

Analisis Perbandingan Performansi Kontroler RYU Dan POX Berbasis Software Defined Network (SDN) Pada Routing OSPF

Farid Fajar Shodiq¹, Agus Prihanto²

^{1,2} Teknik Informatika, Universitas Negeri Surabaya

¹faridshodiq16051204043@mhs.unesa.ac.id

²agusprihanto@unesa.ac.id

Abstrak— Routing adalah proses pencarian lintasan yang akan di lalui oleh paket data. Software Defined Network (SDN) adalah sebuah konsep konfigurasi jaringan computer baru yang dimana data plane dan control plane dipisahkan. Kelebihan dari Software Defined Network dibandingkan dengan jaringan konvensional yaitu, pemusatan konfigurasi pada kontroler Software Defined Network. Pada penelitian ini, skenario jaringan Software Defined Network di bangun menggunakan software mininet untuk pembentukan topologinya dengan menerapkan algoritma Dijkstra pada kontroler RYU dan POX untuk pemilihan jalur terpendeknya. Pada pengujian dibentuk 6 topologi dengan pemberian bobot yang berbeda, tujuan perbedaan bobot untuk memastikan bahwa algoritma Dijkstra benar benar dapat di terapkan pada kontroler RYU dan POX. Pemberian bobot juga dilakukan agar kontroler dapat membentuk tabel routing untuk mengidentifikasi jalur tercepat yang akan di lalui oleh paket data tersebut. Hasil pengujian pada semua topologi diambil rata-ratanya untuk membandingkan performansinya. Hasil rata-rata dari semua pengujian adalah 1) pengujian waktu konvergensi pada RYU 0.4143s dan POX 0.5231s, 2) pengujian throughput yang di hasilkan RYU 12781.79kb/s dan POX 11412.24kb/s, 3) pengujian delay yang di hasilkan RYU 0.223ms dan POX 0.247ms, 4) pengujian jitter yang di hasilkan RYU 0.08ms dan POX 1.21ms. 5) rerata pengujian Packet Loss yang di hasilkan RYU 0 dan POX 0.2%.

Kata Kunci—SDN, Mininet, RYU, POX.

I. PENDAHULUAN

Pada saat ini teknologi sangat berkembang pesat mulai dari teknologi jaringan salah satunya ialah *Software Defined Network (SDN)*. *Software Defined Network* adalah sebuah konsep konfigurasi jaringan computer baru yang dimana data plane dan control plane dipisahkan, sedangkan pada jaringan tradisional kedua fungsi tersebut berada pada satu perangkat atau pengaturan yang sama. Perangkat yang memisahkan antara data plane dan control plane pada *Software Defined Network* adalah *Controller*. *Controller* memiliki peranan penting dalam *system software defined network* karena pada controller memiliki beberapa protocol jaringan seperti Openflow, Netconf, OF- config dan masih banyak controller lainnya. Beberapa jenis controller yang di gunakan pada *Software Defined Network* antara lain adalah RYU Controller, POX Controller, Floodlight, Maestro, dan ONOS controller. setiap controller memiliki perbedaan dalam kelebihan dan

kekurangan, sehingga mempengaruhi performansi jaringan tersebut.

Pada kontroler SDN bisa di konfigurasi tersendiri untuk Routingnya. *Routing Protocol* adalah proses pengiriman data/paket yang di teruskan ke jaringan satu dengan jaringan yang lainnya. *Routing Protocol* pada *Software Defined Network* menggunakan *RouteFlow* sebagai perantara pengiriman paket datanya. *Protocol RouteFlow* Juga di support oleh beberapa Jenis kontroler SDN. Pada *RouteFlow* memungkinkan kita Mengatur sebuah jaringan sekala besar secara terpusat.

Algoritma Dijkstra adalah algoritma yang di pakai atau di terapkan untuk memecahkan permasalahan dalam pemilihan jarak terpendek. Algoritma Dijkstra membuat jalur ataupun lintasan untuk membentuk tabel rute routing terpendeknya. Pada algoritma Dijkstra pada routing OSPF membutuhkan beberapa bobot pada simpul untuk dapat menentukan jalur terpendeknya. Pada setiap simpulnya memiliki node yang dapat menghasilkan angka angka untuk pemilihan jalur terpendeknya pada sebuah simpul. Pada penelitian yang di lakukan oleh Musril Harianto dengan judul Penerapan Open Shortest Path First (OSPF) untuk menentukan Jalur Terbaik Dalam Jaringan [1] penelitian tersebut menjelaskan bahwa Algoritma dijkstra digunakan untuk mencari rute terpendek dari suatu graf berarah dengan bobot sisi tidak negative dan algoritma dijkstra ini akan mencari jalur dengan nilai cost minimum di antara simpul yang terdapat pada graf.

Pada penelitian yang di lakukan oleh Ridha Mauldina. Penerapan Algoritma OSPF Menggunakan RouteFlow [2]. Jurnal tersebut membahas analisis jaringan *Software defined network* dengan menggunakan routing OSPF. Menggunakan simulasi mininet menggunakan protocol Openflow dan menggunakan controller POX dimana didapat pengujian 4 switch, 6 switch, 8 switch sampai 10 switch yang di simpulkan semakin bertambahnya switch mempengaruhi hasil dari nilai convergence timenya yang ikut bertambah juga. Pada penelitian lainnya yang di lakukan oleh Romi Afan Analisis efek Penggunaan Kontroler Ryu dan Pox pada performansi jaringan SDN [3] penelitian ini bertujuan untuk membandingkan performansi dari controller Ryu dan Pox. Menggunakan simulasi mininet menggunakan protocol openflow dimana routing pada penelitian ini menggunakan algoritma jonson. Dimana controller ryu dan pox bertugas sebagai control plane untuk mengatur aliran data. Pengujian pada penelitian ini menggunakan 6 switch, 8 switch dan 10 switch hasil pengujian QOS pada pengujian ini menghasilkan bahwa performa kontroler Ryu lebih unggul dari pada kontroler POX.

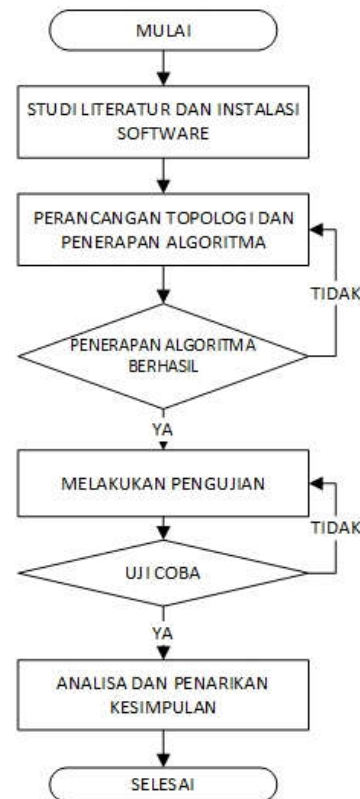
Pada jurnal yang di buat oleh Moh Wahyudi Putra. Analisi perbandingan Performasi kontroler Floodlight, Maestro, RYU, POX dan ONOS dalam Arsitektur Software Defined Network (SDN) [4] menyimpulkan bahwa setiap kontroler memiliki kelebihan tersendiri mulai dari perbedaan dari Throughput dan Latency dalam penambahan switch yang bervariasi. Pada kontroler Ryu dan POX Ketika menjalankan uji coba QOS pada throughput menunjukan Ryu lebih stabil di bandingkan kontroler POX.

Pada penelitian yang di lakukan oleh Shie-Yuan Wang pada tahun 2017 [5]. Penelitian ini membandingkan performasi antar kontroler RYU dan NOX menggunakan protocol Openflow. Pada penelitian ini menggunakan topologi spanning tree yang di jalakan pada software Estinet Openflow. pada setiap node pada topologi memiliki beban yang akan digunakan untuk pengiriman paket TCP pada setiap kontrolernya. Pada penelitian ini kontroler ryu mengalami kegagalan tautan pada saat penjalankan paket TCP selama beberapa kali sedangkan pada kontroler NOX mampu memilih jalur baru saat pengujian pengiriman paket TCP tersebut.

Berdasarkan latar belakang di atas penulis membangun sebuah jaringan yang berkonsep *Software Defined Network* (SDN) yang di simulasikan dengan software Mininet dengan menerapkan routing OSPF berbasis algoritma Dijkstra pada jaringan tersebut dan menggunakan RYU dan POX Sebagai Kontroler Software Defined Network. Menambahkan beberapa switch dalam konfigurasi jaringan untuk menguji coba performasi dari 2 kontroler tersebut untuk menghasilkan QOS (*Quality Of Services*) yang lebih baik antara kedua kontroler tersebut. Pengujian QOS terdiri dari convergence, througput, delay, jitter, dan packet loss. Hasil dari pengujian QOS sebagai tolak ukur performansi dari kontroler RYU dan POX dan pembuktian kontroler yang lebih baik.

II. METODOLOGI PENELITIAN

Pada alur penelitian dalam performasi kontroler Ryu dan Pox menggunakan algoritma Dijkstra untuk penentuan rute routing pada jaringan software defined network. Tahap penelitian berikutnya dengan membangun konfigurasi system topologi yang akan di diteliti. Penelitian ini menggunakan software emulasi jaringan bernama mininet yang di pasang pada perangkat pc dengan system operasi ubuntu 20.0.1. Selain software emulasi jaringan diperlukan juga controller ryu dan pox sebagai inti dari konfigurasi jaringan pada mininet. System konfigurasi yang telah di bangun akan di analisis kinerja dan diambil suatu kesimpulan dari masing masing controllernya. Alur pada penelitian ini dijelaskan pada gambar dibawah.



Gbr. 1 Alur Penelitian

Parameter pengujian performansi pada penelitian ini di dasarkan pada uji coba dengan cara end-to-end Qos yang meliputi waktu konvergensi, troghput, delay, jiter, dan paket loss. Proses routing menggunakan algoritma dijkstra ditandai dengan pemberian bobot pada switch sehingga akan terjadi penentuan bobot yang terkecil pada tiap topologi saat menjalankan kontroler RYU atau POX. Hasil dari pengujian dan analisis data tersebut akan di jadikan kesimpulan akhir dari penelitian.

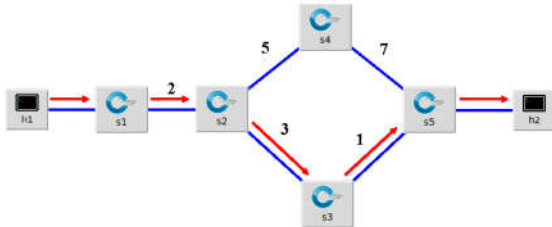
A. Perancangan Sistem

Pada perancangan system penelitian ini adalah penerapan algoritma dijkstra yang akan menjadi inti dari pemilihan rute terpendek pada *controller Software defined network* (SDN). Kedua *controller* tersebut menggunakan bahasa pemrograman python yang di berikan bobot untuk pemilihan jalur terpendek pada 3 topologi yang telah di konfigurasi. Proses instalasi di awali dengan menggunakan virtual box, pada virtual box di lakukan proses instalasi operating system Ubuntu 20.1 sebagai Operating system pada SDN. Proses instalasi perancangan topologi di awali dengan instalasi mininet pada Operating system yang telah di siapkan. Pada mininet di bangun bebrapa topologi yaitu topologi 1 sampai dengan topologi 6. Pada topologi tersebut di lakukan konfigurasi kontroler. Pada penelitian ini menggunakan kontroler RYU dan POX sebagai penerapan algortima dijkstra. Pada setiap topologi juga di lakukan pembobotan yang berfungsi untuk pemilihan jalur terpendek.

B. Penjelasan Topologi

Penelitian ini menggunakan 6 topologi yang akan diujikan, dan topologi memiliki beban pada setiap jalurnya yang telah di konfigurasi pada kontroler RYU dan POX menggunakan algoritma djikstra untuk pemilihan rutenya. Berikut rancangan topologi jaringan yang digunakan :

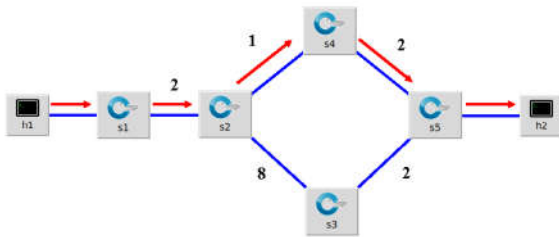
1) Topologi 1



Gbr. 2 Topologi 1

Pada topologi 1 terdiri 2 Host (H) dan 5 switch (S) terhubung oleh kontroler RYU atau POX. Pembobotan pada topologi ini untuk S1 ke S2 yaitu 2, S2 ke S3 yaitu 3, dan S3 ke S5 yaitu 1.

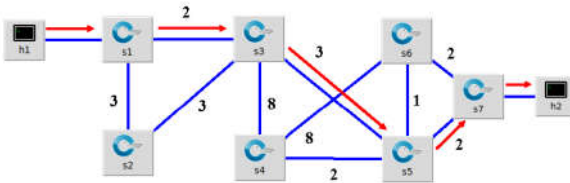
2) Topologi 2



Gbr. 3 Topologi 2

Pada topologi 2 terdiri 2 Host (H) dan 5 switch (S) terhubung oleh kontroler RYU atau POX. Pembobotan pada topologi ini untuk S1 ke S2 yaitu 2, S2 ke S4 yaitu 1, dan S4 ke S5 yaitu 2.

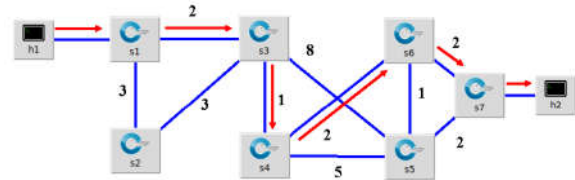
3) Topologi 3



Gbr. 4 Topologi 3

Pada topologi 3 terdiri 2 Host (H) dan 7 switch (S) terhubung oleh kontroler RYU atau POX. Pembobotan pada topologi ini untuk S1 ke S3 yaitu 2, S3 ke S5 yaitu 3, dan S5 ke S7 yaitu 2.

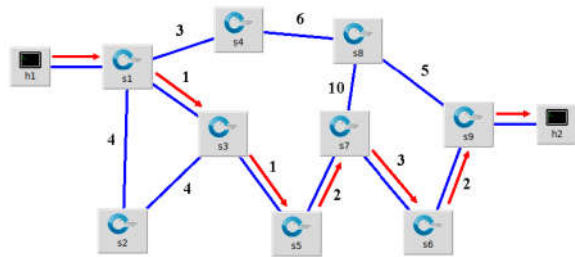
4) Topologi 4



Gbr. 5 Topologi 4

Pada topologi 4 terdiri 2 Host (H) dan 7 switch (S) terhubung oleh kontroler RYU atau POX. Pembobotan pada topologi ini untuk S1 ke S3 yaitu 2, S3 ke S4 yaitu 1, S4 ke S6 yaitu 2, dan S6 ke S7 yaitu 2.

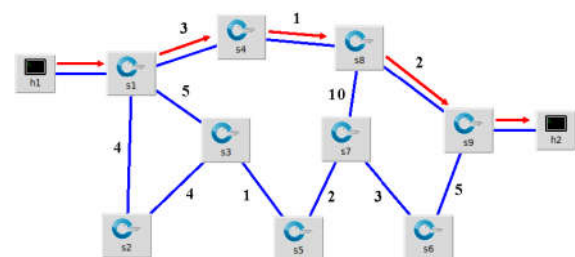
5) Topologi 5



Gbr. 6 Topologi 5

Pada topologi 5 terdiri 2 Host (H) dan 9 switch (S) terhubung oleh kontroler RYU atau POX. Pembobotan pada topologi ini untuk S1 ke S3 yaitu 1, S3 ke S5 yaitu 1, dan S5 ke S7 yaitu 2. S7 ke S6 yaitu 4, dan S6 ke S9 yaitu 2.

6) Topologi 6



Gbr. 7 Topologi 6

Pada topologi 6 terdiri 2 Host (H) dan 9 switch (S) terhubung oleh kontroler RYU atau POX. Pembobotan pada topologi ini untuk S1 ke S4 yaitu 3, S4 ke S8 yaitu 1, dan S8 ke S9 yaitu 2.

C. Scenario pengujian

Pengujian performasi jaringan dilakukan pada 6 topologi dengan cara menghitung waktu convergence dan dengan cara end-to-end menggunakan *Software Distributed Internet Traffic Generator* (D-ITG). Pengujian di lakukan sebanyak 5 kali dengan mengirim paket UDP dengan jumlah 25 dengan ukuran paket 1Gb dengan durasi pengujian selama 10 detik.

Lalu di ambil rerata dari hasil yang di dapat pada setiap pengujian dan selanjutnya di bandingkan hasil dari pengujian untuk menentukan kontroler mana yang memiliki performansi terbaik. Pada pengujiannya berdasarkan parameter convergence, throughput, delay, jitter dan packet loss. Berikut penjelasan dari parameter yang akan di gunakan

1) Waktu Konvergensi

Waktu konvergensi adalah waktu perhitungan yang di perlukan oleh suatu jaringan untuk membentuk tabel routing dari awal sampai akhir untuk mengetahui jalur mana yang akan di lalui suatu paket untuk sampai pada alamat tujuannya [6]

2) Throughput

Throughput adalah nilai kecepatan pengiriman data berdasarkan satuan bit per second (bps) yang dapat dihitung berdasarkan total paket yang sukses diterima pada waktu interval tertentu dan dibagi dengan waktu interval tersebut. Semakin besar nilai Troughput maka semakin bagus maka semakin baik performasi jaringan tersebut [7]. Perhitungan rumus Troghput ditampilkan pada persamaan (1)

$$\text{Throughput} = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman data}} \quad (1)$$

3) Delay

Delay atau latency adalah waktu yang dibutuhkan paket untuk mencapai tujuan, karena adanya antrian, atau jeda yang di butuhkan node asal ke node tujuan [7]. Perhitungan rumus Delay ditampilkan pada persamaan (2). Untuk standarisasi nilai delay menurut TIPHON dapat dilihat pada tabel 1.

$$\text{Rata - rata Delay} = \frac{\text{total delay}}{\text{total paket diterima}} \quad (2)$$

TABEL I
STANDARISASI NILAI DELAY VERSI TIPHON

Kategori Delay	Besar Delay
Sangat Bagus	< 150 ms
Bagus	150 s/d 300 ms
Sedang	300 s/d 450 ms
Buruk	> 450 ms

4) Jitter

Jitter didefinisikan jumlah variasi delay yang di sebabkan oleh variasi panjang paket dalam waktu pengolahan paket dari awal sampai akhir tujuan [7]. Perhitungan rumus Jitter ditampilkan pada persamaan (3). Untuk Standarisasi nilai jitter menurut TIPHON dapat dilihat pada Tabel 2

$$\text{Jitter} = \frac{\text{total variasi delay}}{\text{total paket yang diterima}} \quad (3)$$

TABEL III
STANDARISASI NILAI JITTER VERSI TIPHON

Kategori Degradasi	Peak Jitter
Sangat Bagus	0 ms
Bagus	0 s/d 75 ms
Sedang	76 s/d 125 ms
Buruk	126 > 225 ms

5) Packet Loss

Packet loss didefinisikan sebagai kegagalan transmisi paket data mencapai tujuannya [7]. Perhitungan rumus Packet Loss ditampilkan pada persamaan (4). Untuk Standarisasi nilai packet loss menurut TIPHON dapat dilihat pada Tabel 3.

$$\text{Packet loss} = \frac{\text{paket dikirim} - \text{paket diterima}}{\text{paket dikirim}} \quad (4)$$

TABEL IIIII
STANDARISASI NILAI JITTER VERSI TIPHON

Kategori Degradasi	Packet Loss
Sangat Bagus	0 %
Bagus	3 %
Sedang	15 %
Buruk	25 %

III. HASIL DAN PEMBAHASAN

Saat seluruh persiapan Konfigurasi routing OSPF simulasi telah dilakukan, maka skenario pengujian konfigurasi topologi bisa dijalankan. Skenario pengujian performasi routing OSPF pada controller Ryu dan POX menggunakan spesifikasi :

Processor	: 2,4 GHz Quad Core Intel Core i5
RAM	: 8 GB 2133 MHz DDR4
Storage	: 1 Tera HDD
Mininet	: 2.3.0
Virtualbox	: 6.1.26 r145957
Ryu	: 4.34
POX	: 1.4
D-ITG	: 2.8.1

Berikut hasil dari pengujian yang telah dilakukan.

A. Hasil Waktu Konvergensi

Pada tabel di bawah ini hasil dari pengujian waktu convergence pada tiap topologi dengan pengujian sebanyak 5 kali dan dihasilkan rerata untuk kontroler RYU dan POX.

TABEL IVV
PERBANDINGAN RERATA KONVERGENSI KONTROLER RYU DAN POX

No	RYU (s)	POX (s)
A	Topologi 1	
1	0.2397	0.1032
2	0.2104	0.2036
3	0.0983	0.2124
4	0.3295	0.2350

No	RYU (s)	POX (s)
5	0.2402	0.2460
B	Topologi 2	
6	0.2812	0.1589
7	0.2613	0.1711
8	0.3552	1.2637
9	0.2334	0.3545
10	0.2012	1.3201
C	Topologi 3	
11	0.6576	0.3345
12	0.5564	3.2063
13	0.7457	0.3204
14	0.7125	0.2134
15	0.5191	0.2024
D	Topologi 4	
16	0.3435	0.4837
17	0.6045	0.3143
18	0.4458	1.1619
19	0.7474	0.5487
20	0.4818	0.3339
E	Topologi 5	
21	0.5387	0.2514
22	0.3287	0.3273
23	0.4980	0.2895
24	0.3941	0.4268
25	0.3830	0.4695
F	Topologi 6	
26	0.4322	0.1794
27	0.4136	0.2843
28	0.4622	0.2195
29	0.3702	0.4221
30	0.4421	1.1015
Rerata	0.4143	0.5231

Pada tabel IV didapatkan hasil untuk kontroler RYU reratanya 0.4143s dan untuk kontroler POX 0.5231s, selisih kedua kontroler 0,1088s. selisih keduanya terjadi dikarenakan pada kontroler RYU lebih cepat dan lebih stabil dalam pembentukan tabel routing awal dari pada kontroler POX.

B. Hasil Pengujian Throughput

Pada tabel di bawah ini hasil dari pengujian Throughput pada tiap topologi dengan pengujian sebanyak 5 kali dan dihasilkan rerata untuk kontroler RYU dan POX.

TABEL V
PERBANDINGAN RERATA THROUGHPUT KONTROLER RYU DAN POX

No	RYU (Kb/s)	POX (Kb/s)
A	Topologi 1	
1	12779.22	10529.25
2	12803.31	11677.06
3	12779.89	12074.02
4	12798.67	12368.70
5	12797.13	11843.33
B	Topologi 2	
6	12804.95	12090.44
7	12804.21	11214.06
8	12807.38	11360.42
9	12802.06	10964.46

No	RYU (Kb/s)	POX (Kb/s)
10	12801.94	12071.39
C	Topologi 3	
11	12769.24	12181.37
12	12788.62	10784.33
13	12771.75	10481.87
14	12815.19	11349.94
15	12779.56	9903.99
D	Topologi 4	
16	12781.47	12132.27
17	12786.66	11474.46
18	12518.72	12046.23
19	12782.77	11399.30
20	12802.40	11190.74
E	Topologi 5	
21	12794.33	11141.12
22	12801.72	11400.61
23	12786.94	10388.02
24	12786.16	10247.85
25	12794.50	12236.67
F	Topologi 6	
26	12781.99	12197.65
27	12777.09	10648.54
28	12780.06	11444.68
29	12786.71	11468.86
30	12789.03	12055.48
Rerata	12781.79	11412.24

Pada tabel V didapatkan hasil untuk kontroler RYU reratanya 12781.79 Kb/s dan untuk kontroler POX 11412.24 Kb/s, selisih kedua kontroler adalah 1369,55 Kb/s. Selisih keduanya terjadi di karenakan jumlah switch yang di lalui paket mempengaruhi kinerja dari kontroler POX.

C. Hasil Pengujian Delay

Pada tabel di bawah ini hasil dari pengujian Delay pada tiap topologi dengan pengujian sebanyak 5 kali dan dihasilkan rerata untuk kontroler RYU dan POX.

TABEL VI
PERBANDINGAN RERATA DELAY KONTROLER RYU DAN POX

No	RYU (ms)	POX (ms)
A	Topologi 1	
1	0.205	0.231
2	0.202	0.234
3	0.219	0.266
4	0.205	0.232
5	0.207	0.235
B	Topologi 2	
6	0.214	0.231
7	0.202	0.233
8	0.206	0.229
9	0.222	0.229

No	RYU (ms)	POX (ms)
10	0.200	0.256
C	Topologi 3	
11	0.216	0.231
12	0.211	0.206
13	0.230	0.245
14	0.229	0.267
15	0.216	0.024
D	Topologi 4	
16	0.231	0.247
17	0.229	0.252
18	0.223	0.274
19	0.229	0.262
20	0.226	0.283
E	Topologi 5	
21	0.257	0.305
22	0.25	0.297
23	0.253	0.317
24	0.257	0.33
25	0.256	0.296
F	Topologi 6	
26	0.224	0.231
27	0.225	0.240
28	0.206	0.248
29	0.227	0.253
30	0.207	0.234
Rerata	0.223	0.247

Pada tabel VI didapatkan hasil untuk kontroler RYU dengan reratanya 0.223 ms dan untuk kontroler POX 0.247 ms, selisih kedua kontroler 0.024 ms. selisih keduanya terjadi dikarenakan kontroler Ryu dalam proses pembentukan tabel routing lebih cepat dari pada kontroler POX yang mengakibatkan antrian paket dalam pengiriman menjadi lebih lama.

D. Hasil Pengujian Jitter

Pada tabel di bawah ini hasil dari pengujian Jitter pada tiap topologi dengan pengujian sebanyak 5 kali dan dihasilkan rerata untuk kontroler RYU dan POX.

TABEL VII
PERBANDINGAN RERATA JITTER KONTROLER RYU DAN POX

No	RYU (ms)	POX (ms)
A	Topologi 1	
1	0.101	0.85
2	0.047	1.288
3	0.11	0.223
4	0.048	0.841
5	0.066	0.849
B	Topologi 2	
6	0.072	0.837
7	0.066	0.884
8	0.054	1.04

No	RYU (ms)	POX (ms)
9	0.048	1.091
10	0.043	0.889
C	Topologi 3	
11	0.075	1.538
12	0.068	0.914
13	0.043	1.311
14	0.046	0.839
15	0.059	1.244
D	Topologi 4	
16	0.074	1.097
17	0.064	1.046
18	0.346	2.017
19	0.069	1.199
20	0.085	0.951
E	Topologi 5	
21	0.07	2.018
22	0.092	3.861
23	0.087	1.094
24	0.075	1.835
25	0.083	1.742
F	Topologi 6	
26	0.036	0.889
27	0.036	1.017
28	0.072	0.939
29	0.148	0.937
30	0.051	0.912
Rerata	0.08	1.21

Pada tabel VII didapatkan hasil untuk kontroler RYU dengan reratanya 0.08 ms dan untuk kontroler POX 1.21 ms, selisih kedua kontroler 1.13 ms. selisih keduanya terjadi dikarenakan kontroler RYU dalam proses pembentukan tabel routing lebih cepat dari pada kontroler POX yang mengakibatkan pemilihan rute ke penerima lebih lama di bandingkan kontroler RYU.

E. Hasil Pengujian Packet Loss

Pada tabel di bawah ini hasil dari pengujian Packet Loss pada tiap topologi dengan pengujian sebanyak 5 kali dan dihasilkan rerata untuk kontroler RYU dan POX.

TABEL VIII
PERBANDINGAN RERATA PACKET LOSS KONTROLER RYU DAN POX

No	RYU (%)	POX (%)
A	Topologi 1	
1	0	0
2	0	1
3	0	2
4	0	0
5	0	0
	Topologi 2	
6	0	0
7	0	0

No	RYU (%)	POX (%)
8	0	0
9	0	0
10	0	0
Topologi 3		
11	0	1
12	0	0
13	0	1
14	0	0
15	0	0
Topologi 4		
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
Topologi 5		
21	0	1
22	0	0
23	0	0
24	0	1
25	0	0
Topologi 6		
26	0	0
27	0	0
28	0	0
29	0	0
30	0	0
Rerata	0	0.2

Pada tabel VIII menunjukkan hasil RYU kontroler tidak mengalami packet loss sedangkan Pada controller POX mengalami paket loss sebesar 0.2% pada seluruh pengujian topologi.

Pada semua pengujian rerata dibandingkan untuk performansi dari kontroler RYU dan kontroler POX pada 5 parameter. Pengujian convergence time waktu yang lebih rendah memiliki performa yang lebih baik. Pada pengujian Throughput semakin tinggi nilai Throughput semakin baik performansi jaringan tersebut. Pada pengujian Delay semakin rendah waktu yang di dapat semakin baik performansi jaringan tersebut. Pada pengujian Jitter semakin rendah waktu yang di dapat semakin baik performansi jaringan tersebut. Pada paket loss semakin rendah persentase packet loss yang di dapat semakin baik performansi jaringan tersebut

Setelah di lakukan pengujian QOS pada ke 6 topologi pada pengujian parameter Convergence, Throughput, Delay, Jitter, dan Packet Loss didapatkan hasil untuk perbandingan kedua kontroler dengan dilakukan sistem point, jika kontroler yang lebih baik diberi point 2 dan lainnya diberi point 1. Pada hasil performansi di dapatkan point pada tabel di bawah ini.

TABEL IX
TABEL POINT KONTROLER RYU DAN POX

Parameter	RYU	POX
Convergence	2	1
Troughput	2	1
Delay	2	1
Jitter	2	1
Packet Loss	2	1
Total Point	10	5

Pada tabel diatas bisa dilihat bahwa kontroler RYU mendapatkan point lebih banyak daripada kontroler POX pada semua parameter penilaian. Perbandingan point dari kedua kontroler dapat dilihat pada grafik di bawah ini.

Gbr. 8 Grafik point kontroler RYU dan POX

Pada Grafik di atas dapat dilihat bahwa point RYU lebih besar dari pada POX dalam pengujian QOS. Kontroler RYU baik pada seluruh aspek pengujian di karenakan kontroler RYU lebih stabil dalam pengiriman data dengan beberapa jumlah switch yang di lalui sedangkan kontroler POX performa dibawah RYU.

IV. KESIMPULAN

Berdasarkan hasil dari pengujian yang telah dilakukan, maka dapat ditarik kesimpulan dalam perbandingan performansi pada kontroler RYU dan POX :

1. Penerapan algoritma djikstra pada kontroler RYU dan POX bisa di jalankan secara optimal dengan pemberian bobot antar node yang mempengaruhi jalur yang akan di lalui oleh paket data.
2. Berdasarkan standart TIPHON pada uji coba QOS yang meliputi parameter Throughput, Delay, Jitter, dan Packet Loss kedua kontroler RYU dan POX masuk kategori sangat bagus dan tidak melebihi dari standart TIPHON pada semua topologi.
3. Pada pengujian waktu konvergensi kontroler RYU menghasilkan waktu yang lebih singkat dari pada kontroler POX dengan selisih waktu 0,1088s selisih waktu ini dikarenakan saat kedua kontroler pada posisi *standby*

menunggu perintah dari mininet kontroler POX memiliki penghitungan bobot lebih lama dalam proses pembentukan table routingnya. Sedangkan RYU memiliki waktu yang lebih stabil dan cepat dalam pembentukan table routing pada semua topologi.

4. Pada pengujian throughput antara kontroler RYU dan kontroler POX memiliki selisih rerata 1369,55 Kb/s Pada pengujian Delay antara kontroler RYU dan kontroler POX memiliki selisih rerata 0.024 ms. Pada pengujian Jitter antara kontroler RYU dan kontroler POX memiliki selisih rerata 1.13 ms. Pada pengujian Packet Loss antara kontroler RYU dan kontroler POX memiliki selisih rerata 0.2%. kontroler RYU lebih unggul dan lebih stabil dalam pengiriman paket dalam semua topologi sedangkan Kontroler POX terjadi penurunan pada pengujian 4 parameter. Penurunan performa dikarenakan jumlah switch yang dilalui oleh paket bertambah yang mengakibatkan performansi dari kontroler POX menurun.

V. SARAN

Pengujian Performansi pada kontroler RYU dan POX bisa di kembangkan dengan menambahkan beberapa switch pada setiap topologi. Penambahan parameter yang di ujikan seperti mengukur CPU dan *memory usage*.

REFERENSI

- [1] H. A. Musril, "Penerapan Open Shortest Path First (Ospf) Untuk Menentukan Jalur Terbaik Dalam Jaringan," *J. Elektro dan Telekomun. Terap.*, vol. 4, no. 1, p. 421, 2017, doi: 10.25124/jett.v4i1.989.
- [2] R. M. Negara and R. Tulloh, "Analisis Simulasi Penerapan Algoritma OSPF Menggunakan RouteFlow pada Jaringan Software Defined Network (SDN)," vol. 9, no. 1, pp. 75–83, 2017.
- [3] M. S. . Romi Afan1, Ir. Agus Virgono., M.T.2, Drs. Ir. R Ruman M., Bc.TT., "ANALISIS EFEK PENGGUNAAN KONTROLER RYU DAN POX PADA PERFORMANSI JARINGAN SDN," vol. 5, no. 3, pp. 6025–6041, 2018.
- [4] M. W. Putra, E. S. Pramukantoro, and W. Yahya, "Analisis Perbandingan Performansi Kontroler Floodlight , Maestro , RYU , POX dan ONOS dalam Arsitektur Software Defined Network (SDN)," vol. 2, no. 10, pp. 3779–3787, 2018.
- [5] S. Wang, "Comparisons of SDN OpenFlow Controllers over EstiNet : Ryu vs . NOX," *Int. Symp. Adv. Softw. Defin. Networks, April 19-24, 2015, Barcelona, Spain*, no. Fedora 14, pp. 1–6, 2015.
- [6] A. Kartika, A. Rahmasari, and I. M. Suartana, "Penerapan Algoritma Floyd-Warshall untuk Pemilihan Rute Routing OSPF pada Jaringan SDN," *J. Inf. Eng. Educ. Technol.*, vol. 04, pp. 29–35, 2020.
- [7] R. Wulandari, "ANALISIS QoS (QUALITY OF SERVICE) PADA JARINGAN INTERNET (STUDI KASUS : UPT LOKA UJI TEKNIK PENAMBANGAN JAMPANG KULON – LIPI)," *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 162–172, 2016, doi: 10.28932/jutisi.v2i2.454.