Performansi *Software Defined Network Controller* Pada *Streaming* Video Menggunakan *Real-time Transport Protocol*

http://dx.doi.org/10.28932/jutisi.v7i2.3644

Riwayat Artikel
Received: 31 Mei 2021 | *Final Revision*: 27 Juni 2021 | *Accepted*: 12 Juli 2021

Dodi Muhamad Kodar✉ #1, Rohmat Gunawan#2, Alam Rahmatulloh#3

Program Studi Informatika, Universitas Siliwangi
Jalan Siliwangi Nomor 24, Kota Tasikmalaya

[1]dodimuhamad7@gmail.com

[2]rohmatgunawan@unsil.ac.id

[3]alam@unsil.ac.id

***Abstract* — Computer network conditions that are dynamic and also complex make network operators often make mistakes in the configuration of a network device. However, these mistakes can be resolved by the emergence of a centralized programming approach which is now known as Software Defined Network (SDN). In addition, with the development of information technology, several services in computer networks also grabbed a lot of attention from the world, such as video streaming services. A video streaming service that is so massive is capable of having an impact on data traffic on computer networks. In SDN networks, controllers are a core component of this architecture. Currently, many different types of SDN controllers can be used, however, choosing a controller in an SDN network must be an important concern. This is related to the performance of the SDN controller which is a requirement for SDN network development. This research will analyze the performance of the video streaming service using the Real-time Transport Protocol (RTP) streaming protocol in testing two SDN controllers, namely Pox and Opendaylight. Based on the tests that have been carried out, it produces Quality of Service (QoS) values in the form of throughput, delay, and jitter. The Pox controller excels in the value of QoS delay and throughput. As for the jitter value, both Pox and Opendaylight controllers have good indexes.**

***Keywords*— Opendaylight; Pox; QoS; RTP; SDN**

I. PENDAHULUAN

Jaringan Komputer yang dibangun dalam skala besar tentu mempunyai tingkat kompleksitas yang rumit sehingga seorang administrator jaringan harus bisa mengkonfigurasi jaringan secara manual [1]. Tidak menutup kemungkinan kondisi jaringan yang dinamis juga kompleks membuat operator jaringan kerap melakukan kesalahan. Akan tetapi, kesalahan-kesalahan tersebut kini dapat sedikit diatasi dengan munculnya sebuah pendekatan program terpusat yang dikenal dengan *Software Defined Network (SDN*) [2] [3]. Pemisahan antara *control plane* dan *data plane* menjadi konsep perbedaan jaringan SDN dengan jaringan konvensional [4]. SDN dibangun untuk memudahkan memanajemen jaringan, meningkatkan ketersediaan jaringan, mengefektifkan biaya operasional jaringan, serta inovasi jaringan di masa depan [5].

Penelitian lanjutan mengenai pengembangan arsitektur SDN diperlukan untuk memastikan bahwa arsitektur SDN ini dapat diimplementasikan dengan baik dalam jaringan sesungguhnya, dikarenakan jaringan ini masih dibangun dalam emulator yang mendukung arsitektur SDN seperti mininet dengan kelebihannya bersifat *opensource* dan berjalan *dalam multi-platform*.

Komponen utama dari jaringan SDN adalah kontroler yang merupakan software yang bersifat fleksibel untuk dikonfigurasi sehingga administrator jaringan dapat mengontrol mekanisme jaringan dengan lebih mudah [6]. Kelayakan performa dari *controller* merupakan satu aspek penting dalam pemilihan sebuah *controller*. *Controller* yang digunakan dalam suatu jaringan harus dipilih dengan tepat agar tidak menjadi hambatan dalam pengembangan jaringan [7].

Aspek multimedia merupakan salah satu *SDN featured* yang saat ini menjadi tantangan dalam pengembangan SDN, salah satunya yaitu pemanfaatan multimedia streaming dalam jaringan komputer. Layanan *streaming* video saat ini telah

berkembang pesat dan menyita perhatian banyak orang di berbagai belahan dunia terutama dengan populernya layanan *streaming* seperti Youtube, Netflix, dan lain-lain. Oleh sebab itu layanan *streaming* yang begitu massif ini berdampak pada lalu lintas data pada jaringan Komputer [8]. Beberapa protokol yang biasanya digunakan dalam *streaming* video diantaranya: *Realtime Transport Protocol (RTP), Realtime Transport Streaming Protocol (RTSP),* dan *Hypertext Transfer Protocol (HTTP)* [9].

Realtime Transport Protocol (RTP) adalah salah satu *network protocol* berupa standarisasi paket untuk mengirimkan audio dan video pada jaringan komputer. RTP dirancang untuk aplikasi yang mengirimkan data secara *real time* dan menyediakan fungsi-fungsi transport jaringan *end to end* [10].

Pengujian performansi kontroler SDN dalam layanan *streaming* video menggunakan fitur *Realtime Transport Protocol (RTP)* merupakan fokus utama dari penelitian ini. *Controller* Pox dan Opendaylight dipilih untuk dijalankan dalam pengujian karena kedua kontroler ini compatible dijalankan dalam sistem operasi Linux. Beberapa skenario disiapkan untuk dikerjakan dalam percobaan, dan dilakukan pengukuran *Quality of Service (QoS*) dengan parameter: *delay, throughput, dan jitter*. Hasil dari setiap percobaan dicatat dalam tabel dan disajikan dalam grafik beserta penjelasan hasil analisis.

II. METODE PENELITIAN

Gambar 1. Metode Penelitian

Tahapan penelitian pada gambar 1 ini terdiri dari 4 (empat) tahap, yaitu: (1). studi literatur, (2). identifikasi kebutuhan, (3). implementasi dan pengujian, (4). dokumentasi.

A. Studi Literatur

Pada tahap ini dilakukan pencarian informasi dari paper publikasi, buku, web terkait *Software Defined Network (SDN), Mininet, Streaming Multimedia*, dan Kontroler pada SDN, serta observasi mengenai parameter QoS diantaranya *throughput, delay* dan *jitter*.

Pada penelitian Ahmad Heryanto, Afrilia [5] dilakukan percobaan terkait pengukuran QoS pada jaringan SDN yang dibangun menggunakan emulator Mininet. Hasil yang didapatkan berupa nilai QoS dari parameter pengujian *delay, jitter,* dan *packet loss*. Berdasarkan perhitungan QoS, nilai *delay* pada jaringan SDN lebih baik dibanding jaringan konvensional. Sedangkan nilai *jitter* menunjukkan jaringan berbasis SDN lebih stabil dibanding nilai *jitter* jaringan konvensional. Nilai *packet loss* pada jaringan konvensional lebih unggul dibanding jaringan SDN, hal ini karena konfigurasi kontroler yang terpusat, sehingga memerlukan waktu yang ideal untuk dapat menerapkan aturan yang telah dibuat pada perangkat jaringan dalam menerima konfigurasi.

Penelitian Moh Wahyudi Putra, Eko Sakti Pramukantoro, Widhi Yahya [7] dilakukan untuk mendapatkan nilai perbandingan dari tiap-tiap *controller* SDN berdasarkan parameter *throughput* dan *delay*. Percobaan pengujian *controller* dilakukan dengan skenario membuat topologi jaringan SDN dengan jumlah switch dan host yang berbeda. Hasil penelitian menunjukkan *controller* Maestro lebih unggul dibandingkan *controller* Floodlight, Pox, Ryu, dan Onos.

Penelitian Idris Z. Bholebawa1, Upena D. Dalal [11] berfokus pada pengujian dua buah *controller* jaringan SDN yakni Pox dan Floodlight *controller*. Pengujian *controller* dilakukan berdasarkan skenario jaringan yang berbeda. Skenario jaringan terdiri dari topologi single, linear, tree, dan custom topologi yang dibuat pada emulator Mininet. Berdasarkan parameter *throughput*, *controller* Floodlight lebih diunggulkan dan performanya efisien dibandingkan *controller* Pox.

Penelitian yang dilakukan C.Fancy, M.Pushpalatha [6] bertujuan untuk mengetahui fitur-fitur yang ada pada *controller* SDN. Pox dan Floodlight dipilih untuk dibandingkan berdasarkan parameter-parameter pengujian. Performa dari dua *controller* tersebut dinilai berdasarkan parameter *delay* dan *throughput*. Skenario pengujian jaringan SDN terdiri dari beberapa skenario yang bergantung pada jumlah host 10, 20, 30, 40, dan 50 host. Pada *controller* Floodlight waktu transmisi packet dan nilai *throughput* lebih baik dibanding *controller* Pox.

Penelitian Jehad Ali, Seungwoon Lee, Byeong-hee Roh [12] membandingkan performa dari dua *controller* jaringan SDN antara Ryu dan Pox *Controller*. Pengujian *controller* dilakukan berdasarkan skenario topologi yang berbeda: Single, Linear, Tree. Pada percobaannya juga dilakukan dengan penambahan skema jaringan Dumbbell, *Data Center Networks (DCN)* dan *Software- Defined naval networks.* Parameter pengujian *controller* ini terdiri dari *throughput* dan *delay*.

Dalam penelitian Abhimata Zuhra Pramudita, I Made Suartana [13], perbandingan kontroler dilakukan pada jaringan *Software Defined Network (SDN)* menggunakan emulator mininet. Pengujian kontroler dilakukan pada traffic besar dan

parameter pengujian berupa *throughput, delay, packet loss, memory* dan CPU. Hasil dari pengujian bergantung pada jumlah host dan switch. Hasil pengujiannya menunjukan Kontroler Ryu memiliki performa lebih baik dibanding kontroler OpenDayLight dalam parameter *throughput, delay*, dan *packet loss*.

Penelitian Riska, Hendri Alamsyah [9] bertujuan untuk mengetahui *Quality of Service (QoS)* dari simulasi jaringan SDN menggunakan kontroler ONOS. Pengujian dilakukan untuk mengetahui performance dari switch openFlow pada jaringan SDN. Pengujian QoS menggunakan tool iperf3 untuk traffic TCP dan UDP berdasarkan parameter *jitter, delay, bandwidth*, dan *throughput*.

B. Identifikasi Kebutuhan

Identifikasi kebutuhan diawali dengan persiapan perangkat simulasi jaringan SDN. Arsitektur SDN pada penelitian ini dibangun pada emulator Mininet yang dipasang pada sistem operasi Linux Ubuntu dalam virtual machine. Instalasi kontroler *Pox* dan *Opendaylight* dilakukan setelah Mininet dapat berjalan dengan baik.

C. Implementasi dan Pengujian

Implementasi topologi SDN dilakukan di dalam program Mininet dengan menggunakan *controller* yang berbeda yakni Pox dan Opendaylight. Proses streaming video dilakukan menggunakan software VLC dan dijalankan pada host jaringan SDN. Beberapa skenario streaming video dengan berbagai resolusi file video disiapkan. Capture packet data menggunakan aplikasi *Wireshark* dilakukan untuk pengukuran QoS dengan parameter *throughput, delay* dan *jitter* dengan.

D. Dokumentasi

Hasil pengujian QoS akan dinilai berdasarkan standar TIPHON *(Telecommunications and Internet Protocol Harmonization Over Network)*. Standar TIPHON digunakan karena merupakan standar penilaian QoS yang umum digunakan dan dikeluarkan oleh badan standar resmi *European Telecommunications Standards Institute (ETSI)*. Performa dua *controller* Pox dan *Opendaylight* saat melakukan video streaming dibandingkan menggunakan parameter Qos *throughput, delay* dan *jitter* berdasarkan hasil pengujian *streaming* video. Hasil perhitungan QoS dicatat dalam tabel serta disajikan dalam bentuk grafik.

III. HASIL DAN PEMBAHASAN

A. Persiapan Perangkat Simulasi Jaringan

Persiapan perangkat simulasi jaringan diawali dengan mempersiapkan software dan hardware sesuai spesifikasi yang dibutuhkan. Instalasi Linux Ubuntu pada Virtual Machine, dilanjutkan instalasi Mininet sebagai emulator SDN serta Pox dan Opendaylight sebagai controller dilakukan untuk mendukung simulasi arsitektur SDN. Hardware yang digunakan dalam percobaan yaitu Laptop dengan spesifikasi seperti ditampilkan pada tabel 1, sedangkan software yang digunakan ditampilkan pada tabel 2.

TABEL 1
SPESIFIKASI HARDWARE

No.	Komponen	Spesifikasi
1.	Processor	Intel Core i5 CPU @2.5 GHz
2.	Memory	8 GB DDR4
3.	Hardisk	1 TB
4.	Graphics	2GB GDDR3 VRAM NVIDIA GeForce 930MX
5.	Network Adapter	RealtekPCIeGbE Family Controller

TABEL 2
SPESIFIKASI SOFTWARE

No.	Komponen	Versi
1.	Mininet	2.3
2.	Pox Controller	Betta Branch
3.	Opendaylight Controller	Karaf 0.8.4
4.	VLC	2.2.2
5.	Wireshark	2.6.10

p-ISSN : 2443-2210
e-ISSN : 2443-2229

B. Instalasi Mininet, Pox, dan Opendaylight Controller

1) Mininet

Mininet merupakan sebuah emulator yang berfungsi untuk membuat prototype jaringan berskala besar secara cepat pada sumber daya yang terbatas. Perintah untuk instalasi emulator Mininet pada Linux Ubuntu ditampilkan pada Kode Program 1.

```
$ sudo apt install git
$ sudo mkdir -p /opt && cd /opt
$ git clone git://github.com/mininet/mininet
$ cd mininet
$ git tag
$ git checkout -b 2.2.2 $ util/install.sh -sfv
$ sudo mn
```

Kode Program 1. Perintah Instalasi Mininet

2) Pox Controller

POX merupakan platform open source yang berdasarkan pada bahasa pemrograman Python untuk aplikasi Software Defined Network (SDN) dan merupakan kontroler dari Openflow. POX memungkinkan proses perancangan dan pembangunan jaringan yang lebih cepat dalam jaringan Software Defined Network (SDN). Perintah untuk instalasi Pox Controller pada Linux ditampilkan pada Kode Program 2.

```
$ git clone http://github.com/noxrepo/pox
$ cd pox
$ ~/pox$ git checkout (version) (betta branch)
```

Kode Program 2. Perintah Instalasi Pox Controller

3) Opendaylight Controller

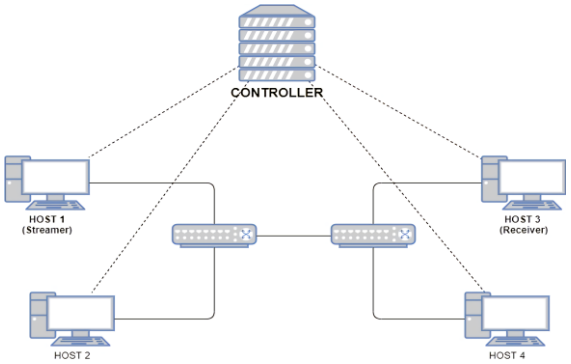
Controller OpenDaylight adalah sebuah proyek open source dengan Controller platform yang modular, *pluggable*, dan fleksibel. Perintah untuk instalasi Opendaylight kontroler pada Linux ditampilkan pada Kode Program 3.

```
$ wget https://nexus.opendaylight.org/content/repositories/
$ unzip opendaylight
$ sudo apt install java-jre
$ export JAVA_HOME=/usr/lib/jvm/jre-11
$ cd /root/opendaylight-0.12.1

$ ./bin/karaf
```

Kode Program 3. Perintah Instalasi Opendaylight Controller

C. Rancangan Topologi SDN



Gambar 2. Rancangan Topologi SDN

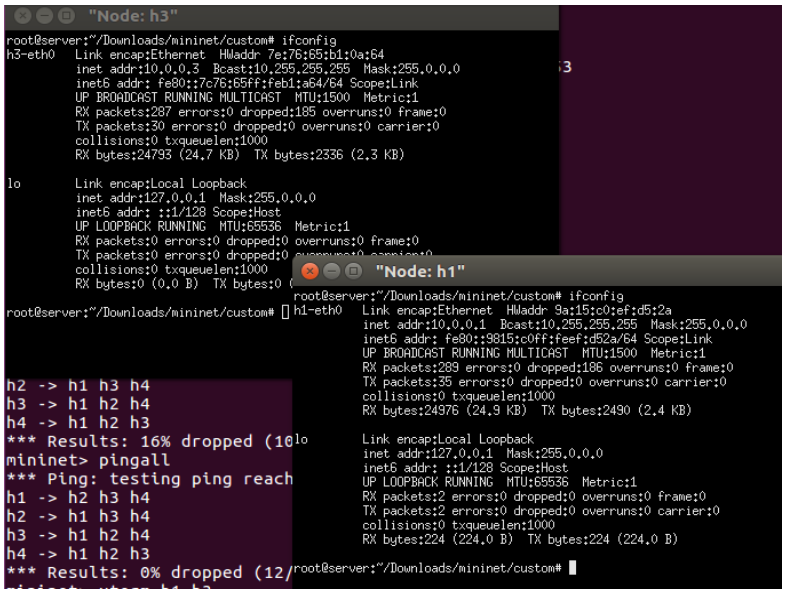
Implementasi topologi SDN dilakukan dengan program Mininet menggunakan kontroler yang berbeda yakni Pox dan Opendaylight. Layanan *streaming* video dilakukan melalui software VLC yang dijalankan pada host. Pengujian QoS jaringan dilakukan menggunakan parameter *throughput*, *delay* dan *jitter* dengan alat bantu Wireshark. Wireshark adalah aplikasi yang sering digunakan oleh administrator jaringan dalam menganalisis paket-paket data dalam suatu jaringan komputer.

Gambar 2 merupakan perancangan topologi jaringan SDN yang diimplementasikan menggunakan Mininet. Topologi jaringan SDN terdiri dari 4 (empat) host dan 2 (dua) switch). Setiap 2 (dua) host akan terhubung dengan 1 (satu) switch. Salah satu *host* akan bertugas sebagai pemberi layanan *streaming* (streamer) serta *host* yang lain akan diberikan tugas sebagai penerima layanan *streaming*.

D. Implementasi dan Pengujian Streaming Video

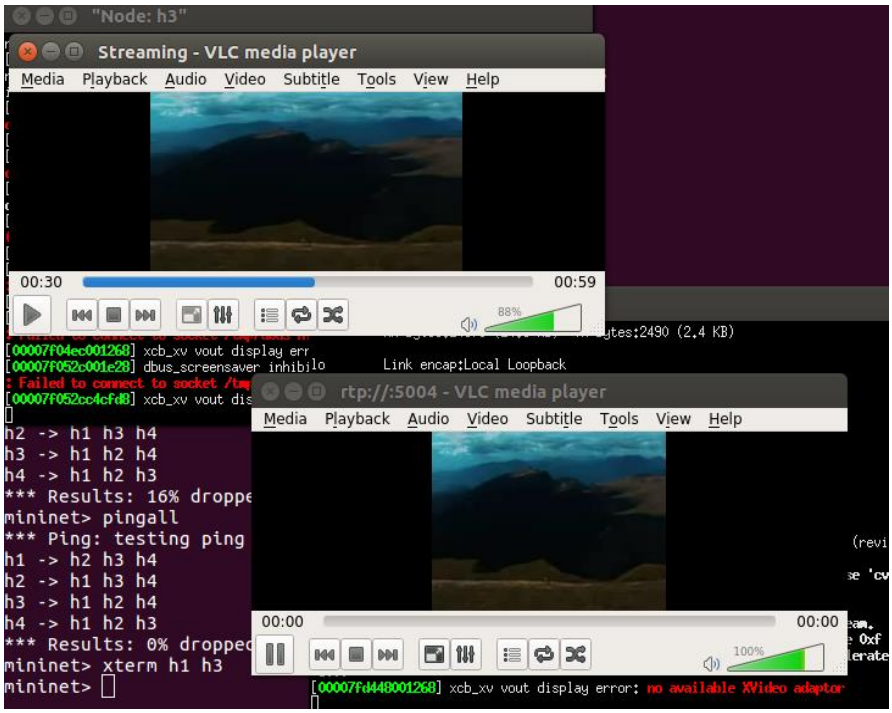
Skenario streaming video pada *Software Defined Network* (SDN) diawali dengan menjalankan controller Opendaylight dan Pox. Selanjutnya membuat topologi pada emulator Mininet dan dihubungkan dengan tiap controller. Topologi yang dibuat terdiri dari 2 switch dan 4 host. Setiap 1 switch terhubung dengan masing-masing 2 host.

Konfigurasi melalui terminal setiap host pada jaringan SDN dilakukan untuk menentukan server dan receiver. Host yang digunakan adalah host yang berfungsi sebagai server streaming video (streamer), dan host yang berfungsi sebagai receiver data streamer seperti ditampilkan pada gambar 3.



Gambar 3. Terminal Host Mininet

Pada gambar 3 ditampilkan terminal dari host yang akan dijadikan streamer dan receiver video melalui aplikasi VLC. Pada aplikasi VLC, fitur protocol stream yang dipakai adalah *Realtime Transport Protocol (RTP)* melalui host yang bertugas menjadi streamer. Tampilan proses *streaming* video seperti terlihat pada gambar 4.



Gambar 4. Proses Streaming Video

E. Pengujian Quality of Service (QoS)

Pengujian QoS pada jaringan SDN terhadap implementasi layanan multimedia *streaming* video dilakukan dengan *capturing packet* data menggunakan Wireshark dengan menerapkan *packet filtering* terhadap paket UDP. Beberapa skenario dengan berbagai resolusi video disiapkan. Video yang akan diuji pada setiap *controller* yaitu berformat mp4 dengan masing-masing resolusi video 144p, 240p, dan 480p dan durasi video 1 (satu) menit.

1) *Throughput*

Throughput adalah laju data aktual per satuan waktu atau biasa dikenal sebagai *bandwidth* dalam kondisi yang sebenarnya. Standar *throughput* dapat dilihat pada tabel 3.

TABEL 3
KRITERIA THROUGHPUT STANDAR TIPHON

Kategori Throughput	Throughput	Indeks
Bad	0 – 338 kbps	0
Poor	338 – 700 kbps	1
Fair	700 – 1200 kbps	2
Good	1200 kbps – 2,1 Mbps	3
Excellent	> 2,1 Mbps	4

2) *Delay*

Delay merupakan total waktu tunda suatu paket atau waktu yang dibutuhkan paket tersebut untuk menempuh jarak ke tujuannya yang diakibatkan oleh proses transmisi data dari satu titik ke titik lainnya. Standar *delay* dapat dilihat pada tabel 4.

TABEL 4
KRITERIA DELAY STANDAR TIPHON

Kategori Delay	Delay	Indeks
Poor	> 450 s	1
Medium	300 - 450 s	2
Good	150 - 300 s	3
Perfect	< 150 s	4

3) *Jitter*

Jitter merupakan variasi delay atau ukuran variabilitas yang terjadi akibat adanya selisih waktu atau interval antara *delay* pertama dan delay selanjutnya. Standar *jitter* dapat dilihat pada tabel 5.

TABEL 5
KRITERIA JITTER STANDAR TIPHON

Kategori Jitter	Jitter	Indeks
Poor	125 – 225 ms	1
Medium	75-125 ms	2
Good	0 – 75 ms	3
Perfect	1 ms	4

F. Dokumentasi Quality of Service (QoS) Berdasarkan Standar TIPHON

TABEL 6
EVALUASI PENILAIAN THROUGHPUT

Resolusi Video	Controller	Nilai	Indeks	Kategori
144p	ODL	40,3 Kbps	0	Bad
	Pox	30,9 Kbps	0	Bad
240p	ODL	50,8 Kbps	0	Bad
	Pox	40,4 Kbps	0	Bad
480p	ODL	125,8 Kbps	0	Bad
	Pox	120,8 Kbps	0	Bad

Tabel 6 menunjukkan bahwa *throughput* berdasarkan standar TIPHON pada tiga resolusi video (144p, 240p, 480p) dan dua *controller* (Opendaylight dan Pox) yang berbeda dikategorikan buruk.

TABEL 7
EVALUASI PENILAIAN DELAY

Resolusi Video	Controller	Nilai	Indeks	Kategori
144p	ODL	35,3 Ms	4	Good
	Pox	35,6 Ms	4	Good
240p	ODL	23,6 Ms	4	Good
	Pox	35,6 Ms	4	Good
480p	ODL	9,05 Ms	4	Good
	Pox	11,30 Ms	4	Good

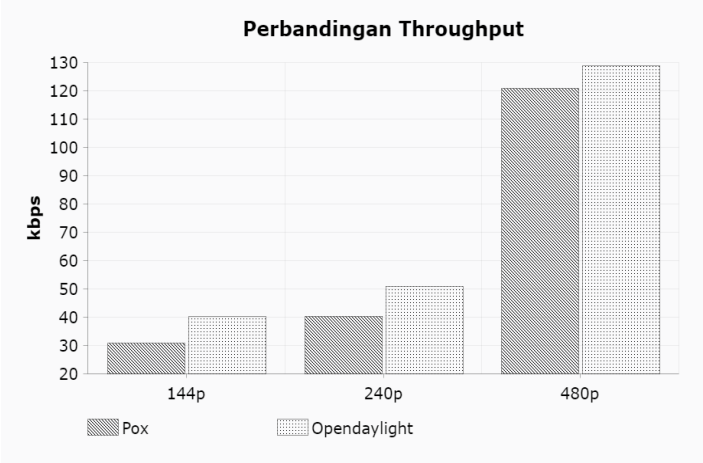
Tabel 7 menunjukkan bahwa *Delay* berdasarkan standar TIPHON pada tiga resolusi video (144p, 240p, 480p) dan dua *controller* (Opendaylight dan Pox) yang berbeda dikategorikan bagus.

TABEL 8
EVALUASI PENILAIAN JITTER

Resolusi Video	Controller	Nilai	Indeks	Kategori
144p	ODL	1,664 Ms	4	Good
	Pox	0,117 Ms	4	Good
240p	ODL	0,0256 Ms	4	Good
	Pox	-0,146 Ms	4	Good
480p	ODL	-1,169 Ms	4	Good
	Pox	-1,890 Ms	4	Good

Tabel 8 menunjukkan bahwa *Jitter* berdasarkan standar TIPHON pada tiga resolusi video (144p, 240p, 480p) dan dua *controller* (Opendaylight dan Pox) yang berbeda dikategorikan bagus.

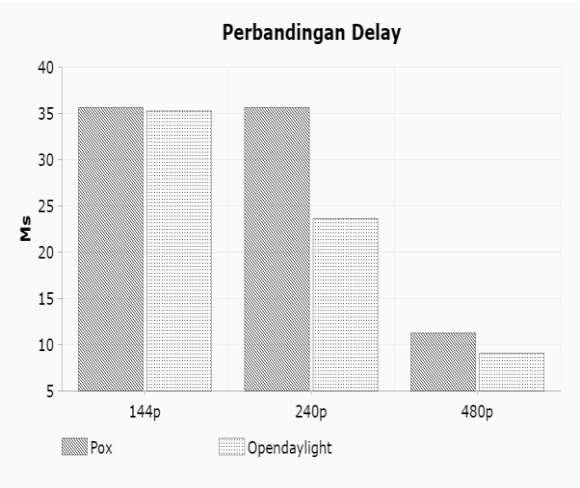
G. Perbandingan Nilai Quality of Service (QoS)



Gambar 5. Perbandingan Throughput

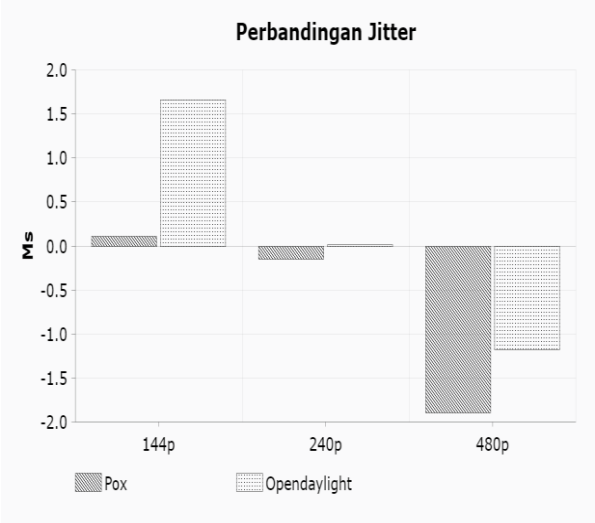
Gambar 5 menampilan nilai dari *throughput* dua *controller* yakni Opendaylight dan Pox pada setiap resolusi video yang berbeda. Data pada gambar ini menunjukkan nilai *throughput* dari *controller* Opendaylight pada resolusi video *streaming* 144p lebih tinggi 9,4 Kbps dibandingkan dengan Pox controller, lalu pada resolusi 240p nilai *throughput controller* Opendaylight lebih tinggi 10,4 Kbps dibandingkan Pox *controller*, dan pada resolusi 480p nilai *throughput* Opendaylight *controller* lebih tinggi 5 Kbps dibandingkan Pox *controller*.

Gambar 6. merupakan gambaran nilai dari *delay* dua *controller* yakni Opendaylight dan Pox pada setiap resolusi video yang berbeda. Data pada gambar ini menunjukkan nilai *delay* dari *controller* Opendaylight pada resolusi video *streaming* 144p lebih rendah 0,3 Ms dibandingkan dengan Pox *controller*, lalu pada resolusi 240p nilai *delay controller* Opendaylight lebih rendah 12 Ms dibandingkan Pox *controller*, dan pada resolusi 480p nilai *delay controller* Opendaylight lebih rendah 2,25 dibandingkan Pox *controller*.



Gambar 6. Perbandingan Delay

Gambar 7 merupakan gambaran nilai dari Jitter dua controller yakni Opendaylight dan Pox pada setiap resolusi video yang berbeda. Data pada gambar ini menunjukkan nilai jitter dari controller Opendaylight pada resolusi video streaming 144p lebih tinggi 1,54 Ms dibandingkan dengan Pox controller, lalu pada resolusi 240p nilai delay controller Opendaylight lebih tinggi 0,17 Ms dibandingkan Pox controller, dan pada resolusi 480p nilai delay controller Opendaylight lebih tinggi -0,72 dibandingkan Pox controller. Nilai jitter yang negatif membuat delay waktu yang dibutuhkan lebih sedikit akibat adanya gangguan paket atau dengan kata lain bahwa paket tiba lebih awal dari yang seharusnya.



Gambar 7. Perbandingan Jitter

IV. SIMPULAN

Berdasarkan hasil percobaan pada penelitian, *streaming* video menggunakan *Realtime Transport Protocol (RTP)* pada jaringan *Software Defined Network (SDN)* dapat berjalan baik. Host jaringan SDN yang berperan sebagai *streamer* mampu memberikan layanan *streaming* kepada host tujuan. Nilai *Throughput* dan *delay* pada *Controller* Opendaylight lebih unggul dibandingkan dengan *Controller* Pox. Namun untuk parameter jitter, *controller* Opendaylight dan Pox tidak menampilkan perbedaan yang signifikan. Pengembangan skema jaringan dengan menambahkan beberapa router, penggunaan routing protocol dan pemilihan protokol streaming yang berbeda merupakan beberapa tantangan yang dapat dilakukan pada penelitian berikutnya.

DAFTAR PUSTAKA

[1] I. P. A. E. Pratama and I. M. A. Wikantyasa, "Implementasi dan Analisis Simulasi QOS dan Perfomance Device dengan Menggunakan ONOS dan Iperf3," *Jurnal Informatika Universitas Pamulang*, vol. 4, no. 2, p. 57, 2019. DOI: 10.32493/informatika.v4i2.2730

[2] H. E. Putra and S. I. Lestariningati, "Penerapan Arsitektur Software-Defined," *Penerapan Arsitektur Software-Defined Networking Berbasis Openflow Pada Simulasi Jaringan Virtual*, vol. 7, no. 1, pp. 2–7, 2018.

[3] I. Ummah, "Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking," *Indonesian Journal on Computing (Indo-JC)*, vol. 1, no. 1, pp. 95–106, 2016. DOI: 10.21108/indojc.2016.1.1.20

[4] E. P. Aprilianingsih, R. Primananda, and A. Suharsono, "Analisis Fail Path Pada Arsitektur Software Defined Network Menggunakan Dijkstra Algorithm," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, vol. 1, no. 3, pp. 174–183, 2017.

[5] A. Heryanto and Afrilia, "Software Defined Network Menggunakan Simulator," no. 33, pp. 5–8, 2016.

[6] C. Fancy and M. Pushpalatha, "Performance evaluation of SDN controllers POX and floodlight in mininet emulation environment," *Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS 2017*, no. Iciss, pp. 695–699, 2018. DOI: 10.1109/ISS1.2017.8389262

[7] M. W. Putra, E. S. Pramukantoro, and W. Yahya, "Analisis Perbandingan Performansi Kontroller Floodlight , Maestro , RYU , POX Dan ONOS Dalam Arsitektur Software Defined Network (SDN)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 10, pp. 3779–3787, 2018.

[8] A. Kumar and Yash, "Performance Evaluation of Video Streaming Traffic in Data Centre Servers Using Real- Time Transport Protocol (RTP)," vol. 6, no. 08, pp. 472–477, 2020.

[9] R. Riska and H. Alamsyah, "Analisis Perbandingan Protokol Transport Pada Video Streaming di Jaringan Lokal (LAN) Menggunakan Videolan Client," *Jurnal Media Informatika Budidarma*, vol. 3, no. 2, p. 126, 2019. DOI: 10.30865/mib.v3i2.1110

[10] M. Mohamed and A. B. Abdelnabi, "Quality Of Service For Real Time Video Transferring In Wireless Wimax Technology," vol. 6, no. 10, pp. 74–82, 2017.

[11] I. Z. Bholebawa and U. D. Dalal, "Performance analysis of SDN/openflow controllers: POX versus floodlight," *Wireless Personal Communications*, vol. 98, no. 2, pp. 1679–1699, 2018. DOI: 10.1007/s11277-017-4939-z

[12] J. Ali, S. Lee, and B. H. Roh, "Performance analysis of POX and Ryu with different SDN topologies," *ACM International Conference Proceeding Series*, pp. 244–249, 2018. DOI: 10.1145/3209914.3209931

[13] A. Z. Pramudita and I. M. Suartana, "Perbandingan Performa Controller OpenDayLight dan Ryu pada Arsitektur Software Defined Network," *JINACS (Journal of Informatics and Computer Science)*, vol. 01, no. 4, pp. 174–178, 2020.

[14] N. Aulia and I. Nurcahyani, "Perancangan FTTH Menggunakan Ethernet Passive Optical Network (EPON) Pada Layer Network di Kampus Universitas Islam Indonesia," 2017.