Telecommunications, Networks, Electronics, and Computer Technologies

Vol.1 (2), 2021, 77-84

JURNAL UPI

TELNECT

Journal homepage: http://ejournal.upi.edu/index.php/TELNECT/

Simulasi Multi-topologi Jaringan Berbasis SDN dengan Controller POX

Sadam Fauzi [1], Sifa Larasati[2], Adhwa Alifia Putri[3], Nissa Restyasari[4], Galura Muhammad Suranegara[5]*

[1,2,3,4,5] Program Studi Sistem Telekomunikasi, Universitas Pendidikan Indonesia, Bandung, Indonesia

*Corresponding Author: E-mail: sadamfauzi@upi.edu

ABSTRACTS

Software Define Networking (SDN) adalah salah satu perkembangan teknologi di bidang jaringan yang memisahkan control plane dengan data plane. SDN dapat diterapkan diberbagai topologi. Penelitian ini bertujuan untuk mengetahui dan menganalisis performa berbagai topologi jaringan berbasis SDN dengan Controller POX**.** Penelitian ini mengunakan metode Research and Development (R&D), yang diujikan dengan skala terbatas. Parameter pengujian yang dilakukan adalah delay, dan jitter. Dari hasil penelitian didapatkan hasil bahwa topologi linear, dan star memiliki performa yang sama.Akan tetapi topologi linear menjadi model topologi yang ideal dan lebih efisien karena memiliki jitter yang lebih kecil atau rendah dari semua topologi yang ada dari data penelitian ini. Diharapkan dari hasil penelitian ini bisa menjadikan gambaran bagi pembaca ataupun teknisi dalam memilih topologi saat membangun jaringan yang akan d rancang

ARTICLE INFO

Article History:
Received 15 Oktober 2021
Revised 22 November 2021
Accepted 11 Desember 2021
Available online 15 Desember 2021

Keyword:
Software Defined Network, Mininet, POX, Topologi

1. INTRODUCTION

Perkembangan jaringan komputer terus meningkat untuk memenuhi kebutuhan akan teknologi perangkat jaringan di berbagai perusahaan dan menciptakan jaringan yang handal, salah satunya yaitu Software Defined Network atau yang biasa dikenal dengan singkatan SDN. SDN merupakan suatu konsep pendekatan baru yang memisahkan control plane dan data plane dengan tujuan untuk merancang, membangun dan mengelola jaringan computer [1]. SDN mampu mengelola seluruh jaringan secara efisien dan mengubah arsitektur jaringan yang kompleks menjadi sederhana dan mudah dikelola [2]. SDN menggunakan interface yang disebut dengan controller untuk memusatkan kendali jaringan dalam control plane. Umumnya, controller menggunakan OpenFlow (NOX, POX, Beacon, Floodlight, MuL, Maestro, Ryu). Masing-masing controller memiliki perbedaan serta kelebihan dan kekurangan yang berpengaruh pada performansi jaringan [3].

Multi-Topologi merupakan kumpulan dari beberapa topologi lebih dari satu topologi yang di istilahkan dalam penelitian ini terdiri dari topologi linear, tree, star, fullmesh, dan ring. Tujuan dari penelitian ini adalah menganalisis performa dari setiap topologi. Seperti pada penelitian sebelumnya yang telah dilakukan mengenai simulasi kinerja berbagai topologi jaringan berbasis SDN dengan controller RYU didapatkan hasil bahwa topologi yang berjalan pada controller RYU menghasilkan topologi linear yang terbaik dan efisien [4].. Namun pada penelitian tersebut tidak menggunakan controller POX hal tersebut dasar dari penelitian ini.

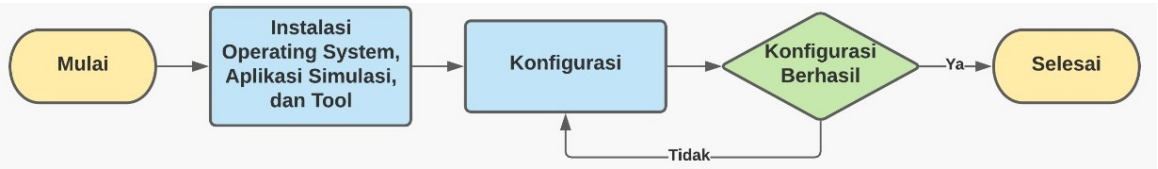
Penelitian menggunakan controller POX dalam menganalisa performa multi-Topologi tersebut. Controller POX merupakan controller berbasis python, dan dapat dijalankan pada sistem operasi Windows, MacOS dan Linux. POX masih dalam tahap pengembangan, dan merupakan perkembangan dari controller terdahulu yaitu NOX (berbasis bahasa C).

Simulasi yang kami jalankan ialah menggunakan mininet exterm dengan trafic iperf, dengan cara melakukan ping xterm yang dijalankan diatas controller POX itu sendiri. Dalam jaringan SDN, controller harus diaktifkan dahulu sebelum menjalankan jaringan. POX yang merupakan controller open source untuk pengembangan aplikasi pada SDN ini menyediakan cara efisien dalam menerapkan protocol OpenFlow. Mininet yaitu emulator jaringan opensource yang digunakan untuk penelitian SDN. Sentralisasi jaringan dengan semua pengaturan berada pada control plane adalah konsep utama pada SDN [4].

Hasil yang didapatkan pada penelitian ini bahwa ada 3 topologi yang menghasilkan efisiensi pengiriman paket terbaik yaitu topologi linear, tree, dan tree. Hal tersebut terjadi karena pengiriman paket yang terjadi lebih baik karena memiliki delay yang cukup rendah.

2. MATERIALS AND METHODS

Berikut merupakan langkah yang dilakukan dalam simulasi ini, dapat dilihat pada gambar berikut.

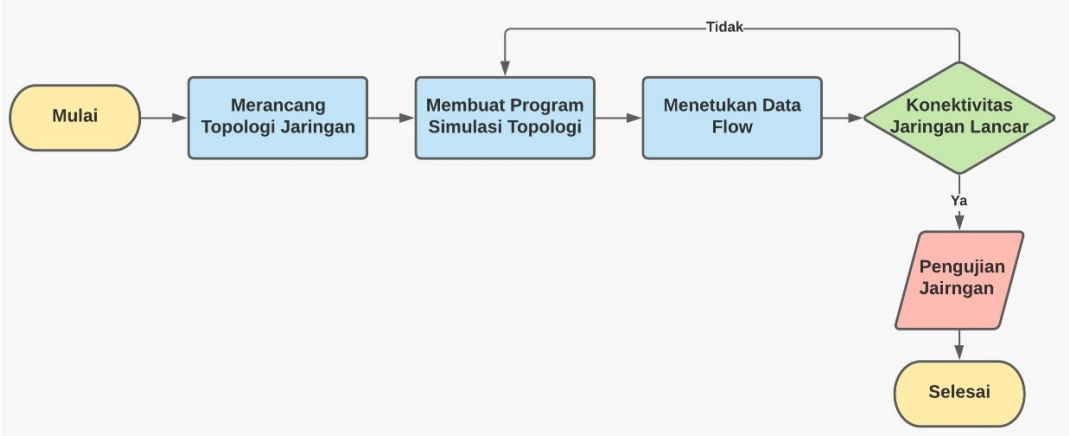


Gambar 1. Perancangan Sistem

Berikut merupakan langkah-langkah dari perancangan sistem:

1. Langkah pertama yang dilakukan adalah instalasi operating system yang mendukung penelitian ini yaitu menggunakan.
2. Konfigurasi operating system dan tool aplikasi yang digunakan. Konfigurasi mininet yang sudah terinstal. Konfigurasi telah berhasil jika ketika server mininet menjalankan suatu topologi jaringan dengan beberapa host, maka pada server terlihat MAC Address di switch yang telah dijalankan.
3. Jika konfigurasi tidak berhasil maka perlu dilakukannya pengecekan pada instalasi dan konfigurasi yang sebelumnya dilakukan, jika terdapat kesalahan maka perlu dilakukannya perbaikan pada konfigurasi tersebut. Jika konfigurasi pada tool dan aplikasi telah berhasil maka dapat melakukan tahap selanjutnya.
4. Penentuan skenario pada tahap pengujian jaringan yaitu jaringan SDN.
5. Pengujian jaringan SDN dan juga pembahasan hasil pengujian yang telah sukses dilakukan.

Tahap selanjutnya adalah pengujian perfoma dari jaringan virtual SDN yang telah di rancang dan dibangun. Pada penelitian ini akan menggunakan berbagai jenis topologi yang berbeda. Berikut merupakan langkah-langkah dari pengujian sistem.



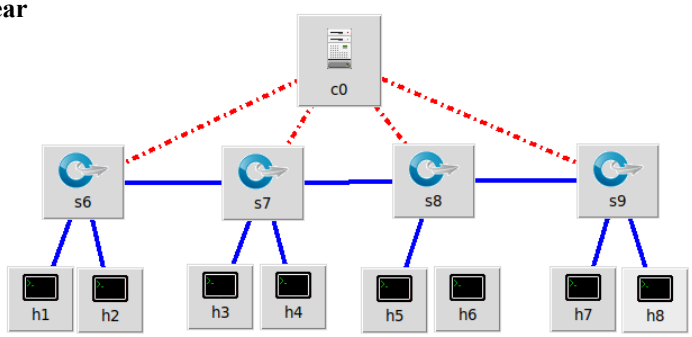
Gambar 2. Pengujian Multi-Topologi

Berikut merupakan penjelasan dari pengujian sistem, pada gambar 2:

1. Langkah pertama adalah menentukan jenis jaringan, pada penelitian ini menggunakan beberapa jenis topologi jaringan. Perancangan jaringan topologi ini menggunakan tools miniedit. Miniedit merupakan user interface grafis yang digunakan untuk merancang skenario dari jaringan SDN. Gambar dari perancangan berada pada setelah penjelasan langkah dari pengujian sistem ini.
2. Menentukan jumlah node. Controller yang digunakan yaitu hanya controller POX, sementara jumlah host dan switch memiliki jumlah lebih dari satu pada setiap jenis topologi jaringan. Dan jumlah dari host dan switch disesuaikan dengan setiap topologi.
3. Menentukan bentuk topologi, pada penelitian ini adalah topologi linear, topologi tree, topologi star, topologi full mesh, dan juga topologi full ring. Tools yang digunakan adalah mininet dan sublime text.
4. Membuat program menggunakan mininet sesuai dengan setiap topologi yang dibuat dan menggunakan pemrograman bahasa python pada controller pox.
5. Langkah selanjutnya adalah menentukan data flow. Data flow ini akan menentukan pergerakan paket data yang akan dikirim pada jaringan. Data flow harus sesuai dengan routing statis yang telah dibuat. Sehingga ketika pengiriman paket data nantinya tidak terjadi kesalahan dan konektivitas jaringan tetap terhubung.
6. Jika konektivitas terdapat masalah (error), maka semua konfigurasi pada host, switch ataupun server perlu diperiksa kembali. Untuk memeriksa konektivitas dapat menggunakan pingall. Jika ada error harus diperiksa kembali pembuatan topologi, pemrograman dan data flow pada jaringan SDN

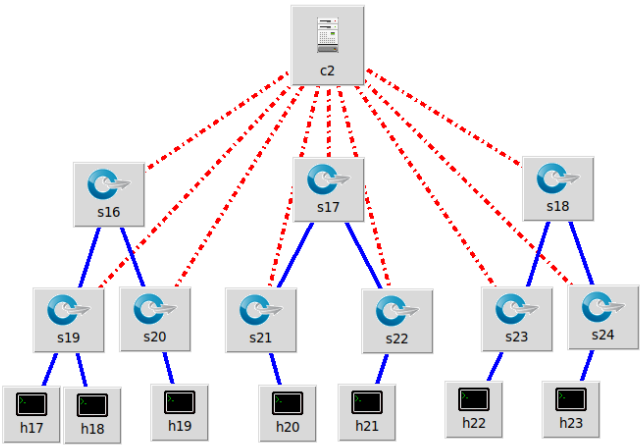
Topologi yang diujikan dari penelitian ini adalah sebagai berikut:

1. **Topologi Linear**



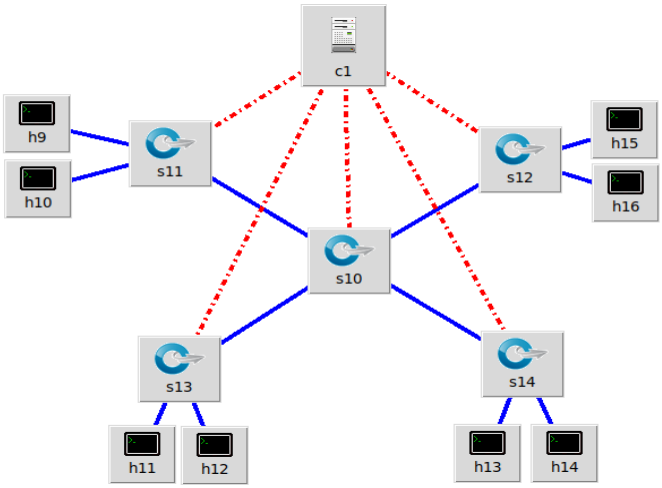
Gambar 3. Topologi Linear

2. Topologi Tree



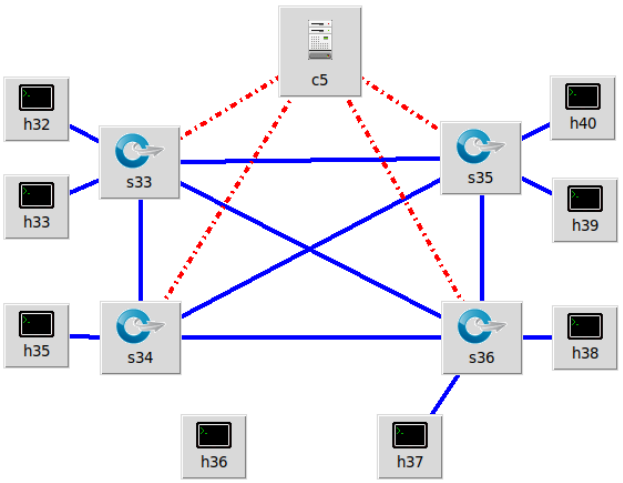
Gambar 4. Topologi Tree

3. Topologi Star



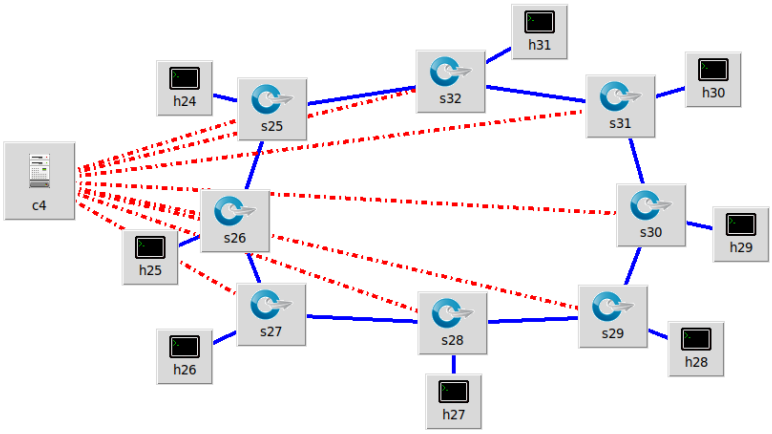
Gambar 5. Topologi Star

4. Topologi Full Mesh



Gambar 6. Topologi Fullmesh

5. Topologi Ring



Gambar 7. Topologi Ring

3. RESULTS AND DISCUSSION

Pengujian yang dilakukan menggunakan perangkat keras & perangkat lunak dengan spesifikasi berikut:

Tabel 1. Spesifikasi Hardware & Software

No	Spesifikasi		
	Hardware		Software
1	*Laptop*	*Processor AMD A6, 2.0 ghz*	*Ubuntu 18.0*
2			*Mininet 2.2.1*
3		*RAM 6gb*	*POX Controller*
4			*SecureRT 9.1*
5		*Quad Core x4*	*Openflow 1.5.1*
6			*Sublime Text 3.0*

Untuk mengetahui performa dari jaringan virtual SDN. Pengujian menggunakan parameter Qality of Service (QoS) yaitu delay, jitter, dan throughput, dii ujicoba sebanyak 5 kali.Dengan mengambil data rata-rata. Hasil dari pengukuran yang dilakuan didapatkan dari pengukuran langsung berasal dari uji coba topologi diatas menggunakan mininet Xterm, traffic iperf, menghasilkan data berikut.

Tabel 2. Perbandingan Pengukuran

Jenis Topologi	Delay (ms)	Jitter (ms)
Linear	*0,02351*	*0,00063*
Tree	*0,02351*	*0,00065*
Star	*0,02352*	*0,00063*
Full Mesh	*0,02490*	*0,00072*
Ring	*0,02598*	*0,00073*

3.1 Discussion

1. Delay

Delay merupakan waktu tunda dari suatu paket karena proses transmisi dari satu titik ke titik lainnya (titik tujuan) [6]. Satuan dari *delay* adalah ms. Rumus untuk menghitung *delay* adalah sebagai berikut:

Delay = Waktu penerimaan paket – Waktu pengiriman paket [7]

Berdasarkan rumus diatas dapat disimpulkan bahwa *delay* merupakan nilai selisih antara waktu penerimaan paket dengan waktu pengiriman paket.

Sementara itu, rumus rata-rata *delay* adalah sebagai berikut [7]:

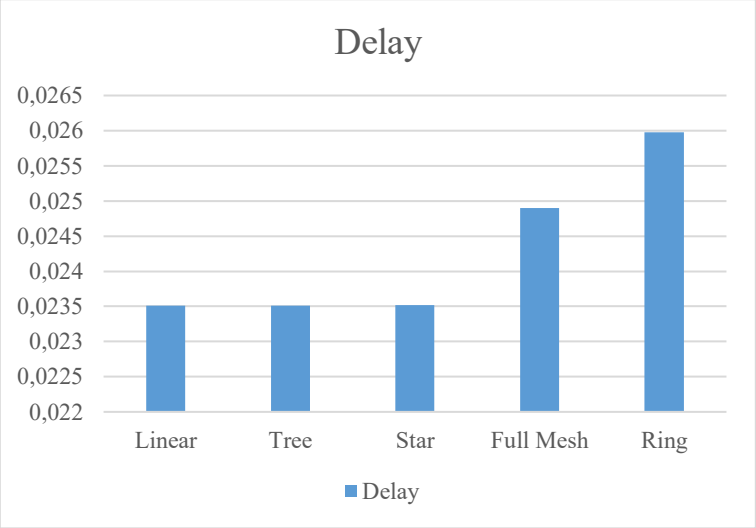
$$\boldsymbol{Rata - rata\ delay} = \frac{\boldsymbol{Total\ delay}}{\boldsymbol{Total\ paket\ yang\ diterima}}\ [7] \qquad (1)$$

Pada standar ITU-T G.1010 memiliki kategori delay seperti di Tabel 2 [5].

Tabel 3. Standar Delay menurut ITU-T G.1010.

Delay (ms)	Kualitas
< 150	Sangat Baik
150 - 300	Baik
300 - 450	Cukup

Dari tabbel diatas dapat dipermudah dalam membandingkan dengan cara membuat sebuah visual dengan gambaran grafik, berikut gambar 8 menunjukan data perbandingan delay yang diperoleh dari hasil pengambilan data, sebagai berikut



Gambar 8. Perbandingan Delay pada Topologi**.**

2. Jitter

Jitter adalah selisih dari nilai total *delay* dengan total paket atau data yang diterima. Semakin besar nilai dari total *delay*, maka nilai QoS akan semakin menurun. Qos (*Quality of Service*) merupakan kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwidth*, dan mengatasi *jitter* dan *delay* [6]. Satuan dari *jitter* adalah ms. Rumus untuk menghitung *jitter* adalah sebagai berikut [6]:

$$\boldsymbol{Jitter} = \frac{\boldsymbol{Total\ variasi\ delay}}{\boldsymbol{Total\ paket\ yang\ diterima\ -\ 1}}\ [8] \qquad (2)$$

Total variasi *delay* merupakan jumlah dari selisih tiap nilai *delay*. Rumus dari total variasi *delay* adalah sebagai berikut:

Total variasi delay = (delay2 – delay1) + (delay3 – delay2) + … + (delayN – delay(N-1)) [8]

Standar Jitter menurut TIPHON yaitu seperti pada tabel 4 [5,7]:

Tabel 4. Standar Jitter menurut TIPHON.

Jitter (ms)	Kualitas
0	Sangat Baik
0 – 75	Baik
76 – 125	Cukup
126 - 225	Buruk

Dari tabbel diatas dapat dipermudah dalam membandingkan dengan cara membuat sebuah visual dengan gambaran grafik, berikut gambar 9 menunjukan data perbandingan jitter yang diperoleh dari hasil pengambilan data, sebagai berikut:

Gambar 9. Perbandingan JItter Pada Topologi.

3. CONCLUSION

Simulasi topologi linear, tree, star, full mesh, dan ring menggunakan controller POX telah dilakukan. Melalui pengujian yang telah didapat, disimpulkan bahwa penggunaan POX sebagai controller SDN menghasilkan delay yang cenderung stabil dengan range delay pada beberapa topologi yaitu (0,02351 - 0,02598) ms, topologi ring yang memiliki delay paling besar. Pada hasil pengukuran jitter dari beberapa topologi cenderung stabil dan memiliki range (0,00073 - 0,00063) ms. Sehingga delay, jitter, dari semua topologi yang diuji cenderung stabil dan sangat baik karena sesuai standar yang dipakai. Dan didapatkan topologi terbaik dan efisien adalah topologi linear, tree, dan star pada penelitian ini karena memiliki tingkat rata-rata delay yang rendah, karena rendah itulah paket dapat dikirim secara cepat dan stabil.

4. REFERENCES

[1] N. A. Faruqi, L. Nurwadi, N. Ismail and D. Maryanto, "Simulasi Kinerja Berbagai Topologi Jaringan Berbasis Sotfware-Defined Network (SDN)," *Seminar Nasional Teknik Elektro,* pp. 232-233, 2017.

[2] R. M. Negara and R. Tulloh, "Analisis Simulasi Penerapan Algoritma OSPF Menggunakan RouteFlow pada Jaringan Sotfware Defined Network (SDN)," *Jurnal Infotel,* vol. 9, no. 1, pp. 75-76, 2017.

[3] I. A. V. M. D. I. R. R. M. B. M. Romi Afan, "ANALISIS EFEK PENGGUNAAN KONTROLER RYU DAN POX PADA PERFORMANSI JARINGAN SDN," *e-Proceeding of Engineering,* vol. 5, no. 3, pp. 6025-6026, 2018.

[4] N. A. Faruqi. dkk, “Simulasi Kinerja Berbagai Topologi Jaringan Berbasis *Software-Defined Network* (SDN).” Senter 2017, 15-16 Desember 2018, pp.232-239. ISBN: 978-602-512-810-3

[5] K. Masykuroh, A. D. Ramadhani, and N. Iryani, “ Analisis Qos dan Qoe Pada Video Pembelajaran Online di Institut Teknologi Telkom Purwokerto (ITTP),” *Jurnal Ilmiah Teknik Elektro,* vol. 23, no.2 pp. 40-47, 2021.

[6] G. F. E. Ardiansa, R. Primananda and M. H. Hanafi, "Manajemen Bandwidth dan Manajemen Pengguna pada Jaringan Wireless Mesh Nework dengan Mikrotik," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer ,* vol. 1, no. 11, pp. 1229-1230, 2017.

[7] S. R. Agrianto, P. D. Kusuma and R. R. M. , "Simulasi dan Analisis Kinerja QoS (Quality of Service) Jaringan Berbasis Simple Network Management Protocol (SNMP)," *e-Proceding of Engineering,* vol. 6, no. 1, pp. 1598-1599, 2019.

[8] P. E. Pratiwi, A. F. Isnawati and A. Hikmaturokhman, "Analisis QoS Pada Jaringan Multi Protocol Label Switching (MPLS) Studi Kasus di Pelabuhan Indonesia III Cabang Tanjung Intan Cilacap," *Akatel Sandhy Putra Purwokerto,* 2013.

[9] Y. I. Suryanto, S. and A. A. Zahra, "Analisis Kinerja Zigbee (802.15.4) WSN pada Topologi Tree dan Star Mode Non Beacon Menggunakan Network Simulator 2," *TRANSIENT,* vol. 4, no. 3, pp. 696-698, 2015.

[10] I. Ummah and D. Abdillah, "Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking," *Ind. Journal on Computing,* vol. 1, no. 1, pp. 95-97, 2016.