

Simulasi Jaringan SDN menggunakan controller RYU Pada Mininet Dengan 5 Topologi Jaringan

Muhamad Deris Ramdhani¹, Bambang Sugiarto², Ade Rukmana³

¹ Fakultas Teknik Universitas Garut, Garut, Jawa Barat, 44151, Indonesia

² Fakultas Teknik Universitas Garut, Garut, Jawa Barat, 44151, Indonesia

³ Fakultas Teknik Universitas Garut, Garut, Jawa Barat, 44151, Indonesia

Korespondensi: mdrtn0@gmail.com

ARTICLE HISTORY

Received:23-12-2021

Revised:28-12-2021

Accepointed:28-12-2021

Abstrak

Perbedaan pada topologi jaringan SDN cukup mempengaruhi pada aspek kinerja QoS. Maka diperlukan penelitian untuk menguji topologi yang paling bagus kinerjanya pada jaringan SDN menggunakan *Controller* RYU. Ada 5 topologi yang diujikan yaitu topologi *linear*, *tree*, *ring*, *fullmesh* dan *star*. Parameter yang diujikan adalah *Jitter*, *Throughput*, *Packet loss* dan *Delay* pada protokol TCP dan UDP. Menurut hasil pengujian didapatkan topologi yang paling bagus adalah topologi *ring* dimana nilai-nilainya berada pada index 4 dan untuk hasil pengujian topologi yang paling jelek menurut nilai QoS yaitu topologi *linear* dimana nilai yang didapatkan berada pada index 2 untuk *throughput* protokol TCP dan index 3 untuk *throughput* pada *protocol* UDP.

Kata kunci: QoS, SDN, Jaringan, Mininet, RYU

SDN Network Simulation using RYU controller on Mininet With 5 Network Topologies

Abstract

The difference in SDN network topology is quite influential on the aspect of QoS performance. So, research is needed to test which topology has the best performance on SDN networks using the ryu controller. There are five topologies tested on SDN networks. The topologies are linear, tree, ring, fullmesh and star. The parameters QoS tested in the topology are Delay, Throughput, Packet loss and Jitter on TCP and UDP protocols. According to the test results of those parameters, the best topology that has the best performance on SDN networks is ring topology where this topology has values at index 4 for all protocols. For the worst topology on SDN networks is linear topology where this topology has value at index 2 for throughput TCP protocol and index 3 for throughput on the UDP protocol.

Keyword: RYU, Mininet, network, QoS

1. Pendahuluan

Kini jaringan statis konvensional sudah mulai digantikan dengan jaringan dinamis yang bersifat *programmable*. Software Defined Network (SDN) merupakan salah satu contoh dari perkembangan jaringan dinamis yang bersifat *programmable* dengan paradigma baru untuk memberikan kemudahan kepada pengguna dalam mendesain, membangun dan mengelola jaringan komputer. Software defined network menyediakan pendistribusian jaringan yang dikelola secara terpusat (centralized) untuk memudahkan layanan jaringan agar lebih efisien, otomatis dan cepat. *Controller* jaringan merupakan *software* yang bersifat fleksibel untuk dikonfigurasi sehingga mekanisme jaringan (forwarder) dapat di kontrol dengan mudah. *Controller* dalam SDN secara langsung melakukan kendali terhadap *data path* dari perangkat. Salah satu *Controller* yang cukup banyak digunakan pada SDN adalah Ryu [1].

Dalam jaringan komputer dua atau lebih komputer dihubungkan bersama-sama melalui media dan komunikasi data perangkat untuk tujuan mengkomunikasikan data dan berbagi sumber daya. Istilah topologi dalam komunikasi jaringan mengacu pada cara komputer atau *workstation* dihubungkan bersama dalam jaringan. Tipe dasar dari topologi jaringan adalah Topologi *Bus*, Topologi *Star*, *Ring* Topologi, Topologi *Mesh* dan Topologi *Tree*. Pada setiap topologi memiliki karakteristik yang berbeda salah satu akibat dari karakteristik yang berbeda adalah perbedaan pada kinerja. sehingga dibutuhkan pengujian untuk mengetahui topologi mana yang paling optimal pada SDN menggunakan *Controller* RYU.

Parameter pengukuran TCP dan UDP adalah *Throughput*, *Delay*, *Packet loss*, dan *Jitter*, dan juga dibandingkan nilai QoS antara jaringan SDN dan Konvensional dibuat dengan topologi yang sama, dan nilai rata-rata dari QoS TCP adalah *Throughput* Konvensional 2,07 Mbit/s, SDN 2,03 Mbit/s untuk delay Konvensional 604,58 ms, SDN 0,21 ms untuk jitter Konvensional 3,17ms, SDN 0,069ms sedangkan UDP *Throughput* Konvensional 2,07 Mbit/s dan SDN 1,99 Mbit/s untuk delay Konvensional 598,84ms, SDN 0,18ms untuk jitter Konvensional 3,13ms, SDN 0,03ms, untuk packet loss Konvensional 0,007%, SDN 0%. [2]

Berdasarkan latar belakang yang telah di jelaskan di atas maka penulis membuat sebuah penelitian yaitu “Simulasi Jaringan SDN menggunakan *Controller* RYU Pada Mininet Dengan 5 Topologi Jaringan”.

2. Metode

Metode penelitian yang digunakan dalam penelitian ini adalah menggunakan metode simulasi. Simulasi pada penelitian ini menggunakan software MININET dan membuat semua topologi untuk diuji dengan menggunakan *Controller* RYU.

2.1 Alat dan Bahan

2.1.1 Mininet

Mininet Sebuah emulator jaringan yang dapat membuat suatu jaringan virtual yang realistis [3]

2.1.2 SDN

SDN mengusung sebuah paradigma baru dalam jaringan yang awalnya bersifat terdistribusi dan cenderung tertutup menjadi jaringan yang bersifat terpusat, *programmable* dan terbuka (open source), Dengan demikian memungkinkan seorang peneliti atau bahkan administrator untuk mengatur jaringan sesuai dengan kebutuhan. Tidak harus terpaku pada *protocol*, API dan standar yang ditetapkan oleh para vendor terhadap *device*/perangkat mereka. SDN memberikan konsep *network topology virtualisation* dan memungkinkan administrator untuk melakukan *customize* pada *control plane* [4]

2.1.3 Desain Jaringan

Desain jaringan adalah salah satu faktor yang mempengaruhi performa jaringan. Desain jaringan yang baik akan meningkatkan performa jaringan dari segi *Reliability*, *security*, *scalability*, dan *manageability*. *Reliability* merujuk kepada seberapa sering jaringan mengalami gangguan dan kemampuannya dalam menghadapi gangguan. *Reliability* dicapai dengan mengimplementasikan *redundancy* yaitu menerapkan beberapa *link* sebagai *link* cadangan. [5]

2.1.4 Arsitektur SDN

Dalam konsep SDN, tersedia *open interface* yang memungkinkan sebuah entitas software/aplikasi untuk mengendalikan konektivitas yang disediakan oleh sejumlah sumber-daya jaringan, mengendalikan aliran trafik yang melewatinya serta melakukan inspeksi terhadap atau memodifikasi trafik tersebut. [6]

2.1.5 Open Flow

OpenFlow adalah protokol terbilang relatif baru yang dirancang dan diimplementasikan di *Stanford University* pada tahun 2008. Protokol *OpenFlow* telah berkembang sejak dimulainya proses standarisasi oleh *Open Network Foundation* (ONF), yaitu sebuah organisasi yang berfokus dalam pengembangan protokol *OpenFlow*, dari versi 1.0–1.5. *OpenFlow* terus berkembang dengan menambahkan fitur-fitur baru, dibawah ini adalah *roadmap* perkembangan *OpenFlow* dari versi 1.0–1.5. [7]

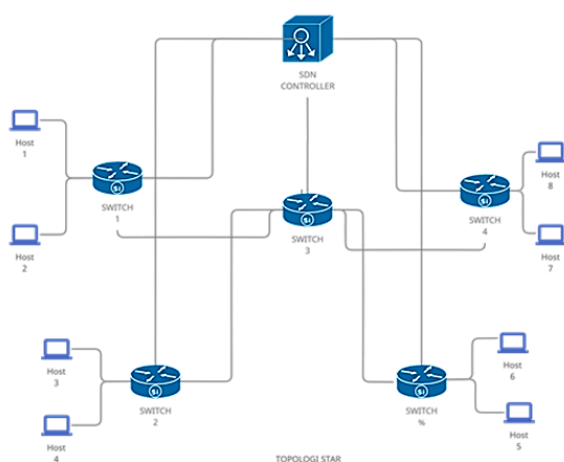
2.1.6 Controller SDN

Controller SDN adalah aplikasi SDN yang mengelola *flow control* untuk mengaktifkan *intelligence networking*. Pengontrol SDN bekerja berdasarkan protokol seperti *OpenFlow* yang memungkinkan server memberitahu kemana paket dikirimkan. [8]

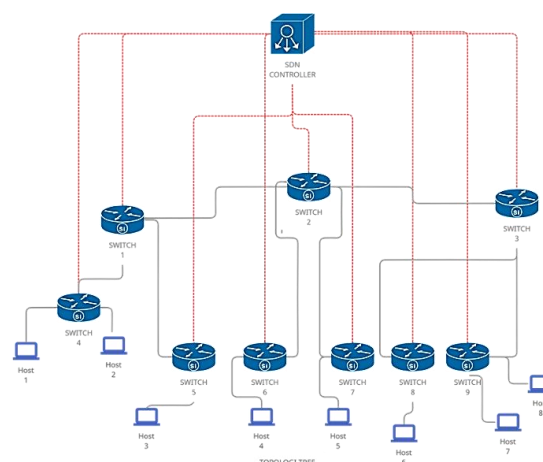
Tabel 1. Alat dan Bahan

No	Perangkat Keras	Kegunaan
1.	Laptop/ <i>Personal Computer</i>	Spesifikasi dari laptop/ <i>personal computer</i> minimal dapat diinstallkan semua perangkat lunak yang dibutuhkan.
	Perangakt Lunak	Kegunaan
2.	Mininet	Pada penelitian ini mininet dijadikan sebuah wadah untuk simulasi SDN. Mininet menciptakan sebuah jaringan virtual yang realistis dan berjalan seperti kernel yang sama seperti aslinya.
3.	Sublime Text Editor	Sublime text disini adalah untuk membuat sebuah topologi atau <i>custom</i> topologi yang dimana pada simulator mininet hanya ada topologi sederhana.
4.	Ryu Controller	RYU <i>controller</i> merupakan <i>controller</i> yang dipakai pada penelitian ini. Dimana <i>controller</i> ini mendukung <i>open flow</i> dari versi 0.1 sampai 1.5.
5.	Iperf	Iperf adalah alat untuk mengukur <i>bandwidth</i> dan kualitas link jaringan, tersedia pada sistem Linux, berukuran kecil dan dapat diinstall dengan mudah. Iperf digunakan untuk ujicoba QoS pada SDN di MikroTik.
6.	Wireshark	Wireshark digunakan sebagai alat analisa kualitas SDN di MikroTik menurut parameter QoS

2.2 Desain Topologi Jaringan

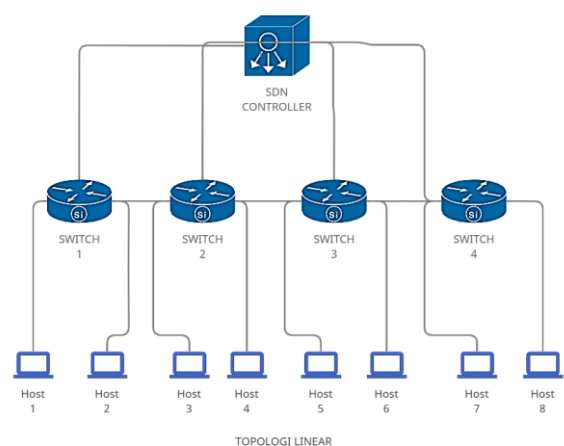


Gambar 1. Topologi Jaringan Star

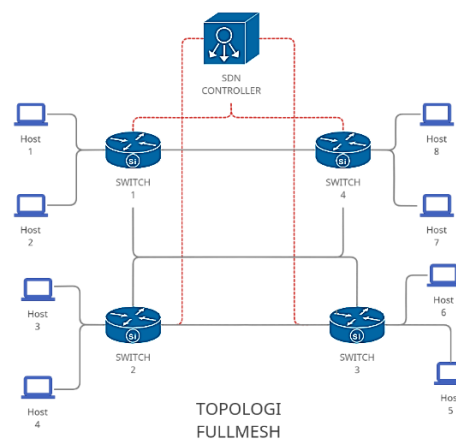


Gambar 2. Topologi Jaringan Tree

Topologi star memiliki *node* inti/tengah yang disambungkan ke *node* lainnya. Pada topologi star ini *switch* yang digunakan berjumlah 10 dan *host* yang berjumlah 20. Topologi *Tree* merupakan gabungan dari beberapa topologi *star* yang dihubungkan dengan topologi *bus*, jadi setiap topologi *star* akan terhubung ke topologi *star* lainnya menggunakan topologi *bus*, biasanya dalam topologi ini terdapat beberapa tingkatan jaringan, dan jaringan yang berada pada tingkat yang lebih tinggi dapat mengontrol jaringan yang berada pada tingkat yang lebih rendah. Pada topologi ini *switch* yang digunakan berjumlah 10 dan *host* yang digunakan adalah 20.

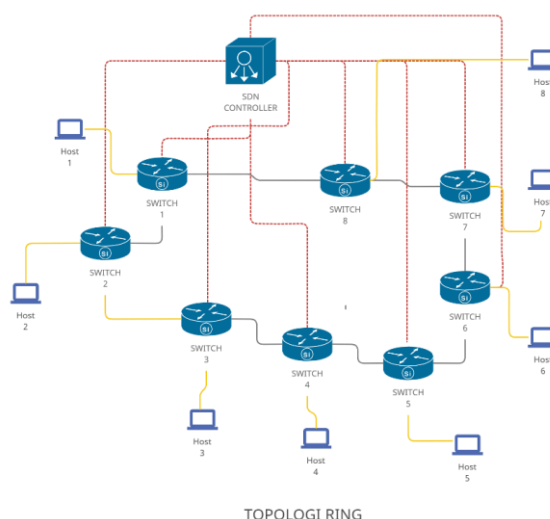


Gambar 3. Topologi Jaringan Linear



Gambar 4. Topologi Jaringan Fullmesh

Topologi *linear* merupakan perluasan dari jenis topologi *bus*, yang mana kabel utama di dalam jaringan harus dihubungkan dengan setiap titik-titik yang ada di komputer dengan *T-Connector*. Adapun topologi *Fullmesh* merupakan topologi gabungan dari topologi *Ring* dan *Star*.

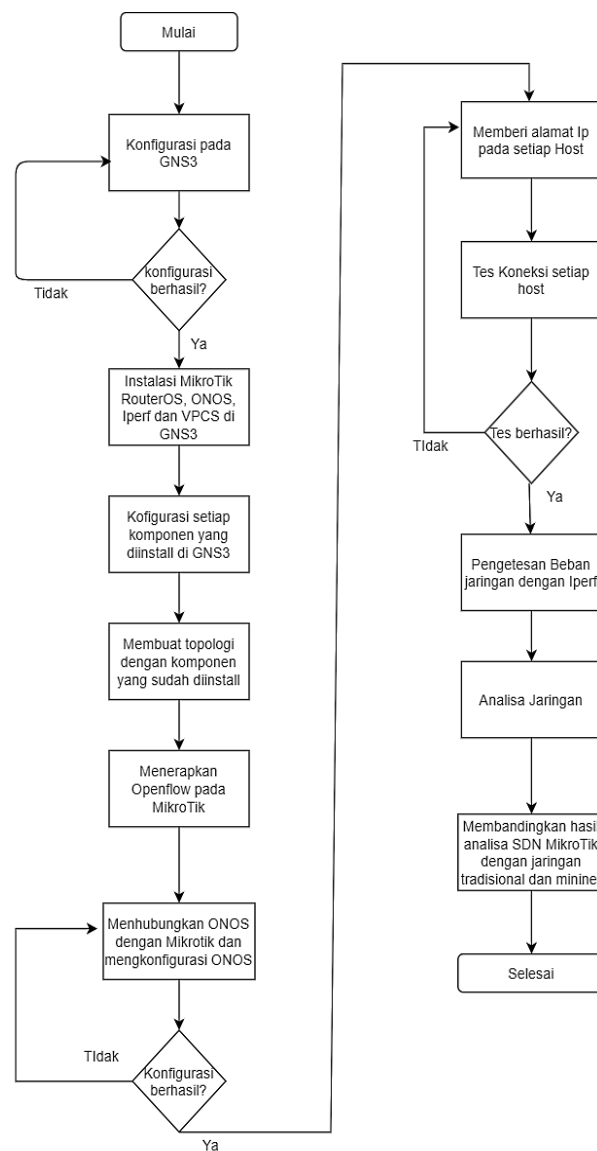


Gambar 5. Topologi Jaringan Ring

Topologi *ring* digunakan dalam jaringan yang memiliki *performance* tinggi, jaringan yang membutuhkan *bandwidth* untuk fitur yang *time-sensitive* seperti video dan audio, atau ketika *performance* dibutuhkan saat komputer yang terhubung ke jaringan dalam jumlah yang banyak. Pada topologi ini *switch* berjumlah 10 dan *host* berjumlah 20.

Topologi diatas adalah topologi yang sudah dibuat dari mulai topologi bawaan mininet dan topologi kostum yang sudah di buat di *text editor*. Komponen yang dipakai di dalam topologi ini adalah *ryu* sebagai kontroler yang berperan untuk mengatur lalu lintas jaringan agar bisa di atur secara terpusat. *Switch* berperan sebagai penghubung antar jaringan agar bisa melakukan pertukaran paket serta meneruskan data ke berbagai perangkat tujuan. *Host* digunakan untuk mengirim atau menerima paket. Gambar di atas rata-rata memiliki 1 kontroler *ryu*, 10 *switch* dan 20 *host*.

2.3. FlowChart Sistem



Gambar 6. Flowchart Sistem

3. Hasil dan Pembahasan

3.1. Hasil Pengukuran QoS Simulasi SDN Mininet

Pengukuran QoS pada Simulasi SDN dilakukan dengan melakukan pembebanan pada topologi menggunakan Iperf. Iperf adalah sebuah aplikasi untuk memberikan pembebanan pada sisi *client* dan *server*. Untuk menjalankan iperf diperlukan 1 host untuk dijadikan *server* dan 1 *host* untuk dijadikan *client*. Pada pengujian ini setiap topologi diberikan 10 *switch* dan 20 *host*. Pada 10 *host* pertama akan dijadikan *server* dan 10 *host* selanjutnya akan dijadikan *client*. Pada pengujian ini, jaringan akan diberi beban sebesar 200Mb untuk *bandwith* dan 50MB untuk melakukan pertukaran data pada *procokol* tcp/ip dengan menguji 10 kali pengujian. Setiap *server-client* iperf akan memberi beban sebesar

200MB sehingga secara bersamaan jaringan akan melakukan transfer data sebanyak 2Gb. Pada host 1-10 ketikkan perintah **iperf -s** untuk menjadikan host sebagai *server* dan pada host 11-19 ketikkan perintah **iperf -c <ip-server> -b 200M -t 1500** untuk mengaktifkan iperf client dengan *bandwidth* 100Mbit/ dan mengirim selama 1500 detik. Untuk *client* yang akan diambil datanya dengan wireshark ketikkan perintah **iperf -c <ip-server> --n 100M -b 100 -i 1** agar dapat mengirim data sebesar 200MB pada *bandwidth* 200Mbit/s.

3.1.1. Perbandingan QoS Pada 5 Topologi Jaringan

Setelah pengukuran QoS selesai pada topologi jaringan, maka tahap selanjutnya adalah dengan membandingkan diantara 5 topologi jaringan manakah yang lebih baik.

Tabel 2. Perbandingan QoS TCP SDN Pada 5 Topologi

Topologi	Delay	Packet Loss	Throughput	Jitter
Star	1.534 ms	0.017 %	49.558 Mbit/s	0.90×10^{-6}
Tree	2.021 ms	0.014 %	39.396 Mbit/s	1.19×10^{-6}
Linear	2.880 ms	0.070 %	31.984 Mbit/s	2.21×10^{-6}
Fullmesh	2.653 ms	0.167 %	38.147 Mbit/s	6.03×10^{-7}
Ring	1.561 ms	0 %	49.745 Mbit/s	1.07×10^{-6}

Berdasarkan tabel 3.1. diatas. Nilai *delay* yang paling bagus pada *protocol* TCP adalah nilai yang paling rendah. Jika melihat pada tabel diatas nilai *delay* yang paling bagus adalah topologi *star* karena memiliki nilai yang paling rendah diantara 5 topologi. Nilai *packet loss* yang paling bagus pada *protocol* TCP adalah nilai yang paling rendah. Jika melihat pada tabel, nilai *packet loss* yang paling bagus adalah topologi *ring* karena memiliki nilai yang paling rendah diantara 5 topologi. Nilai *throughput* yang paling bagus pada *protocol* TCP memiliki nilai yang paling besar, jika melihat pada tabel perbandingan diatas, nilai *throughput* terbaik adalah topologi *ring* dengan nilai paling besar diantara 5 topologi. Nilai *jitter* yang paling bagus pada *protocol* TCP adalah nilai yang paling rendah jika melihat pada tabel diatas nilai *jitter* yang paling bagus adalah topologi *star* karena memiliki nilai yang paling rendah.

Tabel 3. Perbandingan QoS UDP SDN Pada 5 Topologi

Topologi	Delay	Packet Loss	Throughput	Jitter
Star	0.232 ms	0 %	48.976 Mbit/s	86.7×10^{-5}
Tree	0,233 ms	0 %	49.008 Mbit/s	3.50×10^{-5}
Linear	0.244 ms	0 %	44.525 Mbit/s	4.44×10^{-5}
Fullmesh	0.226 ms	0 %	48.674 Mbit/s	0.84×10^{-5}
Ring	0.224 ms	0 %	50.388 Mbit/s	$8,04 \times 10^{-5}$

Berdasarkan pada tabel 3.2. diatas. Nilai *delay* yang paling bagus pada *protocol* UDP adalah topologi *ring* karena memiliki nilai *delay* paling rendah diantara 5 topologi. Nilai *packet loss* pada *protocol* UDP memiliki tidak ada yang jelek karena memiliki nilai 0 % pada 5 topologi jaringan. Nilai *throughput* yang paling bagus pada *protocol* UDP adalah topologi *ring* karena memiliki nilai yang paling tinggi diantara 5 topologi. Nilai *jitter* yang paling bagus pada *protocol* UDP adalah topologi *fullmesh* karena nilai yang dihasilkan paling rendah.

3.1.2. Hasil analisis perbandingan QoS TCP SDN dan perbandingan QoS UDP SDN pada 5 Topologi.

3.1.2.1. Topologi Star

Topologi *Star* mendapatkan nilai *delay* sebesar 1.534 *ms* pada TCP sedangkan pada *protocol* UDP nilai *delay* yang didapatkan sebesar 0.232 *ms*. Nilai *packet loss* yang didapatkan pada topologi *star* sebesar 0.017 % pada *protocol* TCP sedangkan pada *protocol* UDP mendapatkan nilai sebesar 0%. Nilai *throughput* pada topologi *star* memiliki nilai sebesar 49.558 Mbit/s pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 48.976 Mbit/s. Nilai *jitter* yang didapatkan pada topologi *star* memiliki nilai sebesar 0.90×10^{-6} *ms* pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 86.7×10^{-5} *ms*.

3.1.2.2. Topologi Tree

Topologi *Tree* mendapatkan nilai *delay* sebesar 2.021 *ms* pada TCP sedangkan pada *protocol* UDP nilai *delay* yang didapatkan sebesar 0,233 *ms*. Nilai *packet loss* yang didapatkan pada topologi *tree* sebesar 0.014 % pada *protocol* TCP sedangkan pada *protocol* UDP mendapatkan nilai sebesar 0%. Nilai *throughput* pada topologi *tree* memiliki nilai sebesar 39.396 Mbit/s pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 49.008 Mbit/s. Nilai *jitter* yang didapatkan pada topologi *tree* memiliki nilai sebesar 1.19×10^{-6} *ms* pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 3.50×10^{-5} *ms*.

3.1.2.3. Topologi Linear

Topologi *Linear* mendapatkan nilai *delay* sebesar 2.880 *ms* pada TCP sedangkan pada *protocol* UDP nilai *delay* yang didapatkan sebesar 0.244 *ms*. Nilai *packet loss* yang didapatkan pada topologi *linear* sebesar 0.070 % pada *protocol* TCP sedangkan pada *protocol* UDP mendapatkan nilai sebesar 0%. Nilai *throughput* pada topologi *linear* memiliki nilai sebesar 31.984 Mbit/s pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 44.525 Mbit/s. Nilai *jitter* yang didapatkan pada topologi *linear* memiliki nilai sebesar 2.21×10^{-6} *ms* pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 4.44×10^{-5} *ms*.

3.1.2.4. Topologi Fullmesh

Topologi *Fullmesh* mendapatkan nilai *delay* sebesar 2.653 *ms* pada TCP sedangkan pada *protocol* UDP nilai *delay* yang didapatkan sebesar 0.226 *ms*. Nilai *packet loss* yang didapatkan pada topologi *fullmesh* sebesar 0.167 % pada *protocol* TCP sedangkan pada *protocol* UDP mendapatkan nilai sebesar 0%. Nilai *throughput* pada topologi *fullmesh* memiliki nilai sebesar 38.147 Mbit/s pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 48.674 Mbit/s. Nilai *jitter* yang didapatkan pada topologi *fullmesh* memiliki nilai sebesar 6.03×10^{-6} *ms* pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 0.84×10^{-5} *ms*.

3.1.2.5. Topologi Ring

Topologi *Ring* mendapatkan nilai *delay* sebesar 1.561 *ms* pada TCP sedangkan pada *protocol* UDP nilai *delay* yang didapatkan sebesar 0.224 *ms*. Nilai *packet loss* yang didapatkan pada topologi *ring* sebesar 0 % pada *protocol* TCP sedangkan pada *protocol* UDP mendapatkan nilai sebesar 0%. Nilai *throughput* pada topologi *ring* memiliki nilai sebesar 49.745 Mbit/s pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 50.388 Mbit/s. Nilai *jitter* yang didapatkan pada topologi *ring* memiliki nilai sebesar 1.07×10^{-6} *ms* pada *protocol* TCP sedangkan pada *protocol* UDP memiliki nilai sebesar 8.04×10^{-5} *ms*.

4. Kesimpulan

Berdasarkan hasil perbandingan QoS pada 5 topologi didapatkan hasil untuk nilai yang paling bagus pada *protocol* TCP adalah topologi *star*. Untuk nilai *delay* pada *protocol* UDP topologi yang paling bagus adalah topologi *ring*. Pada nilai *packet loss* yang paling bagus untuk *protocol* TCP adalah topologi *ring* sedangkan untuk nilai *packet loss* pada *protocol* UDP semua topologi memiliki nilai yang sama bagusnya. Untuk nilai *throughput* pada *protocol* TCP adalah topologi *ring* sedangkan untuk *protocol* UDP adalah topologi *ring*. Pada nilai *jitter* yang paling bagus untuk *protocol* TCP adalah topologi *star* sedangkan untuk *protocol* UDP yang paling bagus adalah topologi *fullmesh*. Berdasarkan perbandingan di atas topologi yang paling bagus secara keseluruhan adalah topologi *ring* dan topologi yang paling jelek secara nilai keseluruhan adalah topologi *linear*.

Ucapan Terima Kasih

Penelitian ini dapat dilaksanakan dengan baik berkat bantuan dari berbagai pihak, untuk itu penulis mengucapkan terimakasih kepada Prodi Teknik Elektro, Fakultas Teknik, Universitas Garut, dan pihak lain yang telah memberikan kerjasama yang baik dalam penelitian ini.

DAFTAR PUSTAKA

- [1] I. M. S. Abhimata Zuhra Pramudita, *Perbandingan Performa Controller OpenDayLight dan Ryu pada* , vol. 1, p. 4, 2020.
- [2] R. T. Y. S. H. Roni Fernando Simarmata, *SIMULASI JARINGAN SOFTWARE DEFINED NETWORK MENGGUNAKAN PROTOKOL* , Vols. Vol.4, No.3, p. 2887, 2018.
- [3] T. T. M. SDN, "SOFTWARE DEFINED NETWORK," *INSTALASI SOFTWARE PENDUKUNG SDN*, p. 1, 2016.
- [4] S. N. H. S. N. H. Abu Riza Sudiyatmoko, *Analisis Performansi Perutingan Link State Menggunakan* , vol. Vol. 8 No.1, pp. 40, 41, 2016.

- [5] E. Wahyudi, "uji firewall dan routing pada SDN Openflow BAB 2," [Online]. Available: https://www.academia.edu/10827293/uji_firewall_dan_routing_pada_SDN_Openflow_BAB_2. [Accessed senin oktober 2021].
- [6] rizkynurwin, "tentang software defined networking sdn," senin Agustus 2017. [Online]. Available: <https://rizkynurwin1926.wordpress.com/2017/08/21/tentang-software-defined-networking-sdn/>. [Accessed senin oktober 2021].
- [7] T. R. Hapsari, "core network laboratory tech age berkenalan dengan openflow," jumat september 2018. [Online]. Available: <https://medium.com/core-network-laboratory-tech-page/berkenalan-dengan-openflow-3caca9194e51>. [Accessed senin oktober 2021].
- [8] Melisa, "UNIKOM Melissa Bab 2," [Online]. Available: https://elibrary.unikom.ac.id/id/eprint/587/8/UNIKOM_Melissa_Bab%20II.pdf. [Accessed senin oktober 2021].