (DoS)

DoS secara langsung dan spesifik menyerang target tertentu seperti perangkat pada jaringan dengan tujuan membuat jaringan menjadi *down* dan tidak dapat memberikan layanan [8]. Dengan melakukan pengiriman paket secara terus menerus DoS menyerang jaringan, sehingga jaringan akan kewalahan dalam membangun komunikasi dan ketersediaan serta *resource* pada jaringan terpakai sepenuhnya akibat serangan tersebut [19].

2.6 InfluxDB dan Grafana

InfluxDB merupakan sebuah database yang bersifat open-source dengan berbasis bahasa GO, InfluxDB digunakan untuk mengumpulkan data matrik dari sensor – sensor IoT dan perintah pada InfluxDB memiliki kesamaan dengan SQL. Didalam InfluxDB terdapat Telegraf yang merupakan database yang menjadi subscriber dari MQTT dan melakukan publish pada InfluxDB [20]. Grafana merupakan perangkat lunak yang berbasis *open-source* yang sering digunakan untuk melakukan monitoring dan analisis data, Grafana mendukung penggunaan dengan banyak source

data dan dapat memberikan monitoring serta analisis data secara real-time. Grafana mendukung penggunaan basis data seperti Graphite, Prometheus, InfluxDB, ElasticSearch, MySQL, PostgreSQL, data yang dibaca oleh Grafana merupakan data matrik [21].

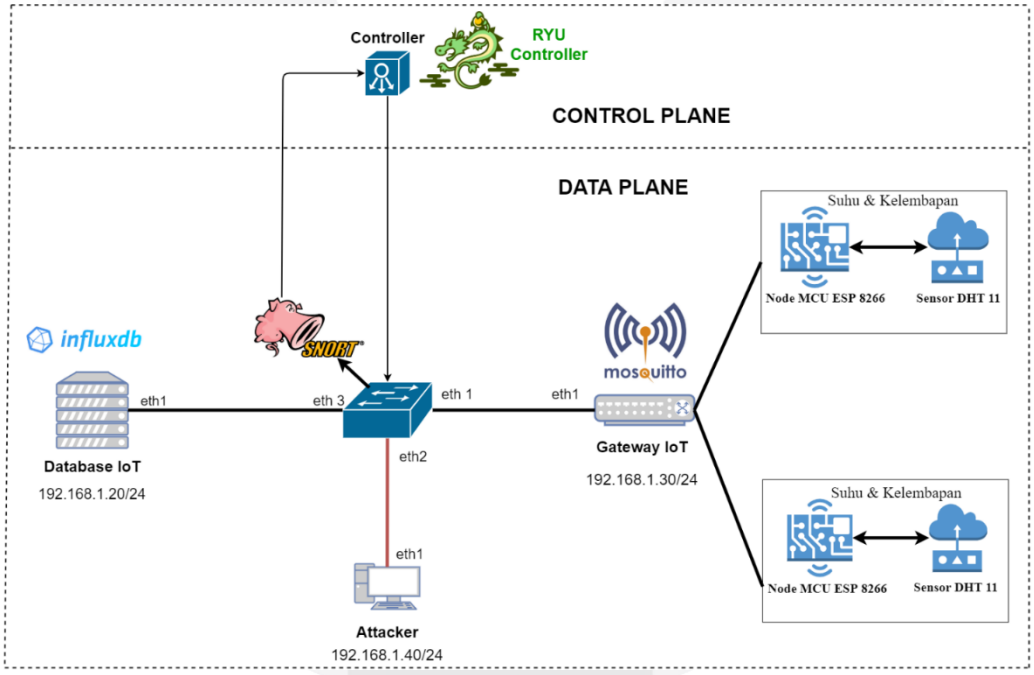
2.7 *Message Queuing Telemetry Transport* (MQTT)

MQTT merupakan protokol yang berjalan pada protokol TCP, kelebihan dari MQTT adalah penggunaan daya yang minimum serta ringan yang membuat protokol MQTT mendukung penggunaan IoT [25]. MQTT mulai diterapkan sekitar tahun 1999 dalam penggunaan protokol MQTT terdapat empat elemen yang penting yaitu subscriber dan publish, pesan, topic dan broker. Subscribe dan publish merupakan proses dimana perangkat IoT saling berkomunikasi. Pesan merupakan isi dari data dari IoT. Topic adalah alamat kemana data akan di kirim, seperti topic kelembapan dan suhu dari perangkat IoT akan di kirimkan kepada perangkat lain menuju topik yang sama. Broker bertanggung jawab terhadap seluruh aktifitas data IoT yang berjalan dalam jaringan dengan menggunakan protokol MQTT [22].

3. Pembahasan

3.1. Desain Sistem

Sistem yang dirancang pada Tugas Akhir ini digunakan untuk mendeteksi paket yang masuk kedalam jaringan SDN dengan adaptive IDS untuk melayani trafik data IoT. Sistem kemanan yang dibangun pada jaringan menggunakan adaptive IDS yang dapat mendeteksi serangan dengan metode *signature-based*. Penyerang atau attacker akan mengirim paket yang akan terekam pada log snort, bila terdapat gejala paket data yang terekam sesuai dengan aturan yang telah di tetapkan pada snort maka paket – paket selajutnya yang dikirim oleh penyerang akan di *drop* sehingga serangan dapat dicegah. Desain sistem seperti pada gambar 3.1.



Gambar 3.1 Desain Sistem

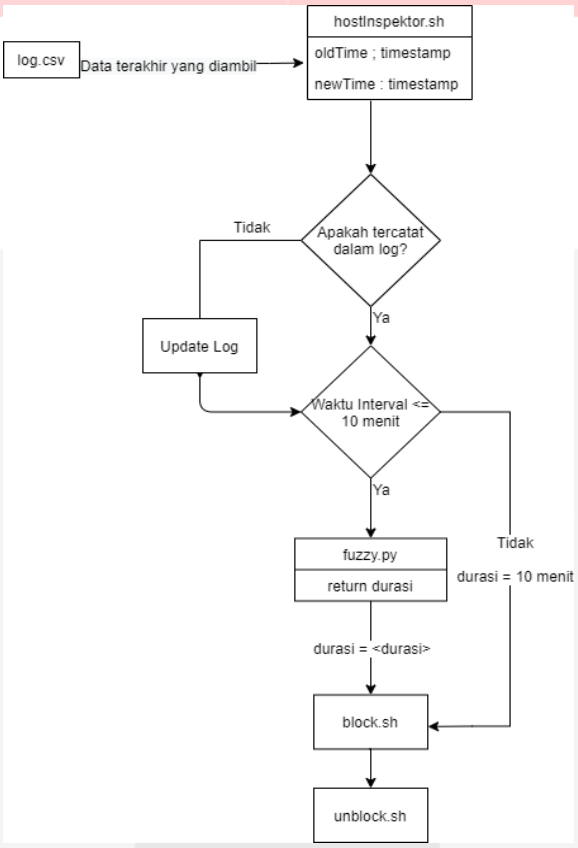
Pada gambar 3.1 Sistem jaringan SDN pada Tugas Akhir ini menggunakan 3 personal computer (PC) dan 1 mini PC Intel NUC. Pada PC database terdapat beberapa aplikasi yang digunakan yaitu Grafana dan InfluxDB yang berfungsi untuk menerima data IoT dan menampilkan data dalam bentuk grafik real-time. PC gateway IoT terdapat aplikasi MQTT broker sebagai penghubung antara data IoT dengan aplikasi python yang akan mengirimkan data IoT dalam bentuk data TCP. Pada PC *controlle*r ada beberapa aplikasi yang dipasang diataranya *OpenVswitch*, *Snort*, RYU dan IDS *adaptive*. Pada Intel NUC akan digunakan sebagai penyerang dan terdapat beberapa

aplikasi serangan diataranya Nmap untuk melakukan dan Hping3. Perangkat yang dilindungin merupakan database IoT yang menampung data IoT dengan bantuan SDN.

3.2. Alur Kerja Sistem

Paket data yang masuk dan telah di *capture* oleh sistem akan kembali di periksa isi dari payload yang terdapat pada paket tersebut, bila *payload* pada paket merupakan IPv4 *payload* maka akan dilanjutkan pada pemeriksaan IP sumber dan IP tujuan dari paket tersebut. Kemudian dalam pemeriksaan dari log data yang sudah ada paket tersebut sudah melakukan request dengan jumlah frekuensi dan interval waktu yang melebih dari yang ditentukan paket akan dikategorikan kedalam bentuk paket berbahaya, selanjutnya paket akan dibuang atau di drop dan log data akan diperbaharui atau update untuk melakukan *blocking*. Algoritma *Fuzzy* digunakan untuk menentukan lama waktu blocking paket yang diangap sebagai Nmap dan DoS. Pada gambar 3.2 merupakan skema dari penerapan *blocking* yang dilakukan.

Gambar 3.2 Diagram Alir Percobaan *Blocking*



Gambar 3.3 Alur Cara Kerja Sistem

Kemudian dalam alur cara kerja dari sistem, seperti pada gambar 3.3. Paket yang terdeksi oleh snort akan terekam dan tercatat pada log yang diberi nama alert.cvs. log tersebut berisi informasi mengenai *timestamp, source* IP, *destination* IP, *Protocol* dan pesan infromasi dalam paket tersebut. Paket yang sudah masuk kedalam log akan diangap sebagai paket berbahaya, kemudian pada *hostInspector*.sh akan dicari selisih waktu dari waktu penyerangan sebelumnya dengan penyerangan yang terbaru dari *host* yang sama, bila selisih waktunya kurang atau sama dengan 5 menit maka akan masuk ke dalam fuzzy.py dimana akan dilakukan perhitungan dengan menggunakan algoritma *fuzzy* dan waktu block paket akan disesuaikan. Jika hasil selisi waktu penyerangan lebih dari 5 menit, maka waktu *block* paket tersebut secara default akan ditetapkan hanya 5 menit tanpa melalui proses algoritma *fuzzy*, kemudian akan di tangani oleh *block*.sh setelah

waktu *block* habis maka akan ditangani oleh *unblock*.sh dimana data paket tesebut akan dihapus dari *log database* dan tidak akan diproses lagi.

3.4. Skenario Pengujian

Pengujian dilakukan menggunakan serangan DoS dan *port scanning* dengan menggunakan beberapa tools pada sisi attacker seperti Nmap dan Hping3 yang di pasang pada sistem operasi Ubuntu 16.04. Tools tersebut melakukan serangan dengan menggunakan metode serangan TCP *flooding*. Kemudian pada sisi Controller digunakan tool Prometheus untuk melakukan monitoring memory dan CPU usage pada *host controller*. Pada pengujian ini *adaptive* IDS mengamankan database IoT yang menampung data IoT dengan bantuan jaringan SDN. Pengamanan pada *database* IoT dilakukan karena data IoT yang ditampilkan pada database berbasis website yang dapat diakses *localhost* ataupun publik sehingga memungkinkan adanya serangan DoS pada database IoT jika data dapat diakses secara public.

Tahap pertama skenario pengujian dengan melakukan serangan dari *attacker* kepada *Database* dengan mengirim paket ACK untuk melakukan DoS dengan metode SYN *flooding* menggunakan *tools* hping3 untuk mendapatkan data *Quality of Service* (QoS) tanpa menggunakan sistem *Adaptive* IDS dan algoritma *fuzzy*.

Skenario kedua melakukan serangan dari *attacker* kepada *Database* dengan mengirim paket ACK untuk melakukan serangan DoS dengan metode SYN *flooding* untuk medapatkan data pada log snort IDS dengan menggunakan Adaptive IDS dan algoritma *fuzzy*. Lalu skenario pengujian terakhir melakukan perbandingan hasil IDS dengan dan tanpa algoritma fuzzy, dengan melakukan perbandingan *Quality of Service* (QoS) dengan melakukan perhitungan parameter *delay*, *throughput*, *jitter*, *packet loss* dari kedua skenario pengujian.

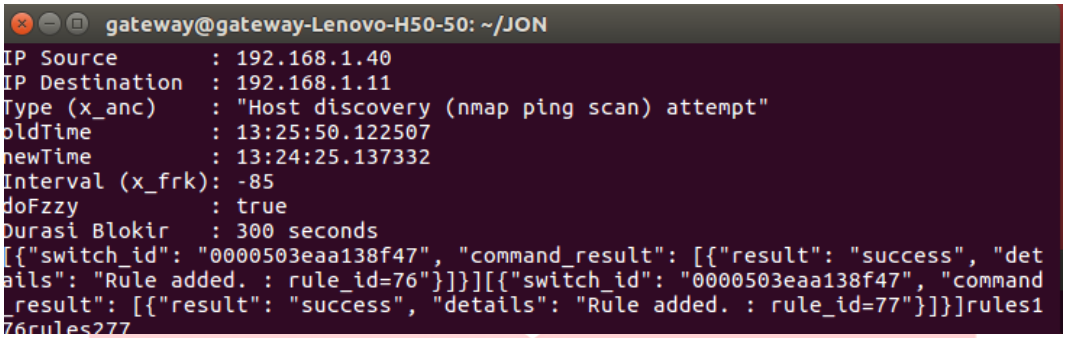
Skenario ketiga melakukan uji kemanan sistem yang telah dibuat dengan melihat lama waktu blokir dan pengaruh algoritma fuzzy yang digunakan pada jaringan SDN yang menangani data IoT dari serangan DoS dan *port scanning*.

3.4.1. Pengujian Fungsionalitas Aturan dan Validasi IDS

Pada Tugas akhir ini sistem dibuat untuk mendeteksi DoS dan *host discovery* dengan menggunakan algoritma fuzzy menggunakan bahasa pemrograman Python. Algoritma *fuzzy* yang digunakan melakukan deteksi serangan dengan bantuan snort sebagai IDS kemudian algoritma *fuzzy* akan mendeteksi jenis serangan dan banyaknya frekuensi serangan untuk menentukan lamanya waktu blokir dari *host* penyerang. Gambar 3.4 dan Gambar 3.5 menunjukan deteksi serangan DoS dan host discovery yang terdeteksi oleh snort kemudian di blokir oleh algoritma *fuzzy*.

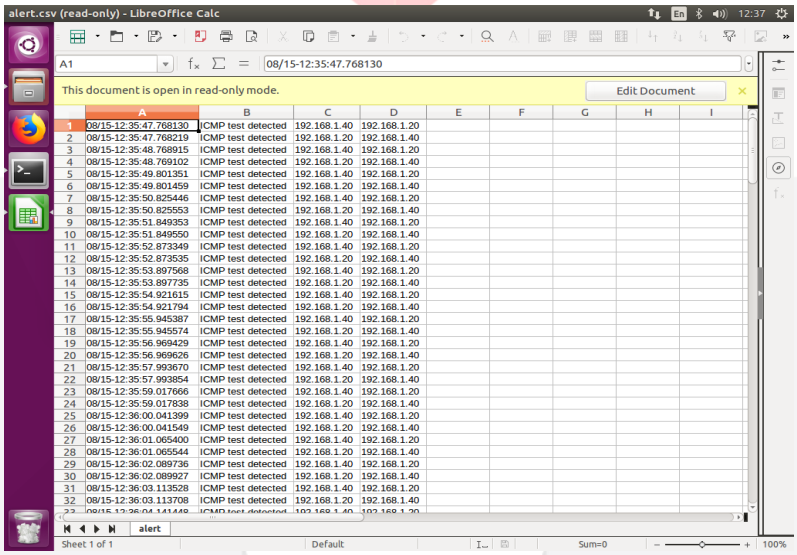
```
IP Source       : 192.168.1.40
IP Destination  : 192.168.1.20
Type (x_anc)    : "TCP SYN packet flooding (simple or distributed) attempt"
oldTime         :
newTime         : 17:44:59.030165
Interval (x_frk): 63899
doFzzy          : false
Durasi Blokir   : 300 seconds
[{"switch_id": "0000503eaa138f47", "command_result": [{"result": "success", "details": "Rule added. : rul
e_id=14"}]}][{"switch_id": "0000503eaa138f47", "command_result": [{"result": "success", "details": "Rule
added. : rule_id=15"}]}]rules1rules2
```

Gambar 3.4 Serangan DoS terdeteksi



Gambar 3.5 Serangan *host discovery* terdeteksi

Snort digunakan sebagai IDS untuk melakukan deteksi serangan yang ada pada jaringan, serangan yang terdeketsi oleh IDS akan dicatatat pada log yaitu alert.cvs kemudian log tersebut akan dibaca oleh algoritma fuzzy dengan bantuan program bash. Seperti pada gambar 3.6 merupakan log serangan yang terdapat pada alert.cvs. dapat dilihat pada tabel 1 waktu serangan pada serangan ping yang di anggap sebagai “ICMP test detected” sama dengan hasil yang keluar pada adaptive IDS.



Gambar 3.6 Log serangan pada *alert*.cvs

Gambar 3.7 serangan yang terdeksi oleh fuzzy melalui snort

3.4.2. Analisis QoS

Analisis parameter QoS yang sudah didapatkan pada percobaan serangan pada saat tidak menggunakan kemanan dan saat menggunakan kemanan menghasilkan perbandingan seperti pada tabel 3.1 merupakan QoS pada serangan DoS dan tabel 3.2 merupakan QoS pada serangan *host discovery*.

Tabel 3.1 QoS serangan DoS

Pengujian parameter	Hasil Sebelum Menggunakan Kemanan	Hasil Setelah Menggunakan Keamanan
Packet loss	99.99 (%)	0 (%)
Jitter	1,52 (s)	$1{,}15 \times 10^{-6}$ (s)
Througtput	2001,709 (Kbps)	$2{,}67 \times 10^{-4}$ (Kbps)
Delay	5,39 (s)	$3{,}29 \times 10^{-2}$ (s)

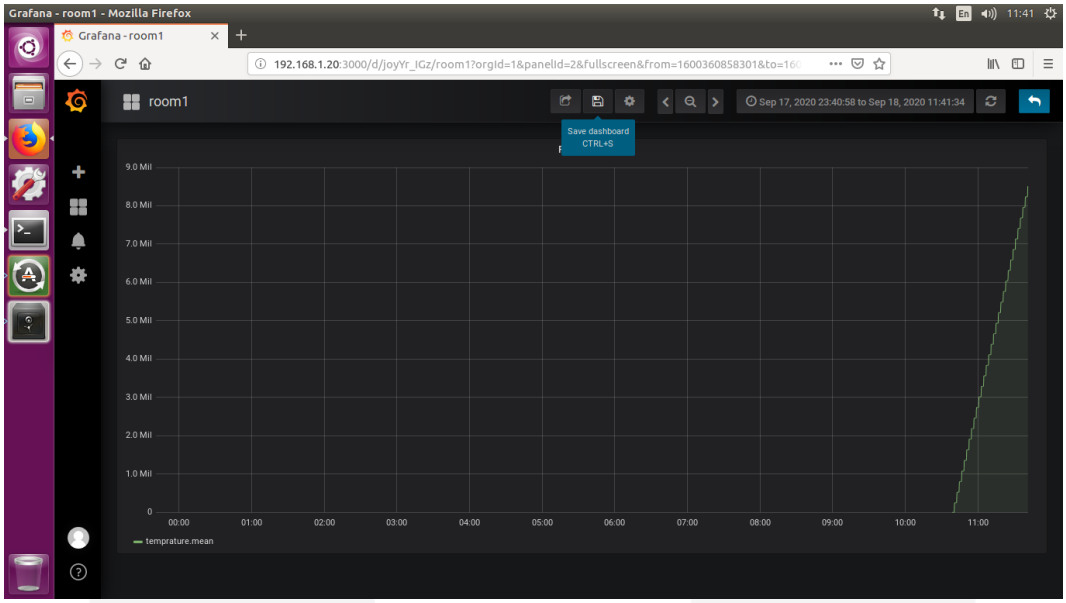
Tabel 3.2 QoS serangan *host discovery*

Pengujian parameter	Hasil Sebelum Menggunakan Kemanan	Hasil Setelah Menggunakan Keamanan
Packet loss	0 (%)	0 (%)
Jitter	$0{,}57 \times 10^{-5}$ (s)	$0{,}55 \times 10^{-6}$ (s)
Througtput	$4{,}51 \times 10^{-1}$ (Kbps)	$6{,}25 \times 10^{-4}$ (Kbps)
Delay	$1{,}64 \times 10^{-2}$ (s)	$1{,}65 \times 10^{-2}$ (s)

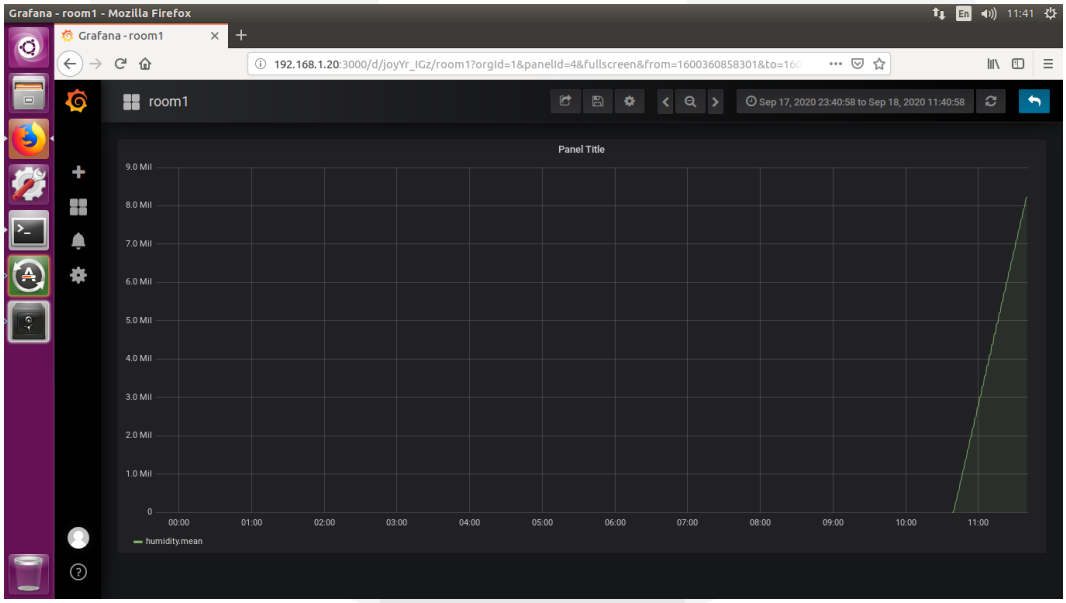
Pada tabel 3.1 dan tabel 3.2 dapat disimpulkan saat menggunakan kemanan packet *loss* saat serangan DoS menjadi 0% dari 99,99% saat tidak menggunakan kemanan. *Delay* pada serangan DoS mengalami penurunan sebesar 5,3471 detik sedangkan pada serangan *host discovery* waktu *delay* yang didapatkan hanya berbeda 0,0001 detik saat menggunakan kemanan dan tidak menggunakan kemanan. *Throughput* dari serangan DoS saat belum menggunakan kemanan lebih besar dari pada saat menggunakan kemanan dikarenakan data yang diterima merupakan data serangan sehingga menghasilkan *throughput* yang lebih besar saat tidak menggunakan kemanan.

3.4.2. Dashboard Sistem *Monitoring*

Pada gambar 3.8 dan gambar 3.9 merupakan tampilan dari dashboard sistem *monitoring* terpusat yang di tampilkan pada *database* IoT sebagai *host* yang menerima data IoT yang dikirimkan oleh sensor melalui *Gateway* IoT, pada *dashboard* sistem monitoring terdapat dua informasi yang ditampilkan yaitu topik suhu dan topik kelembapan yang di tampilkan secara *realtime* tanpa *delay*.



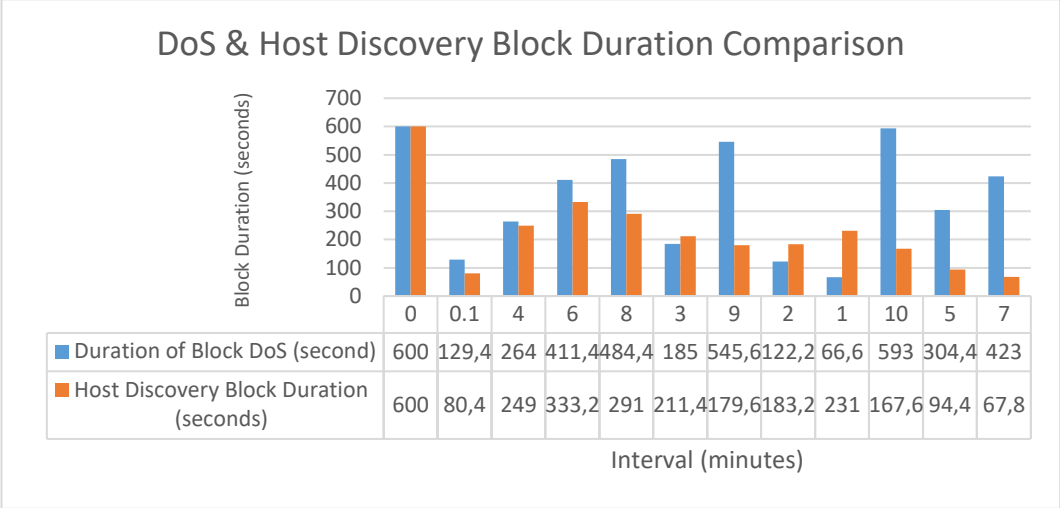
Gambar 3.8 Sistem *Monitoring* IoT menampilkan data Suhu



Gambar 3.9 Sistem *Monitoring* IoT menampilkan data Kelembapan

3.4.2. Waktu Blokir Serangan

Pada pengujian ini dilakukan pengujian lama waktu pemblokiran serangan dengan menggunakan adaptive IDS yaitu snort sebagai IDS dan algoritma *fuzzy* sebagai keamanan yang melakukan pemblokiran serangan. Lama durasi blokir *host* yang melakukan serangan ditentukan dari jenis serangan dan banyaknya frekuensi serangan dalam waktu interval yang sudah ditentukan, pada pengujian ini dilakukan lima kali percobaan dengan interval waktu 0.1 menit sampai 10 menit. Gambar 3.10 menunjukan grafik waktu blokir dari serangan DoS dan *host discovery*.



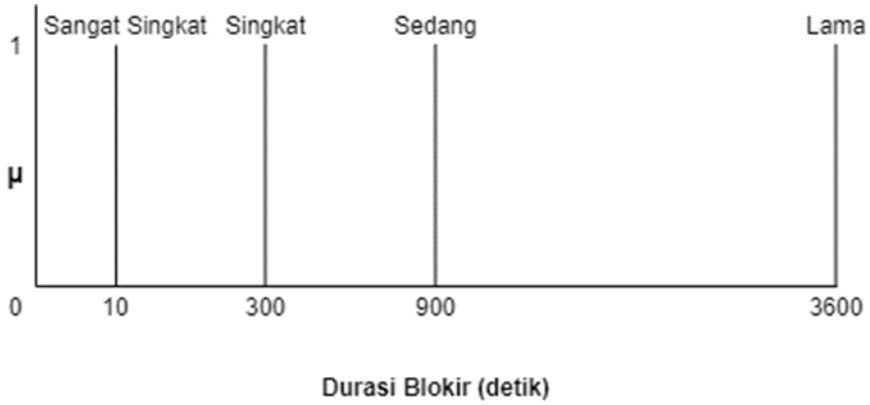
	0	0.1	4	6	8	3	9	2	1	10	5	7
Duration of Block DoS (second)	600	129,4	264	411,4	484,4	185	545,6	122,2	66,6	593	304,4	423
Host Discovery Block Duration (seconds)	600	80,4	249	333,2	291	211,4	179,6	183,2	231	167,6	94,4	67,8

Gambar 3.10 Durasi Blokir Dos dan *host discovery*

Pada gambar 3.10 dapat dilihat bahwa durasi blokir dari serangan DoS lebih tinggi dibandingkan dengan durasi blokir serangan *host discovery* sesuai dengan aturan yang ada pada proses *fuzzification* seperti yang ada pada table 3.3 serta pemodelan Sugeno untuk melakukan *deffuzification* pada Gambar 3.11.

Tabel 3.3 Aturan *Fuzzification*

Interval (menit)	Frekuensi Serangan	Tingkat Serangan	
		Rendah	Tinggi
6 - 10 menit	Jarang	Durasi Sangat Singkat	Durasi Sedang
0,1 - 9 menit	Sedang	Durasi Singkat	Durasi Sedang
0, 1 - 6 menit	Sering	Durasi Singkat	Durasi Lama
0 - 0,1 menit	Sangan Sering	Durasi Sedang	Durasi Lama



Gambar 3.11 Durasi Blokir

Didapatkan hasil durasi blokir serangan yang sesuai dengan aturan yang telah di tentukan pada proses *fuzzification* seperti pada tabel 3.4 untuk serangan DoS dan tabel 3.5 pada serangan host discovery. Kemudian dari data kedua tabel rata – rata hasil serangan yang telah di lakukan percobaan sebanyak 5 kali pada masing – masing serangan didapatkan akurasi dan presisi pada algoritma *fuzzy*.

Tabel 3.4 Hasil Percobaan serangan DoS

DoS			
No	Interval (menit)	Rata - Rata Durasi Blokir (detik)	Klasifikasi
1	0	600	FP
2	0.1	129,4	FP
3	4	264	FP
4	6	411,4	TP
5	8	484,4	TP
6	3	185	FP
7	9	545,6	TP
8	2	122,2	FP
9	1	66,6	FP
10	10	593	TP
11	5	304,4	TP
12	7	423	TP

Tabel 3.5 Hasil Percobaan serangan *host discovery*

Port Scanning			
No	Interval (menit)	Rata - Rata Durasi Blokir (detik)	Klasifikasi
1	0	600	FP
2	0.1	80,4	TP
3	4	249	TP
4	6	333,2	FP
5	8	291	TP
6	3	211,4	TP
7	9	179,6	TP
8	2	183,2	TP
9	1	231	TP
10	10	167,6	FP
11	5	94,4	TP
12	7	67,8	TP

Hasil akurasi dan presisi yang didapatkan dari kedua data tabel diatas adalah sebagai berikut:

1. Akurasi (CR)

Sesuai dengan persamaan nilai akurasi dapat dihitung dengan rumus:

Akurasi DoS = (TP+TN)/(TP+TN+FP+FN)
= (6+0)/(6+0+6+0)
= 6/12 ≅ 50%

Akurasi host discovery = (TP+TN)/(TP+TN+FP+FN)
= (9+0)/(9+0+3+0)
= 9/12 ≅ 75%

4. Kesimpulan dan Saran

4.1. Kesimpulan

Berdasarkan hasil pengujian penerapan IDS pada jaringan SDN dapat melakukan deteksi pada serangan DoS dan *host discovery* dengan membaca hasil log aplikasi snort dan melakukan blokir dengan menggunakan algoritma *fuzzy*. didapatkan hasil *jitter* dari serangan DoS saat menggunakan kemanan sebesar 1,15 x 10-6 detik sedangkan saat sebelum menggunakan kemanan jitter yang didapat sebesar 1,52 detik. Sedangkan untuk serangan *host discovery* didapatkan jitter saat menggunakan kemana sebesar 0,57 x 10-6 detik sedangkan saat sebelum menggunakan kemanan *jitter* yang didapatkan sebesar 0,55 x 10-5 detik. dari *delay* yang didapatkan dari seragan DoS saat sebelum menggunakan kemana sebesar 5,39 detik dan saat menggunakan kemanan sebesar 3,29 x 10-2 detik, sedangkan pada seranagan *host discovery* didapatkan *delay* sebesar 1,64 x 10-2 detik sebelum menggunakan kemanan dan 1,65 x 10-2 detik sesudah menggunakan kemanan. *throughput* dari serangan DoS sebesar 2001,709 Kbps sebelum digunakan kemanan karena banyaknya paket serangan yang terdeteksi dan saat sesudah menggunakan kemanan sebesar 2,67 x 10-4 Kbps karena besaran paket IoT yang kecil, sedangkan saat serangan *host inscpector* sebelum menggunakan kemanan sebesar 4,51 x 10-1 Kbps dan saat menggunakan kemanan sebesar 6,25 x 10-4 Kbps. Nilai yang didapatkan dari pengujian *packet loss* pada serangan DoS didapatkan packet loss sebesar 99.99% sebelum menggunakan kemanan sedangkan saat menggunakan keamanan didapatkan *packet loss* sebesar 0%. Durasi blokir serangan yang didapatkan saat menggunakan algoritma *fuzzy* dengan menggunakan pembuktian Sogeno didapatkan akurasi dari algoritma *fuzzy* sebesar 50% untuk serangan DoS dan 75% untuk serangan *host discovery*.

4.2 Saran

Melakukan Implementasi dengan jumlah host yang lebih banyak dan jumlah *attacker* yang lebih dari satu untuk mengetahui kemampuan dari algoritma *fuzzy* yang digunakan. Melakukan percobaan pada SDN dengan *controller* yang berbeda seperti pada ONOS atau OpenDayLight. Melakukan perubahan topologi dengan memisahkan antara snort atau IDS dengan *controller*, untuk mendapatkan flexibilitas dari jaringan yang lebih baik.

Daftar Pustaka:

[1] S. G. T. Mandar B. Shinde, M. B. Shinde, and S. G. Tamhankar, “Review : Software Defined Networking and OpenFlow,” Int. J. Sci. Res. Netw. Secur. Commun., vol. 1, no. 2, pp. 18–20, 2013.

[2] J. Ali, Dorri. Salil S, Kanhere. Raja, “Blockchain in Internet of Things: Challenges and Solutions,” Ann. Pure Appl. Log., vol. 45, no. 2 PART 1, pp. 129–137, 2016.

[3] N. Verma, S. Sangwan, S. Sangwan, and D. Parsad, “IoT security challenges and counters measures,” Int. J. Recent Technol. Eng., vol. 8, no. 3, pp. 1519–1528, 2019.

[4] K. Tabassum, A. Ibrahim, and S. A. El Rahman, “Security issues and challenges in IoT,” 2019 Int. Conf. Comput. Inf. Sci. ICCIS 2019, 2019.

[5] P. Mishra, A. Biswal, S. Garg, R. Lu, M. Tiwary, and D. Puthal, “Software defined internet of things security: Properties, state of the art, and future research,” IEEE Wirel. Commun., vol. 27, no. 3, pp. 10–16, 2020.

[6] M. Iqbal, F. Iqbal, F. Mohsin, M. Rizwan, and F. Ahmad, "Security issues in software defined networking (SDN): Risks, challenges and potential solutions," Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 10, pp. 298–303, 2019.
[7]R. Vilalta et al., "Improving Security in Internet of Things with Software Defined Networking," 2016.
[8] P. Xiao, Z. Li, H. Qi, W. Qu, and H. Yu, "An efficient DDoS Detection with Bloom Filter in SDN," Proc. - 15th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 10th IEEE Int. Conf. Big Data Sci. Eng. 14th IEEE Int. Symp. Parallel Distrib. Proce, pp. 1–6, 2016.
[9] T. Xing, D. Huang, L. Xu, C. J. Chung, and P. Khatkar, "SnortFlow: A OpenFlow-Based Intrusion Prevention System in Cloud Environment," Proc. - 2013 2nd GENI Res. Educ. Exp. Work. GREE 2013, no. March, pp. 89–92, 2013.
[10] G. Garg and R. Garg, "Review On Architecture & Security Issues of SDN," Int. J. Innov. Res. Comput. Commun. Eng., vol. 2, no. 11, pp. 6519–6524, 2014.
[11] W. El-Hajj, F. Aloul, Z. Trabelsi, and N. Zaki, "On Detecting Port Scanning using Fuzzy Based Intrusion Detection System," IWCMC 2008 - Int. Wirel. Commun. Mob. Comput. Conf., pp. 105–110, 2008.
[12] R. F. Pratama, N. A. Suwastika, and M. A. Nugroho, "Design and implementation adaptive Intrusion Prevention System (IPS) for attack prevention in software-defined network (SDN) architecture," 2018 6th Int. Conf. Inf. Commun. Technol. ICoICT 2018, vol. 0, no. c, pp. 299–304, 2018.
[13] A. M. Zarca et al., "Security Management Architecture for NFV/SDN-aware IoT Systems," IEEE Internet Things J., vol. PP, no. c, pp. 1–1, 2019.
[14] ONF, "SDN Security Considerations in the Data Center," ONF Solut. Br., pp. 1–12, 2013.
[15] X. You, Y. Feng, and K. Sakurai, "Packet in Message Based DDoS Attack Detection in SDN Network Using OpenFlow," Proc. - 2017 5th Int. Symp. Comput. Networking, CANDAR 2017, vol. 2018-Janua, pp. 522–528, 2018.
[16] S. Asadollahi, B. Goswami, and M. B. for I. M. pdfamme. Sameer, "Ryu controller's scalability experiment on software defined networks," 2018 IEEE Int. Conf. Curr. Trends Adv. Comput. ICCTAC 2018, no. February, pp. 1–5, 2018.
[17]S. K. Tayyaba, M. A. Shah, O. A. Khan, and A. W. Ahmed, "Software defined network (SDN) based internet of things (IoT): A road ahead," ACM Int. Conf. Proceeding Ser., vol. Part F130522, no. December 2018, 2017.
[18] Y. Yan and H. Wang, "Open vSwitch Vxlan performance acceleration in cloud computing data center," Proc. 2016 5th Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2016, pp. 567–571, 2017.
[19] S.Vijayarani and M. S. S, "INTRUSION DETECTION SYSTEM – A STUDY," Int. J. Secur. Priv. Trust Manag., vol. 4, no. 11, pp. 250–252, 2015.
[20] A. Alhomoud, R. Munir, J. P. Disso, I. Awan, and A. Al-Dhelaan, "Performance evaluation study of Intrusion Detection Systems," Procedia Comput. Sci., vol. 5, pp. 173–180, 2011.
[21] K. . K. R and A. Indra, "Intrusion Detection Tools and Techniques –A Survey," Int. J. Comput. Theory Eng., vol. 2, no. 6, pp. 901–906, 2010.
[22] T. Xing, D. Huang, L. Xu, C. J. Chung, and P. Khatkar, "SnortFlow: A OpenFlow-based intrusion prevention system in cloud environment," Proc. - 2013 2nd GENI Res. Educ. Exp. Work. GREE 2013, pp. 89–92, 2013.
[23] A. Sharifi, F. F. Zad, F. Farokhmanesh, A. Noorollahi, and J. Sharif, "An Overview of Intrusion Detection and Prevention Systems (IDPS) and Security Issues," IOSR J. Comput. Eng., vol. 16, no. 1, pp. 47–52, 2014.
[24] M. Hellmann, "Fuzzy Logic Introduction," 263 Ave. Gen. Leclerc, CS 74205, 35042 Rennes Cedex, Fr. Lab. Antennes Radar Telecom, F.R.E CNRS 2272, Equipe Radar Polarim., no. 1, 2001.

[25] S. Amnalou and K. A. A. Bakar, “Lightweight security mechanism over MQTT protocol for IoT devices,” Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 7, pp. 202–207, 2020.