REPOSITOR, Vol. 4, No. 1, Februari 2022, Pp. 71-82
ISSN : 2714-7975
E-ISSN : 2716-1382

Analisis Dampak Serangan DoS Pada Software Defined Network

Haris Zulfikar Islam[*1], Mahar Faiqurahman[2], Fauzi Dwi Setiawan Sumadi[3]
[1,2,3]Universitas Muhammadiyah Malang
haris.frozen81@gmail.com[*1], mahar@umm.ac.id[2], fauzisumadi@umm.ac.id[3]

Abstrak

Perkembangan kemajuan teknologi informasi saat ini yaitu Software Defined Networking (SDN) semakin menjanjikan. SDN bisa didefinisikan sebagai arsitektur jaringan yang di mana data plane dan control plane dipisahkan dalam suatu jaringan. Pemisahan antara data plane dan control plane sangat menguntungkan dalam pengelolaan jaringan, karena data plane terletak pada perangkat switch sedangkan control plane pada perangkat yang disebut controller dan sehingga proses penggelolaan suatu manajemen jaringan menjadi lebih efektif. Controller SDN sebagai otak dari software defined network (SDN) sering disebut control plane, karena fungsinya mengatur lalu lintas jaringan. Potensi celah keamanan dapat muncul pada jaringan SDN terutama pada controller. Karena controller yang melakukan pengaturan lalu lintas jaringan di dalam SDN, maka ketika terjadi serangan pada controller akan memberikan dampak pada jaringan SDN keseluruhan. Salah satu serangan pada jaringan SDN yang dapat terjadi adalah serangan DoS. Serangan DoS adalah usaha untuk mengakibatkan perangkat jaringan layaknya controllers, computers, routers, switches, server atau network jaringan tidak tersedia bagi user yang dituju. Serangan di controller SDN mempunyai tujuan tujuan agar controller mengalami overload sehingga serangan dos ini mampu mengakibatkan asitektur SDN mengalami kemacetan atau termasuk menghambat controller menempatkan flow rule untuk paket yang bakal di kirim ke obyek. Tujuan yang ingin di capai dari penelitian ini adalah menganalisa dan membandingkan performa dari controller Ryu, ONOS, OpenDaylight, Floodlight dan OvS ketika menerima serangan DoS. Parameter yang akan diuji pada penelitian ini ada cpu usage, flow rule, packet_in, packet_out. Selanjutnya adalah pengukuran parameter Quality of Service seperti jitter, packet loss, dan throughput saat terjadi serangan Denial of Service. Serangan menggunakan hping3 dengan flag syn, udp dan icmp. Dengan paket masing-masing serangan 1.500.000, 1.000.000, dan 500.000.

Kata Kunci: *DoS, Ryu, Opendaylight, Floodlight, Onos. QOS, SDN.*

Abstract

The development of information technology advances today, namely Software Defined Networking (SDN) is increasingly promising. SDN can be defined as a network architecture in which the data plane and control plane are separated in a network. The separation between the data plane and the control plane is very beneficial in network management, because the data plane is located on the switch device while the control plane on the device is called the controller and so the process of managing a network management becomes more effective. The SDN controller as the brain of a software defined network (SDN) is often called a control plane, because of its function of controlling network traffic. Potential security holes can appear on SDN networks, especially on controllers. Because the controller manages network traffic on the SDN, when an attack occurs on the controller it will have an impact on the entire SDN network. One of the attacks on SDN networks that can occur is a DoS attack. A DoS attack is an attempt to make network devices such as controllers, computers, routers, switches, servers or network not available to the intended user. Attacks on SDN controllers have the goal of causing the controller to overload so that this dos attack can cause the SDN architecture to crash or include preventing the controller from placing flow rules for packets to be sent to objects. The goal of this research is to analyze and compare the performance of the Ryu, ONOS, OpenDaylight, Floodlight and OvS controllers when receiving a DoS attack. The parameters to be tested in this study are CPU usage, flow rule, packet_in, packet_out. Next is the measurement of Quality of Service parameters such as jitter, packet loss, and throughput when a Denial of Service attack occurs. Attacks using hping3 with

syn, udp and icmp flags. With the respective attack packages of 1,500,000, 1,000,000, and 500,000.

Keywords: *DoS, Ryu, Openday light, Floodlight, Onos. QOS, SDN*

1. Pendahuluan

Perkembangan kemajuan teknologi informasi saat ini yaitu Software Defined Networking (SDN) semakin menjanjikan [1]. SDN bisa didefinisikan sebagai arsitektur jaringan yang di mana data plane dan control plane dipisahkan dalam suatu jaringan [2]. Pemisahan antara data plane dan control plane sangat menguntungkan dalam pengelolaan jaringan, karena data plane terletak pada perangkat switch sedangkan control plane pada perangkat yang disebut controller dan sehingga proses penggelolaan suatu manajemen jaringan menjadi lebih efektif [3].

Didalam jaringan SDN dibutuhkan protokol supaya control plane bisa berkomunikasi bersama dengan data plane. Protokol yang digunakan adalah OpenFlow. OpenFlow adalah sebuah protokol yang digunakan untuk mengindentifikasi pengaturan traffic dan pengiriman paket melewati switch yang tersedia di dalam jaringan [3]. Potensi celah keamanan dapat muncul pada jaringan SDN terutama pada controller. Karena controller yang melakukan pengaturan lalu lintas jaringan di dalam SDN, maka ketika terjadi serangan pada controller akan memberikan dampak pada jaringan SDN keseluruhan. [4]

Salah satu serangan pada jaringan SDN yang dapat terjadi adalah serangan DoS. Serangan DoS adalah usaha untuk mengakibatkan perangkat jaringan layaknya controllers, computers, routers, switches, server atau network jaringan tidak tersedia bagi user yang dituju. Serangan di controller SDN mempunyai tujuan tujuan agar controller mengalami overload sehingga serangan dos ini mampu mengakibatkan asitektur SDN mengalami kemacetan atau termasuk menghambat controller menempatkan flow rule untuk paket yang bakal di kirim ke obyek [6].

Berdasarkan pada literatur penelitian sebelumnya dan permasalahan yang telah disampaikan, penulis melakukan penelitian dengan judul "DoS Attack Impact Assessment on Software Defined Networks". Penelitian yang akan dilakukan dilakukan analisa *Quality of Service* yang meliputi parameter *Jitter* dan *Troughput* ketika terjadi serangan *Denial of Service* pada *controller OpenDaylight* [3]. Untuk melakukan serangan, pada penelitian tersebut digunakan tool *hping3* untuk membanjiri serangan. Pada penelitian kedua yang telah dilakukan oleh oleh T. Alharbi, S. Layeghy, dan Marius Portman yang berjudul "Experimental Evaluation of the Impact of DoS Attacks in SDN", dilakukan analisa perbandingan dampak serangan untuk tiga platform *controller* yaitu *Ryu*, *ONOS*, dan *Floodlight* saat serangan *Denial of Service* [4]. Serangan menggunakan *tool scapy* dan *pcap* untuk menyerang *controller* tersebut. Parameter yang akan di pakai adalah *PacketDelivery* dan *Cpu load. Pada penelitian yang ketiga telah dilakukan oleh H. Polat and O. Polat, yang berjudul "The effects of DoS attacks on ODL and POX SDN controllers", dilakukan analisa perbandingan dampak serangan untuk dua controller yaitu Opendaylight dan POX saat serangan Denial of Service [6]. Serangan menggunakan hping3 untuk menyerang controller tersebut. Parameter yang akan di pakai adalah host bandwidth.*

Dari penelitian dan literatur sebelumnya yang telah disampaikan diatas penulis melakukan penelitian dengan judul "Analisa dampak serangan DOS pada SDN". Penelitian yang akan dilakukan yaitu dengan membandingkan controller Ryu, OpenDaylight, Floodlight dan ONOS. Parameter yang akan diuji pada penelitian ini ada cpu usage, flow rule, packet_in, packet_out. Selanjutnya adalah pengukuran parameter Quality of Service seperti jitter, packet loss, dan throughput saat terjadi serangan Denial of Service. Serangan menggunakan hping3 dengan flag syn, udp dan icmp. Dengan paket masing-masing serangan 1.500.000, 1.000.000, dan 500.000.

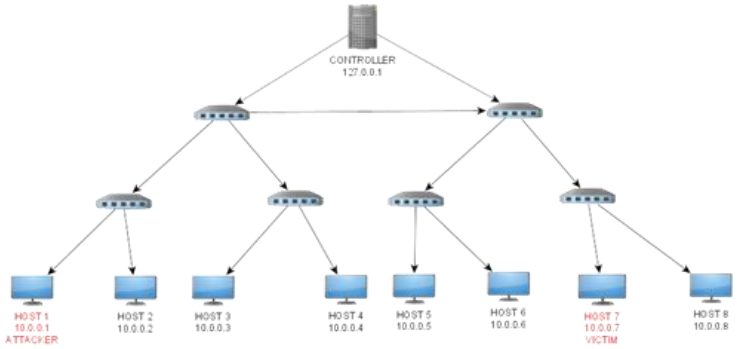
2. Metode Penelitian

2.1 Perancangan Sistem

Pada tahap ini dilakukan perancangan untuk serangan DoS pada SDN. Beberapa komponen yang diperlukan yaitu controller, emulator mininet, script pyhton, tool iperf, dan tool hping3. Parameter yang akan digunakan adalah PACKET_IN, PACKET_OUT menggunakan wireshark pada OvS, jitter, troughput, dan packet loss menggunakan iperf. Pengunaan CPU ussage pada controller dan OvS lalu topologi jaringan yang digunakan akan dijalankan pada emulator mininet dan menggunakan controller ryu, onos, opendaylight dan floodlight.

2.1.1 Topologi Jaringan

Pada tahap ini dilakukan simulasi untuk serangan DoS pada jaringan SDN. Simulasi serangan DoS dilakukan menggunakan tool hping3 dengan cara mengirimkan paket tcp syn, udp dan icmp dalam jumlah 500.000, 1000.000 dan 1500.000 ke host tujuan di jaringan SDN ini. Di dalam analisis sistem ini akan cara membuat jaringan Software Defined Network (SDN) menggunakan emulator mininet dengan menggunakan topologi tree custom dengan depth 6 dan fanout 8. Topologi yang digunakan seperti Gambar 1.



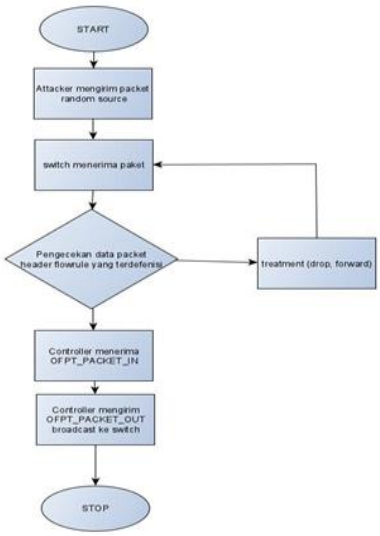
Gambar 1. Rancangan Topologi Sistem

Infrastruktur jaringan SDN yang di implementasikan pada mininet menggunakan topologi three custom yang di mana ada 1 controller, 6 OvS dan 8 host, komponen tersebut memiliki fungsi masing-masing sebagai berikut:

1. Controller bertugas mengontrol jaringan dan meneruskan paket ke switch. Controller akan dijalankan oleh RYU, ONOS, OpenDayLight, dan Floodlight yang bertindak sebagai wewenang terhadap controller.
2. *OpenvirtualSwitch (OvS)* berfungsi sebagai jembatan antara *controller* dan *host* ketika berkomunikasi. Fungsi dari *OvS* tersebut adalah meneruskan packet yang dikirim dari host ke tujuannya.
3. Host berfungsi untuk menguji performansi jaringan dengan melakukan pengiriman data.
4. OpenFlow berfungsi sebagai jembatan komunikasi antara *controller* dan *OvS*.

2.1.2 Pembangunan Sistem

Setelah merancang topologi seperti Gambar 2, maka selanjutnya pembangunan sistem agar mengetahui bagaimana sistem bekerja sesuai penelitian, rancangan sistem yang di gunakan seperti Gambar 2.



Gambar 2. Mekanisme Penyerangan

Dalam pembangunan sistem ini dijelaskan serangan yang dilakukan oleh attacker. Attacker menggunakan tools hping3 untuk mengirim serangan ke victim. Tools hping3 digunakan untuk menginjeksi paket serangan. Lalu dengan hping3 menggunakan random source untuk menyerang victim, sehingga tidak dikenali jaringan dan membuat resource di jaringan SDN down. Saat penyerangan berlangsung tools wireshark digunakan untuk menghitung jumlah PACKET_IN dan PACKET_OUT. Python script untuk menghitung jumlah rata-rata flow rule dan cpu usage. Dan yang terakhir tool iperf digunakan untuk mengitung quality of service (QoS) di dalam jaringan SDN. Serangan tersebut dilakukan pada controller ryu, onos, opendaylight dan floodlight. Berdasarkan Gambar 2.2 dilakukan pengiriman packet DoS. Langkah pertama yang dilakukan adalah attacker mengirimkan packet 500.000, 1.000.000, dan 1.500.000 dengan random source ke victim. Lalu OvS akan menerima packet yang masuk. Setelah itu akan dilakukan proses pengecekan data packet header flow rule yang sudah terdefinisikan. Jika packet sesuai flow rule yang terdefinisi packet akan lanjutkan sesuai dengan treatment-nya (drop packet/forward packet). Lalu jika packet tidak sesuai dengan flow rule maka packet tersebut termasuk packet baru karena belum dapat pemetakan IP dan MAC addressnya. Kemudian packet tersebut dienkapsulasi menjadi OFPT_PACKET_IN dan dikirimkan ke controller. Lalu controller mengenkapsulasi menjadi OFPT_PACKET_OUT, lalu OFPT_PACKET_OUT akan langsung di broadcast ke host victim sampai semua packet terkirim. Destination dibuat random source supaya packet langsung terkirim ke OvS tanpa install flow rule.

2.1.3 Kebutuhan Software

1. Ubuntu Desktop
   Ubuntu merupakan salah satu produk yang didistribusikan oleh Linux dengan berbasis Debian dan didistribusikan secara gratis. Dan giunakan untuk menjalankan mininet dan controller.
2. VMware
   VMware merupakan perangkat lunak virtualisasi yang dapat digunakan untuk mengeksekusi sistem operasi tambahan di dalam sistem operasi utama.
3. Controller RYU
   Ryu adalah satu dari sekian banyak controller yang digunakan untuk memenejemen dan mengontrol jaringan SDN. Dalam pembuatan aplikasinya Ryu controller menggunakan bahasa pemrograman python.
4. Controller ONOS
   Onos adalah satu dari sekian banyak controller yang digunakan untuk memenejemen dan mengontrol jaringan SDN. Dalam pembuatan aplikasinya Onos controller menggunakan bahasa pemrograman java.
5. Controller Opendaylight
   OpenDaylight adalah satu dari sekian banyak controller yang digunakan untuk memenejemen dan mengontrol jaringan SDN. Dalam pembuatan aplikasinya OpenDaylight controller menggunakan bahasa pemrograman java.
6. Controller Floodlight
   Floodlight adalah satu dari sekian banyak controller yang digunakan untuk memenejemen dan mengontrol jaringan SDN. Dalam pembuatan aplikasinya Floodlight controller menggunakan bahasa pemrograman java.
7. Python
   Python adalah bahasa pemrograman yang digunakan untuk membuat aplikasi yang akan dijalankan controller.
8. Java
   Java adalah bahasa pemrograman yang digunakan untuk membuat aplikasi yang akan dijalankan *controller*.
9. Open Virtual Switch (OVS)
   Open Virtual Switch adalah visual multilayer yang diterapkan pada perangkat jaringan agar pengkontrolan sanggup dilaksanakan secara terpusat memakai protokol OpenFlow.
10. Mininet
    Mininet merupakan sebuah emulator yang digunakan untuk membuat jaringan virtual dalam penelitian.
11. Tools Hping3
    Sebuah tools yang digunakan untuk menggunakan lalu lintas yang sebelumnya ditangkap untuk menguji serangan DoS oleh attacker.

12. Iperf
    Sebuah tools untuk mengukur QoS pada sebuah jaringan.

2.1.4 Kebutuhan Hardware

Kebutuhan hardware menggunakan processor Intel® Core™ i7-8750H Processor 2.2 GHz Memory (RAM) 8.00 GB OS Ubuntu 16.04 LTS Harddisk 1000 GB.

3. Hasil Penelitian dan Pembahasan

Pada tahap ini peniliti akan menjelaskan hasil dari pengujian dan analisa sistem yang telah dirancang. Pengujian dilakukan dengan maksud untuk menilai kinerja dari perancangan sistem apakah telah memenuhi kriteria yang diinginkan dengan melakukan analisis dampak serangan DoS pada SDN.

Pada pengujian ini akan dilakukan serangan DoS pada SDN. Beberapa komponen yang diperlukan yaitu controller, emulator mininet, script pyhton, tool iperf, dan tool hping3. Parameter yang akan digunakan adalah PACKET_IN, PACKET_OUT menggunakan wireshark pada OvS, jitter, troughput, dan packet loss menggunakan iperf. Pengunaan CPU ussage pada controller dan OvS lalu topologi jaringan yang digunakan akan dijalankan pada emulator mininet dan menggunakan controller ryu, onos, opendaylight dan floodlight.

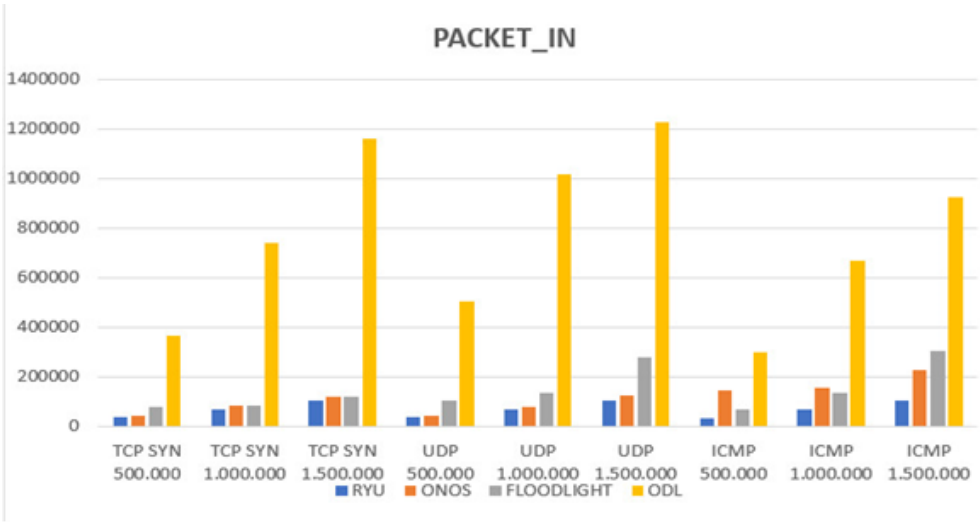
3.1.1 Pengujian jumlah PACKET_IN dan PACKET_OUT terhadap serangan DoS pada Controller Ryu, Controller Opendaylight, Controller Floodlight, dan Controller Onos

Pengujian packet_in dan packet_out terhadap Controller masing- masing diuji dengan pengiriman packet syn tcp, udp dan icmp sebanyak 500.000, 1000.000, dan 1500.000. Tabel 1 merupakan hasil dari pengujian adalah sebagai berikut

Tabel 1. Pengujian Packet_In dan Packet_Out Pada Setiap Controller

PAKET	JUMLAH PAKET	PACKET_IN				PACKET_OUT			
		CONTROLLER				CONTROLLER			
		RYU	ONOS	FLDLGT	ODL	RYU	ONOS	FLDLGT	ODL
TCP SYN	500.000	36.108	44.540	77.756	364.086	36.108	40.038	38.247	1250
	1.000.000	70.260	81.742	83.841	740.718	70.260	90.388	65.668	2157
	1.500.000	104.365	121.723	118.681	1.160.827	104.314	134.706	116.837	3354
UDP	500.000	35.503	41.567	105.945	505.626	35.503	46.467	65.804	1619
	1.000.000	69.547	80.949	136.358	1.014.518	69.547	90.268	132.686	3689
	1.500.000	105.419	122.965	279.592	1.224.289	105.399	135.958	213.854	3465
ICMP	500.000	35.230	145.937	67.193	299.676	35.230	150.359	26.558	1619
	1.000.000	69.853	153.944	136.312	665.936	69.853	162.703	91.082	2539
	1.500.000	105.781	227.044	306.152	922.287	105.772	240.045	256.273	3610

Setelah dilakukan pengujian didapatkan hasil seperti yang ditunjukan pada Tabel 1, selanjutnya dari hasil pengujian tersebut akan disajikan dalam bentuk grafik perbandingan pada Gambar 3. dan Gambar 4.

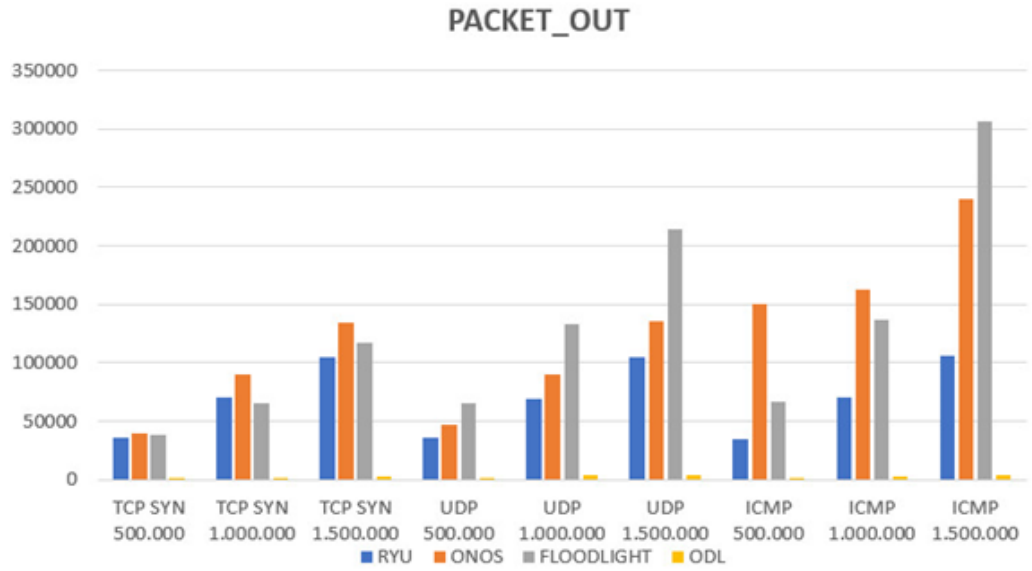


Gambar 3. Grafik Perbandingan Packet_In Saat Terkena Serangan Tcp Syn, Udp, dan Icmp

Dari grafik perbandingan packet in di atas terkena serangan paket tcp syn dengan dampak paling tinggi adalah controller opendaylight dengan hasil 500.000 paket dengan jumlah 364.086, 1.000.000 paket dengan jumlah 740.718 dan 1.500.000 paket dengan jumlah 1.160.827. Sedangkan dampak paling rendah controller ryu dengan hasil 500.000 paket dengan jumlah 36.108, 1.000.000 paket dengan jumlah 70.260 dan 1.500.000 paket dengan jumlah 104.365.

Serangan kedua dengan paket udp dampak paling tinggi adalah controller opendaylight dengan hasil 500.000 paket dengan jumlah 505.626, 1.000.000 paket dengan jumlah 1.014.518 dan 1.500.000 paket dengan jumlah 1.224.289. Sedangkan dampak paling rendah controller ryu dengan hasil 500.000 paket dengan jumlah 35.503, 1.000.000 paket dengan jumlah 69.547 dan 1.500.000 paket dengan jumlah 105.419.

Serangan ketiga dengan paket icmp dampak paling tinggi adalah controller opendaylight dengan hasil 500.000 paket dengan jumlah 299.676, 1.000.000 paket dengan jumlah 665.936 dan 1.500.000 paket dengan jumlah 922.287. Sedangkan dampak paling rendah controller ryu dengan hasil 500.000 paket dengan jumlah 35.230, 1.000.000 paket dengan jumlah 69.853 dan 1.500.000 paket dengan jumlah 105.781.



Gambar 4. Grafik Perbandingan Packet_Out Saat Terkena Serangan Tcp Syn, Udp, dan Icmp

Dari grafik perbandingan packet out di atas terkena serangan paket tcp syn dengan dampak paling tinggi adalah controller onos dengan hasil 500.000 paket dengan jumlah 40.038, 1.000.000 paket dengan jumlah 90.388 dan 1.500.000 paket dengan jumlah 134.706. Sedangkan dampak paling rendah controller opendaylight dengan hasil 500.000 paket dengan jumlah 1250, 1.000.000 paket dengan jumlah 2157 dan 1.500.000 paket dengan jumlah 3354.

Serangan kedua menggunakan paket udp dengan dampak paling tinggi adalah controller floodlight dengan hasil 500.000 paket dengan jumlah 65.804, 1.000.000 paket dengan jumlah 132.686 dan 1.500.000 paket dengan jumlah 213.854. Sedangkan dampak paling rendah controller opendaylight dengan hasil 500.000 paket dengan jumlah 1619, 1.000.000 paket dengan jumlah 3689 dan 1.500.000 paket dengan jumlah 3465.

Serangan kedua menggunakan paket icmp dengan dampak paling tinggi adalah controller onos dengan hasil 500.000 paket dengan jumlah 150.359, 1.000.000 paket dengan jumlah 162.703 dan 1.500.000 paket dengan jumlah 240.045. Sedangkan dampak paling rendah controller opendaylight dengan hasil 500.000 paket dengan jumlah 1619, 1.000.000 paket dengan jumlah 2539 dan 1.500.000 paket dengan jumlah 3610.

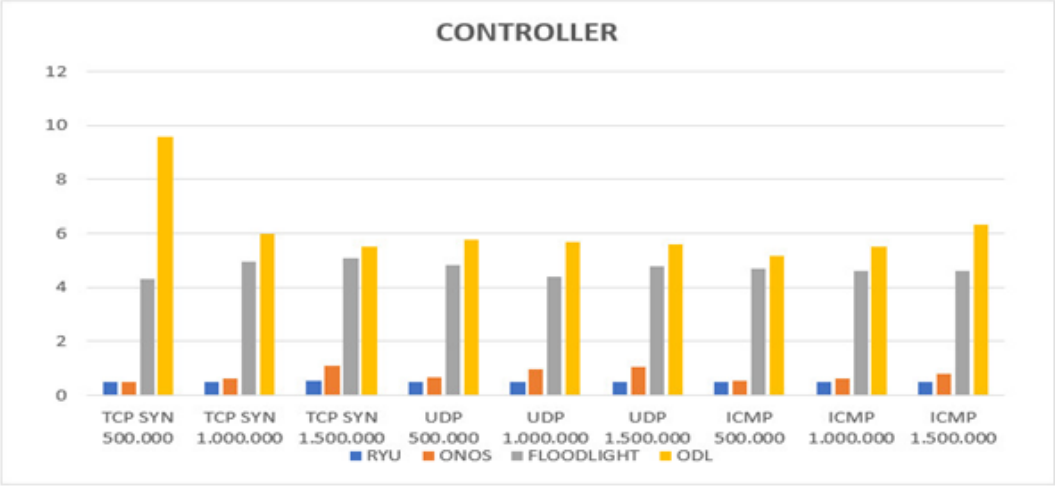
3.1.2 Pengujian cpu ussage terhadap serangan DoS pada Controller Ryu, Controller Opendaylight, Controller Floodlight, dan Controller Onos

Pengujian cpu ussage terhadap Controller masing- masing diuji dengan pengiriman packet syn tcp, udp dan icmp sebanyak 500.000, 1000.000, dan 1500.000. Tabel 2 merupakan hasil dari pengujian adalah sebagai berikut

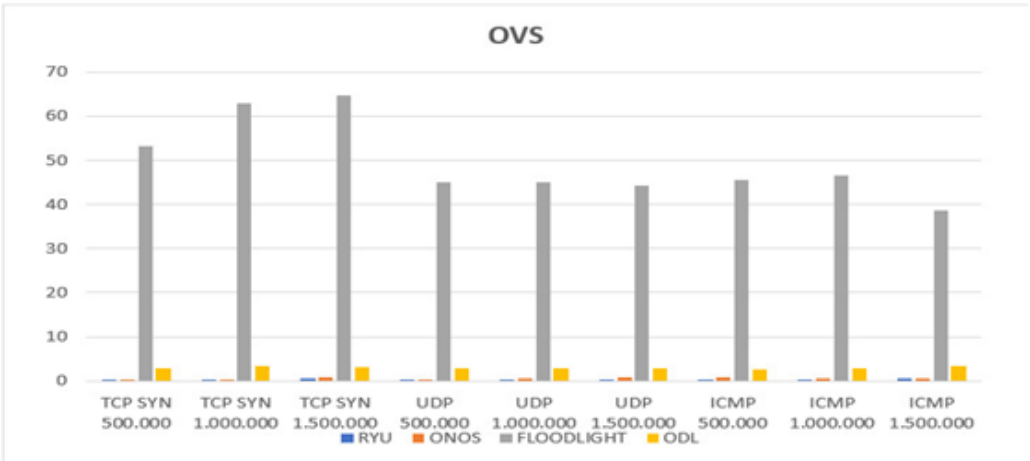
Tabel 2. Pengujian Cpu_Ussage Pada Setiap Controller

PAKET	JUMLAH PAKET	CPU USSAGE (%)							
		CONTROLLER				OVS			
		RYU	ONOS	FLDLGT	ODL	RYU	ONOS	FLDLGT	ODL
TCP SYN	500.000	0,49	0,50	4,33	9,56	0,30	0,33	53,19	2,93
	1.000.000	0,50	0,64	4,94	5,99	0,25	0,32	62,82	3,30
	1.500.000	0,53	1,08	5,0	5,52	0,46	0,76	64,61	3,08
UDP	500.000	0,48	0,67	4,82	5,76	0,27	0,29	45,09	2,84
	1.000.000	0,51	0,95	6,41	5,68	0,26	0,49	45,08	2,76
	1.500.000	0,50	1,07	4,76	5,60	0,25	0,81	44,32	2,86
ICMP	500.000	0,49	0,52	4,71	5,15	0,25	0,75	45,65	2,69
	1.000.000	0,48	0,61	4,59	5,50	0,25	0,45	46,50	2,89
	1.500.000	0,48	0,78	4,59	6,31	0,46	0,55	38,64	3,49

Setelah dilakukan pengujian didapatkan hasil seperti yang ditunjukan pada Tabel 2, selanjutnya dari hasil pengujian tersebut akan disajikan dalam bentuk grafik perbandingan pada Gambar 5 dan Gambar 6.



Gambar 5. Grafik Perbandingan Cpu_Ussage Saat Terkena Serangan Tcp Syn, Udp, dan Icmp



Gambar 6. Grafik Perbandingan Cpu_Ussage Saat Terkena Serangan Tcp Syn, Udp, dan Icmp

Dari grafik perbandingan *cpu ussage* di atas terkena serangan paket *tcp syn* dengan dampak paling tinggi adalah controller *opendaylight* dengan hasil 500.000 paket dengan persentase 9,56%, 1.000.000 paket dengan persentase 5,99% dan 1.500.000 paket dengan persentase 5,52%. Sedangkan dampak paling rendah *controller ryu* dengan hasil 500.000 paket dengan persentase 0,49%, 1.000.000 paket dengan persentase 0,50% dan 1.500.000 paket dengan persentase 0,53%.

Serangan kedua menggunakan paket udp dengan dampak paling tinggi adalah controller opendaylight dengan hasil 500.000 paket dengan persentase 5,76%, 1.000.000 paket dengan persentase 5,68% dan 1.500.000 paket dengan persentase 5,60%. Sedangkan dampak paling rendah controller ryu dengan hasil 500.000 paket dengan persentase 0,48%, 1.000.000 paket dengan persentase 0,51% dan 1.500.000 paket dengan persentase 0,50%.

Serangan ketiga menggunakan paket *icmp* dengan dampak paling tinggi adalah controller *opendaylight* dengan hasil 500.000 paket dengan persentase 5,15%, 1.000.000 paket dengan persentase 5,50% dan 1.500.000 paket dengan persentase 6,31%. Sedangkan dampak paling rendah *controller ryu* dengan hasil 500.000 paket dengan persentase 0,49%, 1.000.000 paket dengan persentase 0,48% dan 1.500.000 paket dengan persentase 0,48%.

Dari grafik perbandingan cpu ussage di atas terkena serangan paket tcp syn dengan dampak paling tinggi adalah OvS floodlight dengan hasil 500.000 paket dengan persentase 53,19%, 1.000.000 paket dengan persentase 62,82% dan 1.500.000 paket dengan persentase 64,61%. Sedangkan dampak paling rendah OvS ryu dengan hasil 500.000 paket dengan persentase 0,30%, 1.000.000 paket dengan persentase 0,25% dan 1.500.000 paket dengan persentase 0,46%.

Serangan kedua menggunakan paket udp dengan dampak paling tinggi adalah OvS floodlight dengan hasil 500.000 paket dengan persentase 45,09%, 1.000.000 paket dengan persentase 45,08% dan 1.500.000 paket dengan persentase 44,32%. Sedangkan dampak paling rendah OvS ryu dengan hasil 500.000 paket dengan persentase 0,27%, 1.000.000 paket dengan persentase 0,26% dan 1.500.000 paket dengan persentase 0,25%.

Serangan ketiga menggunakan paket icmp dengan dampak paling tinggi adalah OvS floodlight dengan hasil 500.000 paket dengan persentase 44,65%, 1.000.000 paket dengan persentase 46,50% dan 1.500.000 paket dengan persentase 38,64%. Sedangkan dampak paling rendah OvS ryu dengan hasil 500.000 paket dengan persentase 0,25%, 1.000.000 paket dengan persentase 0,25% dan 1.500.000 paket dengan persentase 0,46%.

3.1.3 Pengujian flow rule terhadap serangan DoS pada Controller Ryu, Controller Opendaylight, Controller Floodlight, dan Controller Onos

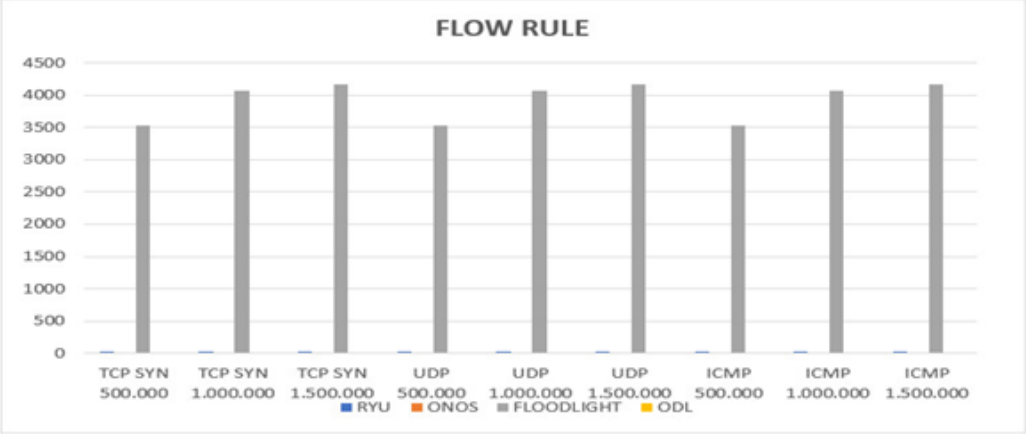
Pengujian rata-rata flow rule masing- masing diuji dengan pengiriman packet syn tcp, udp dan icmp sebanyak 500.000, 1000.000, dan 1500.000. Hasil dari pengujian seperti pada Tabel 3 berikut.

Tabel 3. Pengujian Rata-Rata Flow Rule Pada Semua Controller

PAKET	JUMLAH PAKET	RATA-RATA JUMLAH FLOW RULE			
		CONTROLLER			
		RYU	ONOS	FLDLGT	ODL
TCP SYN	500.000	27	6	3533	5
	1.000.000	27	6	4063	5
	1.500.000	27	6	4163	5
UDP	500.000	27	6	3533	5
	1.000.000	27	6	4063	5
	1.500.000	27	6	4163	5
ICMP	500.000	27	6	3533	5
	1.000.000	27	6	4063	5
	1.500.000	27	6	4163	5

Setelah dilakukan pengujian didapatkan hasil seperti yang ditunjukan pada Tabel 3, selanjutnya dari hasil pengujian tersebut akan disajikan dalam bentuk grafik perbandingan pada Gambar 7.

Dari grafik perbandingan pada Gambar 7 memiliki jumlah flow rule yang sama. Controller ryu dengan dampak serangan tcp syn, udp, dan icmp dengan paket 500.000, 1.000.000 dan 1.500.000 menghasilkan rata-rata flow rule 27. Controller onos dengan dampak serangan tcp syn, udp, dan icmp dengan paket 500.000, 1.000.000 dan 1.500.000 menghasilkan rata-rata flow rule 6. Controller floodlight dengan dampak serangan tcp syn, udp, dan icmp dengan paket 500.000 menghasilkan rata-rata flow rule 3533, serangan tcp syn, udp, dan icmp dengan paket 1.000.000 menghasilkan rata-rata flow rule 4063, dan serangan tcp syn, udp, dan icmp dengan paket 1.500.000 menghasilkan rata-rata flow rule 4163. Controller opendaylight dengan dampak serangan tcp syn, udp, dan icmp dengan paket 500.000, 1.000.000 dan 1.500.000 menghasilkan rata-rata flow rule 5.



Gambar 7. Grafik Perbandingan Flow Rule Saat Terkena Serangan Tcp Syn, Udp, dan Icmp

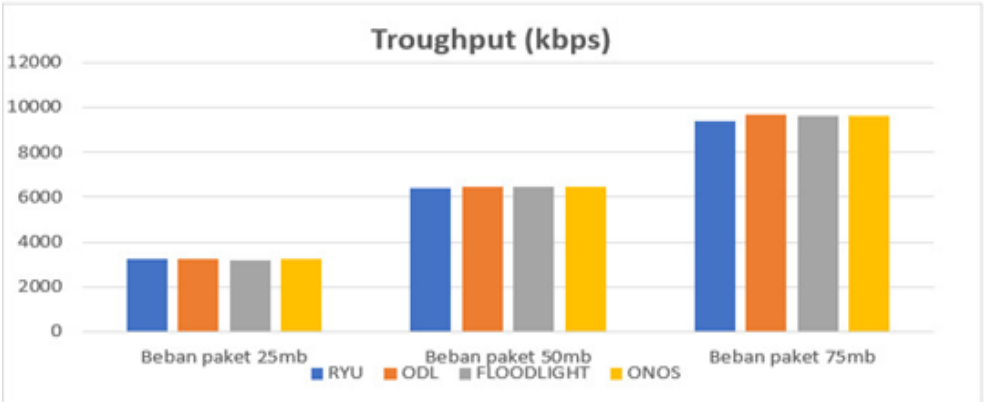
3.1.4 Pengujian dan Analisa Troughput

Pengujian QoS yaitu throughput dilakukan dengan maksud untuk mengetahui kemampuan suatu jaringan dalam hal pengiriman data secara aktual. Throughput merupakan jumlah total kedatangan data yang sukses diamati pada node tujuan selama interval tertentu dan dibagi dengan durasi interval waktu tersebut. Pengujian dilakukan dengan menggunakan aplikasi iperf. Pada jaringan, h1 akan menjadi server sedangkan h7 akan menjadi client yang akan mengirimkan traffic UDP selama 60 detik. Pengujian dijalankan saat bersama serangan, Kemudian pengujian dilakukan dengan memberikan beban traffic 25mb, 50mb dan 75mb.

Tabel 4. Pengujian Troughput Pada Controller Ryu, Opendaylight, Floodlight, Onos

Beban Paket	Troughput (kbps)			
	RYU	ODL	FLOODLIGHT	ONOS
25mb	3215	3216	3193	3216
50mb	6425	6432	6431	6426
75mb	9351	9691	9633	9622

Setelah dilakukan pengujian didapatkan hasil seperti yang ditunjukan pada Tabel 4, selanjutnya dari hasil pengujian tersebut akan disajikan dalam bentuk grafik perbandingan pada Gambar 8.



Gambar 8. Grafik Perbandingan Troughput Controller Ryu, Odl, Floodlight, Onos

Berdasarkan hasil pengujian *throughput* dari ke empat *controller* tersebut menunjukan hasil bahwa semua *controller* memiliki rata-rata troughput yang signifikan sesuai bertambahnya beban variasi paket. Tetapi hasil pengujian troughput di atas memenuhi standarisasi QoS dengan nilai yang sangat baik.

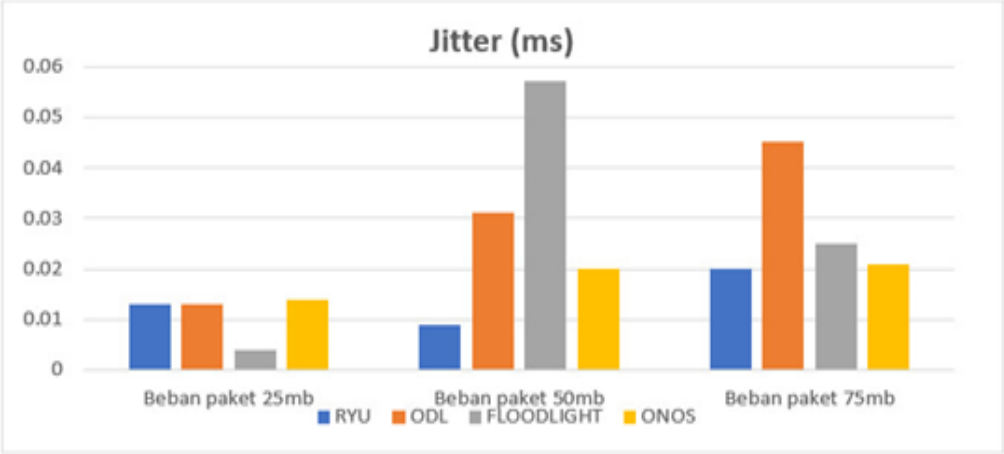
3.1.5 Pengujian dan Analisa Jitter

Pengujian jitter dilakukan untuk mengetahui variasi delay antara pengiriman data pada jaringan yang dapat terjadi karena dipengaruhi beban traffic pada jaringan itu sendiri. Pengujian jitter dilakukan bersamaan dengan packet loss.

Tabel 5. Pengujian Jitter Pada Controller Ryu, Opendaylight, Floodlight dan Onos

Beban Paket	Jitter (ms)			
	RYU	ODL	FLOODLIGHT	ONOS
25 mb	0,013	0,013	0,004	0,014
50 mb	0,009	0,030	0,057	0,020
75 mb	0,020	0,045	0,025	0,020

Setelah dilakukan pengujian didapatkan hasil seperti yang ditunjukan pada Tabel 5, selanjutnya dari hasil pengujian tersebut akan disajikan dalam bentuk grafik perbandingan pada Gambar 9.



Gambar 9. Grafik Perbandingan Jitter Controller Ryu, Opendaylight, Floodlight, dan Onos

Berdasarkan hasil pengujian jitter dari ke empat controller tersebut menunjukan hasil bahwa controller floodlight memiliki jumlah jitter yang lebih tinggi dari semua controller dan semua beban varisai paket yaitu 0,057ms. Tetapi hasil pengujian jitter di atas memenuhi standarisasi QoS dengan nilai yang baik.

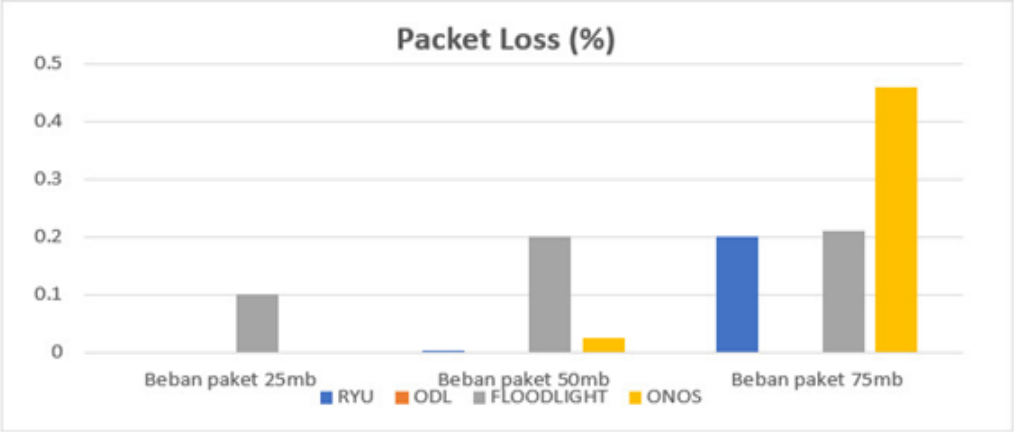
3.1.6 Pengujian dan Analisa Packet Loss

Pengujian packet loss dilakukan untuk mengetahui jumlah data yang dikirim tidak diterima oleh node tujuan dikarena beberapa faktor antara lain terjadinya overload traffic jaringan, ataupun kegagalan pada sisi penerima. Pengujian packet loss dilakukan bersamaan dengan pengujian jitter menggunakan iperf.

Tabel 6. Pengujian Packet Loss Pada Controller Ryu, Opendaylight, Floodlight dan Onos

Beban Paket	Packet Loss (%)			
	RYU	ODL	FLOODLIGHT	ONOS
25mb	0	0	0,1	0
50mb	0,0043	0	0,2	0,025
75mb	0,2	0	0,21	0,46

Setelah dilakukan pengujian didapatkan hasil seperti yang ditunjukan pada Tabel 6, selanjutnya dari hasil pengujian tersebut akan disajikan dalam bentuk grafik perbandingan pada Gambar 10.



Gambar 10. Grafik Perbandingan Packet Loss Controller Ryu, Opendaylight, Floodlight, dan Onos

Berdasarkan hasil pengujian paket loss dari ke empat controller tersebut menunjukan hasil bahwa controller onos memiliki jumlah paket loss yang lebih tinggi yaitu 0,46% dibandingkan dengan semua controller dan beban semua variasi. . Tetapi hasil pengujian packet loss di atas memenuhi standarisasi QoS dengan nilai yang sangat baik.

4. Kesimpulan

Pengujian parameter Packet In dari setiap controller hasilnya bervariasi tergantung bagaimana pemetaan dari controller tersebut mengaturnya. Controller ryu, memiliki packet in yang rendah dari ke empat controller. Sedangkan controller opendaylight memiliki packet in yang paling tinggi dari ke empat controller.

Pengujian parameter cpu ussage di controller ryu menghasilkan yang terendah dibandingkan dengan yang lainnya. Dikarenakan controller ryu memiliki modularitas yang rendah dan memakai bahasa pemrogaman python untuk menjalankan aplikasinya. Sedangkan controller onos, controller floodlight, dan controller opendaylight memiliki modularitas yang tinggi dibandingkan controller ryu, dan memakai bahasa pemrogaman java untuk menjalankan berbagai aplikasinya, sehingga cpu ussage dari ketiga controller tersebut lebih tinggi dari pada controller ryu [17].

Pengujian parameter cpu ussage di OvS ryu, OvS onos, dan Ovs opendaylight menghasilkan persentase yang rendah dibandingkan OvS floodlight memiliki persentase yang tinggi. Dikarenakan pengambilan persentase cpu ussage di OvS berdasarkan jumlah flow rule setiap controller. OvS ryu, OvS onos dan OvS opendaylight memiliki flow rule yang rendah sedangkan OvS floodlight memiliki flow rule yang sangat tinggi.

Pengujian parameter flow rule di controller ryu, controller onos, controller floodlight, dan controller opendaylight menghasilkan hasil yang bebeda beda. Controller ryu, controller onos, dan controller opendaylight memiliki flow rule yang rendah, sedangkan controller floodlight memiliki jumlah flow rule yang tinggi. Hasil flow rule mendapatkan hasil yang berbeda karena controller ryu, controller onos, dan controller opendaylight menggunakan pemetakan mac address untuk menghitung flow rule yang masuk. Sedangkan controller floodlight menggunakan pemetakan ip source untuk menghitung flow rule yang masuk.

Pengujian QoS meliputi jitter, troughput, dan packet loss pada controller ryu, cotroller onos, controller floodlight, dan controller opendaylight memiliki hasil yang tidak signifikan. Dikarenakan host normal masih bisa mengirim paket. Jadi serangan menggunakan random source hanya sedikit berpengaruh pada controller ryu, controller onos, controller opendaylight dan controller floodlight.

Referensi

[1] P. Zhang, H. Wang, C. Hu, and C. Lin, “On denial of service attacks in software defined networks,” *IEEE Netw.*, vol. 30, no. 6, pp. 28–33, 2016, doi: 10.1109/MNET.2016.1600109NM.

[2] H. Farhady, H. Lee, and A. Nakao, “Software-Defined Networking: A survey,” *Comput. Networks*, vol. 81, no. February, pp. 79–95, 2015, doi: 10.1016/j.comnet.2015.02.014.

[3] A. Sangodoyin, T. Sigwele, P. Pillai, Y. F. Hu, I. Awan, and J. Disso, “DoS Attack Impact Assessment on Software Defined Networks,” *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 231, pp. 11–22, 2018, doi: 10.1007/978-3-319-76571-6_2.
[4] T. Alharbi, S. Layeghy, and M. Portmann, “Experimental evaluation of the impact of DoS attacks in SDN,” *2017 27th Int. Telecommun. Networks Appl. Conf. ITNAC 2017*, vol. 2017-January, pp. 1–6, 2017, doi: 10.1109/ATNAC.2017.8215424.
[5] H. Polat and O. Polat, “The effects of DoS attacks on ODL and POX SDN controllers,” *ICIT 2017 - 8th Int. Conf. Inf. Technol. Proc.*, pp. 554–558, 2017, doi: 10.1109/ICITECH.2017.8080058.
[6] Aris Cahyadi Risdianto, Muhammad Arif & Eueung Mulyana, https://eueung.gitbooks.io/buku-komunitas-sdn-rg/content/pengantar_sdn/README.html
[7] C. Zacker, “Networking Concepts,” *CompTIA® Network+® Pract. Tests*, pp. 1–46, 2018, doi: 10.1002/9781119549475.ch1.
[8] I. Z. Bholebawa and U. D. Dalal, “Performance analysis of SDN/openflow controllers: POX versus floodlight,” Wirel. Pers. Commun., vol. 98, no. 2, pp. 1679–1699, 2018, doi: 10.1007/s11277-017-4939-z.
[9] M. Ambrosin, M. Conti, F. De Gaspari, and N. Devarajan, “Amplified distributed denial of service attack in software defined networking,” *2016 8th IFIP Int. Conf. New Technol. Mobil. Secur. NTMS 2016*, no. June 2019, pp. 1–6, 2016, doi: 10.1109/NTMS.2016.7792432.
[10] 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, doi: 10.1109/CANDAR.2017.93.
[11] R. K. Arbettu, R. Khondoker, K. Bayarou, and F. Weber, “Security analysis of OpenDaylight, ONOS, Rosemary and Ryu SDN controllers,” *2016 17th Int. Telecommun. Netw. Strateg. Plan. Symp. Networks 2016 - Conf. Proc.*, pp. 37–44, 2016, doi: 10.1109/NETWKS.2016.7751150.
[12] A. Rajaratnam, R. Kadikar, S. Prince, and M. Valarmathi, “Software defined networks: Comparative analysis of topologies with ONOS,” *Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017*, vol. 2018-Janua, pp. 1377–1381, 2018, doi: 10.1109/WiSPNET.2017.8299989.
[13] Z. K. Khattak, M. Awais, and A. Iqbal, “07097868,” 2014.
[14] S. Asadollahi and B. Goswami, “Experimenting with scalability of floodlight controller in software defined networks,” *Int. Conf. Electr. Electron. Commun. Comput. Technol. Optim. Tech. ICEECCOT 2017*, vol. 2018-Janua, pp. 288–292, 2018, doi: 10.1109/ICEECCOT.2017.8284684.
[15] R. Kandoi and M. Antikainen, “Denial-of-service attacks in OpenFlow SDN networks,” *Proc. 2015 IFIP/IEEE Int. Symp. Integr. Netw. Manag. IM 2015*, pp. 1322–1326, 2015, doi: 10.1109/INM.2015.7140489.
[16] I. Iskandar and A. Hidayat, “Analisa Quality of Service (QoS) Jaringan Internet Kampus (Studi Kasus: UIN Suska Riau),” *J. CoreIT*, vol. 1, no. 2, pp. 67–76, 2015.
[17] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, “SDN controllers: A comparative study,” *Proc. 18th Mediterr. Electrotech. Conf. Intell. Effic. Technol. Serv. Citizen, MELECON 2016*, no. April, 2016, doi: 10.1109/MELCON.2016.7495430.