JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 12 No. 1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v12i1.3766

PENGARUH PENGGUNAAN VLAN *PADA SOFTWARE DEFINED NETWORK* BERBASIS *RYU CONTROLLER*

M. Chairul Anam[*], Hery Dian Septama, Rio Ariestia P[3], Gigih Forda Nama[3]

[1,2,3] Teknik Informatika, Universitas Lampung

Riwayat artikel:

Received: 22 November 2022
Accepted: 29 Desember 2023
Published: 1 Januari 2024

Keywords:
Software Defined Network, VLAN, *Ryu Controller*

Corespondent Email:
m.chairulanam16@gmail.com

Abstrak. *Software Defined Network* adalah arsitektur jaringan yang memisahkan antara control plane dan data plane dalam perangkat jaringan yang membuat control plane menjadi independen dan dapat diprogram. Tugas *control plane* tersebut diambil controller dimana controller dapat mengatur semua aturan yang ada dalam seluruh perangkat jaringan, salah satu jenis controller tersebut adalah *ryu controller*. VLAN merupakan konfigurasi jaringan yang mendistribusikan beberapa segmen berbeda untuk perangkat yang terhubung pada jaringan dan dapat meningkatkan perfoma jaringan. Penelitian ini bertujuan untuk mengetahui bagaimana kinerja VLAN pada *Software Defined Network* yang menggunakan *Ryu Controller* dengan menggunakan metode simulasi dengan 2 skenario, yaitu topologi linear 3 OpenFlow Switch 30 host dan topologi linear 3 OpenFlow Switch 60 host yang diberi *traffic* 100, 200, 300, dan 400 Mb. Pengujian dilakukan menggunakan *app rest_router* pada *ryu controller* yang dapat menjalankan static routing tanpa VLAN dan dengan VLAN supaya terlihat perbedaan antara yang menggunakan VLAN dan tidak. Hasil pengujian menunjukkan bahwa topologi linear 3 switch baik 30 host atau 60 host dapat bekerja dengan baik. Berdasarkan pengujian parameter Qos *Throughput*, *Packet Loss*, dan *Jitter* keseluruhan VLAN kurang baik dibandingkan tanpa VLAN dengan nilai berturut-turut 222,9 Mbps, 0%, 0,0000636 s, dan 0,0000675 ms. Namun, seiring bertambahnya *host*, VLAN lebih dapat diandalkan.

Abstract. *Software Defined Network (SDN) is a network architecture that separates the control plane and data plane in networking devices, which then makes the control plane independent and programmable. The control plane task is taken by the controller, where the controller can manage all the rules in all networking devices, one type of controller is the Ryu controller. VLAN is a network configuration that distributes several different segments to devices connected to the network that can improve network performance. This study aims to determine the performance of VLAN on Software Defined Network using Ryu Controller using a simulation method with 2 scenarios, namely linear topology 3 OpenFlow Switch 30 hosts and linear topology 3 OpenFlow Switch 60 hosts with traffic of 100, 200, 300, and 400 Mb. Testing is carried out using the rest_router app on the Ryu controller which can run static routing without VLAN and with VLAN so that the difference between using VLAN and not can be seen. The test results show that the linear topology of 3 switches, whether 30 hosts or 60 hosts, can work well. Based on the test of QoS parameters Throughput, Packet Loss, and Jitter, the overall of VLAN is less good than without VLAN, with values of 222.9 Mbps, 0%, 0.0000636 s, and 0.0000675 ms, respectively. However, as the number of hosts increases, VLAN is more reliable.*

1. PENDAHULUAN

Dengan adanya peningkatan kebutuhan untuk jaringan komputer dan perangkat yang dibutuhkan dalam berbagai bidang, yang kita ketahui juga pertumbuhannya menjadi semakin meningkat pesat. Hal tersebut telah mempengaruhi kebutuhan akan kinerja jaringan yang lebih baik dari sebelumnya, dikarenakan tujuan utama dan juga penambahan konfigurasi jaringan yang menjadi semakin besar. Konfigurasi tersebut juga menjadi lebih kompleks dan melakukan kontrol dalam sebuah jaringan pun menjadi semakin sulit. Hal tersebut menyebabkan jaringan tidak fleksibel dan sulit untuk diatur [1].

Software Defined Network (SDN) merupakan sebuah konsep baru yang digunakan untuk melakukan perancangan, implementasi dan juga manajemen kebutuhan dalam jaringan komputer. Perangkat jaringan tradisional seperti switch dan router dapat dibagi menjadi 3 logical plane yang berbeda, yaitu data plane, *control plane*, dan *management plane*. *Data plane* tersebut megacu pada perangkat keras yang meneruskan paket, sedangkan *control plane* mengacu kepada bagian yang mengimplementasikan protokol dalam proses routing. Biasanya, dalam perangkat jaringan *control plane* akan diimplementasikan pada *proprietary firmware* yang dikembangkan oleh vendor perlengkapan. *Management plane* akan digunakan untuk mengawasi dan mengontrol fungsi [2].

Controller merupakan aplikasi yang mengatur *flow* yang terjadi pada jaringan SDN, yang dapat melakukan proses *intelligence networking*. *Controller* sendiri biasanya dijalankan pada protocol seperti *OpenFlow*. Kemudian, paket yang telah dikirimkan dapat diketahui tujuannya. Lalu lintas data yang terjadi dalam jaringan SDN seperti penerusan paket data, tindakan yang diperlukan untuk paket, dan bahkan mengirim aturan baru yang disesuaikan supaya dapat menangani paket data dikemudian hari juga diatur oleh controller [3]. Salah satu *controller* dengan sumber terbuka yang paling sering digunakan adalah *ryu controller*. Controller ryu juga memiliki built-in function yang sudah berada dalam *package* saat diinstal pada sistem operasi *linux* seperti *layer 2 switch*, *layer 3 switch*, *router*, dll. Pada penelitian ini akan dilakukan analisa kinerja VLAN pada *Ryu controller* dengan menggunakan simulasi dalam emulator mininet menggunakan topologi dari jaringan yang telah ditentukan untuk mengetahui pengaruh VLAN dalam penerapannya pada *Software Defined Network* dan kemudian performa tersebut akan dinilai berdasarkan standar TIPHON.

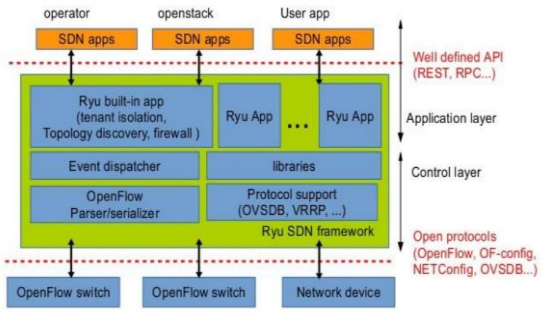
2. TINJAUAN PUSTAKA

2.1 *Software Defined Network*

Software Defined Network (SDN) merupakan sebuah konsep baru yang digunakan dalam merancang, implementasi dan juga manajemen kebutuhan dalam jaringan komputer. Perangkat jaringan tradisional seperti switch dan router dapat dibagi menjadi 3 logical plane yang berbeda, yaitu data plane, control plane, dan management plane. SDN adalah arsitektur jaringan yang sedang berkembang yang memisahkan antara data plane dan control plane dalam perangkat jaringan yang kemudian membuat control plane menjadi independent dan dapat diprogram [2].

2.2 *Ryu Controller*

Ryu merupakan salah satu komponen dasar dari *software defined network* (SDN). Ryu meneyediakan komponen perangkat lunak dengan API yang terdefinisi dengan baik yang mudah dikembangkan dalam membuat sebuah manajemen jaringan dan aplikasi untuk pengontrolan [4].

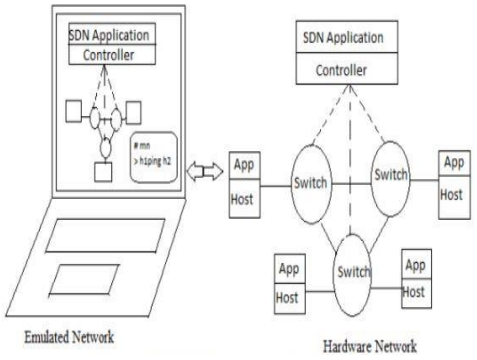


Gambar 1. Arsitektur *Ryu Controller*

2.3 Mininet

Mininet merupakan sebuah simulator jaringan yang dapat membuat jaringan dari virtual host, switch, controller, dan link. Mininet berjalan pada standar dari perangkat lunak jaringan linux, dan setiap switch mendukung untuk penggunaan openflow

dengan tingkat fleksibilitas yang tinggi untuk kustom routing dan *software defined network* (SDN) [5].



Gambar 2. Emulasi jaringan menggunakan mininet [6]

2.4 MiniNAM

MiniNAM merupakan tools berbasis GUI yang ditulis menggunakan Python Tkinter. MiniNAM menyediakan real-time animation dari jaringan apapun yang dibuat oleh emulator mininet. MiniNAM juga memiliki komponen yang yang diperlukan untuk memulai, memvisualisasikan, dan memodifikasi alur dari jaringan mininet secara real-time [7].

2.5 *Virtual Local Area Network (*VLAN**)**

VLAN adalah sebuah subnetwork yang dapat mengelompokkan kumpulan perangkat pada jaringan fisik yang terpisah. LAN merupakan sekelompok komputer dan perangkat yang saling berbagi jalur komunikasi melalui koneksi kabel maupun nirkabel dalam letak geografis yang sama. VLAN memudahkan network administrator untuk membagi jaringan untuk menyesuaikan dengan kebutuhan fungsional dan juga persyaratan keamanan pada sistem tanpa harus membuat kabel baru atau membuat perubahan besar terhadap infrastruktur jaringannya. VLAN sering digunakan oleh perusahaan besar untuk mempartisi ulang perangkatnya untuk manajemen lalu lintas data yang lebih baik [8].

2.6 Wireshark

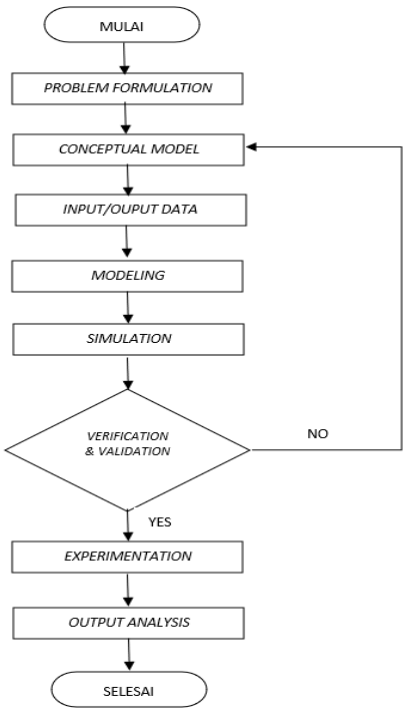
Wireshark merupakan salah satu aplikasi opensource yang dapat menganalisis paket data dalam jaringan. Perangkat ini juga digunakan untuk memecahkan permasalahan dalam jaringan, analisis, pengembangan protokol komuniaksi, dan edukasi. Dari banyaknya network analyzer yang digunakan oleh Network administrator, wireshark paling sering digunakan dalam proses analisis kinerja dari jaringan dan juga mengontrol lalu lintas data yang terjadi didalamnya dengan menangkap paket data yang lewat dan juga informasi dalam berbagai jenis protokol [9].

2.7 *Quality Of Service (*QOS*)*

Quality of service merupakan salah satu metode pendukung dalam sebuah penilaian ataupun evaluasi dalam menunjukkan kinerja dari suatu service yang didalamnya terdapat atribut kerja yang berfungsi sebagai dasar penilaian atau evaluasi service [10].

3. METODE PENELITIAN

Metode penelitian yang digunakan adalah dengan menggunakan simulasi [13]. Dalam menggunakan metode simulasi kita tidak perlu melakukan konfigurasi menggunakan perangkat riil, dengan menggunakan metode ini dapat dilakukan berbagai percobaan terlebih dahulu sebelum diterapkan secara langsung pada perangkat riil dalam sebuah jaringan



Gambar 3. Metode Penelitian

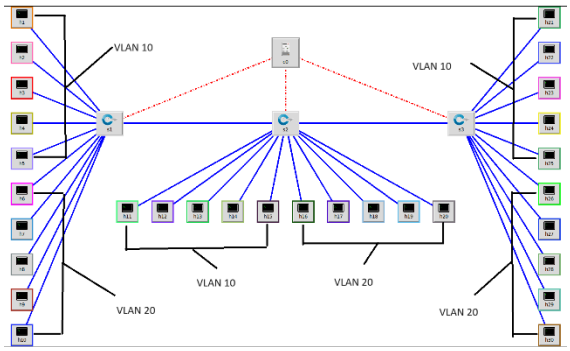
3.1 *Problem formulation*

Tahap ini merupakan tahapan dimana proses pencarian masalah dilakukan dan

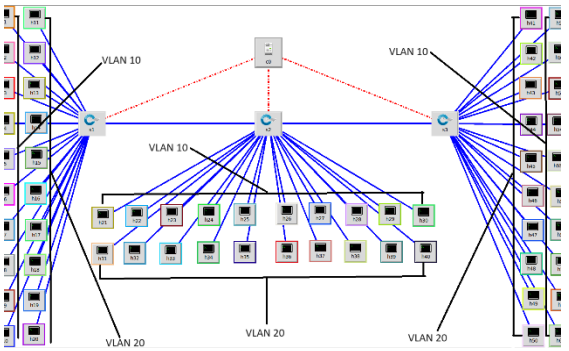
referensi-referensi sebagai landasan teori, metode, dan analisis yang akan dilakukan.

3.2 *Conceptual model*

Tahap ini adalah penggambaran konsep model simulasi virtual, sebelum melakukan proses simulasi terlebih dahulu melakukan pemodelan konseptual dengan merancang topologi yang akan digunakan pada penelitian ini beserta alamat IP dari masing- masing perangkat yang ada pada tiap topologi.



Gambar 4. Topologi 30 host



Gambar 5. Topologi 60 host

3.3 *Analysis of input/ouput data*

Pada tahap ini dilakukan penjabaran input dan ouput apa yang akan digunakan dalam penelitian, lalu didapatkan input dan output berikut:

a. Input

1. *Node*

Node adalah perangkat yang berada pada topologi yang dibuat dalam jaringan *Software Defined Network*.

2. Background Traffic

Background Traffic merupakan jumlah paket yang dikirimkan dan diterima dalam sebuah lalu lintas data pada jaringan. Dalam penelitian ini, Background Traffic yang akan digunakan adalah 100, 200, 300, 400 Mb.

b. Output

1. *Throughput*

Throughput berfungsi sebagai pengukur kecepatan pengiriman paket pada Analisa *RYU controller*.

2. *Delay*

Delay adalah transmisi dari data yang bervariasi pada jaringan melalui protokol jaringan seperti TCP/UDP [11]. Delay juga digunakan sebagai pengukur waktu yang dibutuhkan oleh sebuah data untuk sampai ke alamat tujuan.

3. *Jitter*

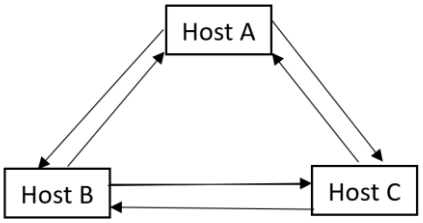
Jitter bisa disebut juga dengan total variasi delay yang menggunakan satuan waktu *millisecond* [12].

4. *Packet Loss*

Packet Loss ialah jumlah dari keseluruhan paket yang hilang atau tidak sampai ke tujuan yang dapat disebakan oleh collision ataupun congestion didalam jaringan. Packet Loss sendiri biasanya ditampilkan dalam jumlah persentase.

3.4 *Modelling*

Dalam tahap ini dilakukan perancangan skenario yang akan dilakukan selama proses *experimentation.* Skenario yang dilakukan adalah melakukan pengujian dengan pemberian traffic dengan besar 100, 200, 300, dan 400 Mb menggunakan iperf secara berulang pada jaringan tanpa VLAN dan juga VLAN dengan ilustrasi skenario seperti berikut ini:



Gambar 6. Ilustrasi Skenario Pengujian

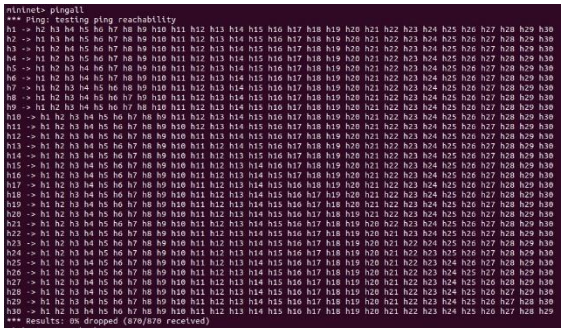
3.5 *Simulation*

Pada tahapan ini akan dilakukan implementasi model yang telah dirancang sebelumnya seperti instalasi dari tools dan juga konfigurasi-konfigurasi dari tools yang nantinya akan digunakan dalam proses *experimentation.*

4. HASIL DAN PEMBAHASAN

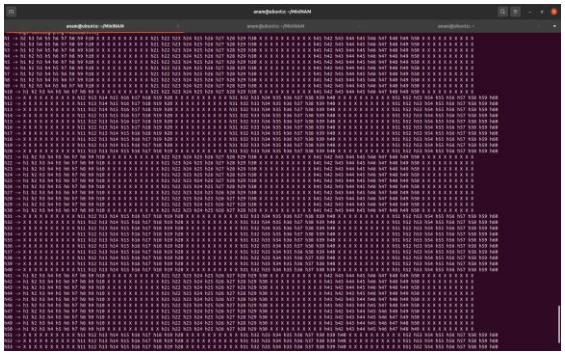
4.1 *Verification and Validation*

Di tahap ini dilakukan pengecekan apakah pengimplementasian yang dilakukan pada tahap simulasi sudah dapat berjalan.

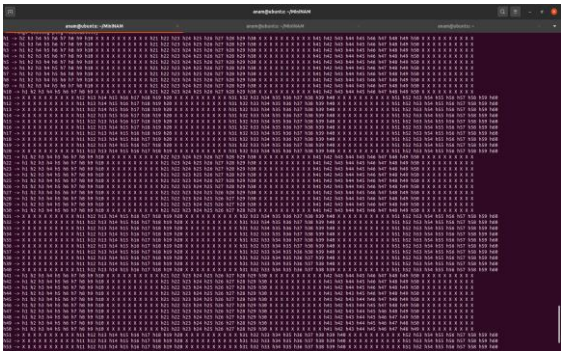


Gambar 7. Pengujian konektivitas tanpa VLAN 30 host

Gambar 8. Pengujian konektivitas VLAN 30 host



Gambar 9. Pengujian konektivitas tanpa VLAN 60 host



Gambar 10. Pengujian konektivitas VLAN 60 host

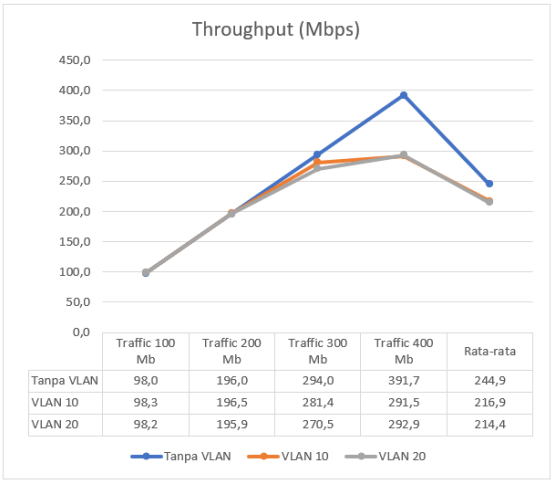
Setelah semua pengujian konektivitas sesuai dengan hasil yang diinginkan seperti pada gambar 7-10 kemudian dilanjutkan ke tahap selanjutnya yaitu tahap *experimentation.*

4.2 *Experimentation*

Tahapan ini adalah tahap dimana proses pengujian dilakukan untuk mendapatkan data yang diinginkan yang sebelumnya didefinisikan pada proses *analysis of input/output data.* Dalam prosesnya data-data tersebut diperoleh dari pengukuran menggunakan wireshark dan iperf. Dimana iperf [15] berperan untuk melakukan *generate traffic* dengan nilai 100, 200, 300, dan 400 Mb baik untuk skenario yang tidak menggunakan VLAN ataupun menggunakan VLAN. Kemudian wireshark berperan sebagai penangkap paket yang lewat saat terjadi transmisi data pada jaringan

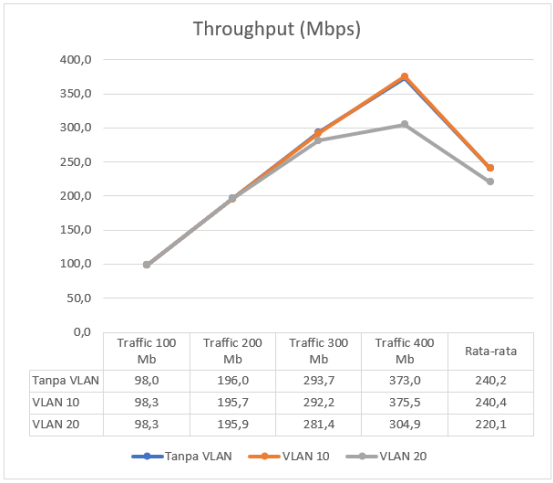
4.3 *Output Analysis*

Tahapan terakhir yang dilakukan adalah proses untuk menganalisa data yang telah diambil oleh wireshark selama proses *experimentation* atau proses pengujian sebelumnya dan tahap terakhir dari metode simulasi [13]. Parameter-parameter yang dianalisis ada *Throughput, Packet Loss, Delay/Latency,* dan *jitter* pada jaringan tanpa VLAN dan juga VLAN.



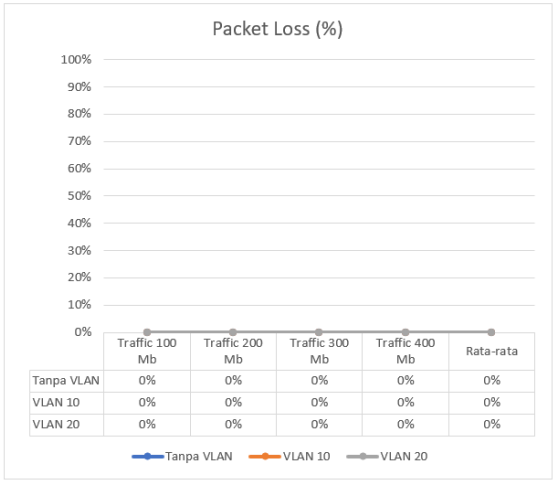
	Traffic 100 Mb	Traffic 200 Mb	Traffic 300 Mb	Traffic 400 Mb	Rata-rata
Tanpa VLAN	98,0	196,0	294,0	391,7	244,9
VLAN 10	98,3	196,5	281,4	291,5	216,9
VLAN 20	98,2	195,9	270,5	292,9	214,4

Gambar 11. Throughput 30 host



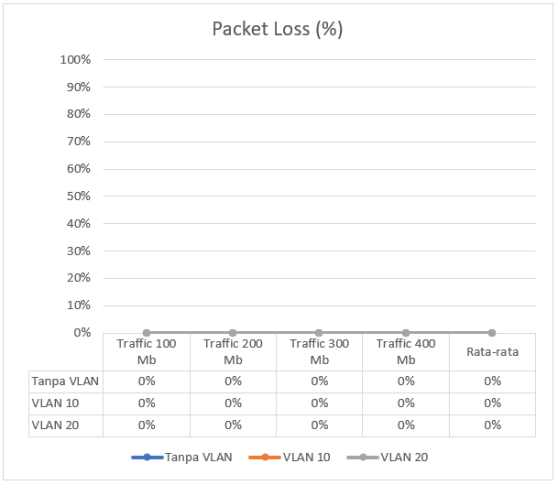
	Traffic 100 Mb	Traffic 200 Mb	Traffic 300 Mb	Traffic 400 Mb	Rata-rata
Tanpa VLAN	98,0	196,0	293,7	373,0	240,2
VLAN 10	98,3	195,7	292,2	375,5	240,4
VLAN 20	98,3	195,9	281,4	304,9	220,1

Gambar 12. Throughput 60 host



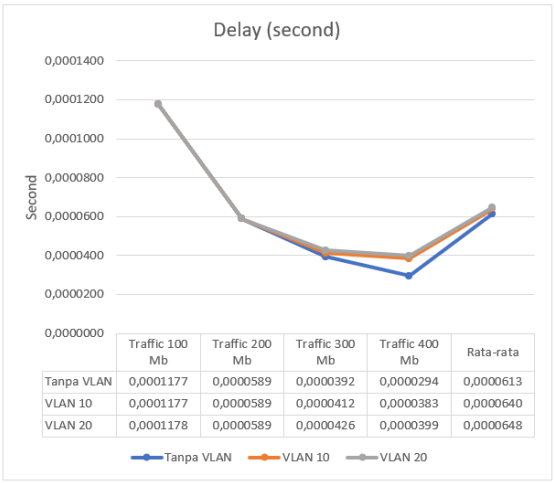
	Traffic 100 Mb	Traffic 200 Mb	Traffic 300 Mb	Traffic 400 Mb	Rata-rata
Tanpa VLAN	0%	0%	0%	0%	0%
VLAN 10	0%	0%	0%	0%	0%
VLAN 20	0%	0%	0%	0%	0%

Gambar 13. Packet Loss 30 host



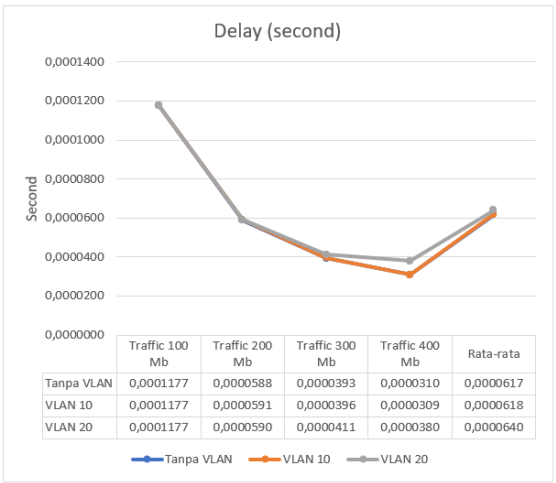
	Traffic 100 Mb	Traffic 200 Mb	Traffic 300 Mb	Traffic 400 Mb	Rata-rata
Tanpa VLAN	0%	0%	0%	0%	0%
VLAN 10	0%	0%	0%	0%	0%
VLAN 20	0%	0%	0%	0%	0%

Gambar 14. Packet Loss 60 host



	Traffic 100 Mb	Traffic 200 Mb	Traffic 300 Mb	Traffic 400 Mb	Rata-rata
Tanpa VLAN	0,0001177	0,0000589	0,0000392	0,0000294	0,0000613
VLAN 10	0,0001177	0,0000589	0,0000412	0,0000383	0,0000640
VLAN 20	0,0001178	0,0000589	0,0000426	0,0000399	0,0000648

Gambar 15. Delay 30 host



	Traffic 100 Mb	Traffic 200 Mb	Traffic 300 Mb	Traffic 400 Mb	Rata-rata
Tanpa VLAN	0,0001177	0,0000588	0,0000393	0,0000310	0,0000617
VLAN 10	0,0001177	0,0000591	0,0000396	0,0000309	0,0000618
VLAN 20	0,0001177	0,0000590	0,0000411	0,0000380	0,0000640

Gambar 16. Delay 60 host

	Traffic 100 Mb	Traffic 200 Mb	Traffic 300 Mb	Traffic 400 Mb	Rata-rata
Tanpa VLAN	0,0000999	0,0000554	0,0000345	0,0000317	0,0000554
VLAN 10	0,0000883	0,0000730	0,0000444	0,0000790	0,0000712
VLAN 20	0,0001501	0,0000808	0,0000541	0,0000156	0,0000752

Gambar 17. Jitter 30 host

	Traffic 100 Mb	Traffic 200 Mb	Traffic 300 Mb	Traffic 400 Mb	Rata-rata
Tanpa VLAN	0,0001282	0,0000386	0,0000396	0,0000468	0,0000633
VLAN 10	0,0001131	0,0000530	0,0000278	0,0000361	0,0000575
VLAN 20	0,0000929	0,0001007	0,0000391	0,0000295	0,0000656

Gambar 18. Jitter 60 host

Berdasarkan gambar 10 dan 11 *throughput* terbaik ada pada jaringan tanpa VLAN baik untuk 30 host dan 60 host. Juga ada asumsi penggunaan iperf dan wireshark secara bersamaan yang tidak dibarengi dengan storage yang cukup untuk menampung hasil pengambilan paket wireshark mengganggu proses generate traffic yang dihasilkan. Untuk *packet loss*, berdasarkan gambar 12 dan 13 memiliki nilai yang sama sebesar 0%, hal tersebut disebabkan oleh wireshark yang tidak dapat mengetahui paket yang hilang pada protokol UDP. Lalu, berdasarkan gambar 14 dan 15 nilai rata-rata delay terbaik dimiliki oleh jaringan tanpa VLAN karena *throughput* yang dihasilkan lebih bagus yang menyebabkan delay menjadi lebih kecil [14]. Namun, jika dilihat lebih seksama peningkatan jumlah host menyebabkan jaringan yang menggunakan VLAN memiliki performa yang lebih baik dibandingkan dengan yang tidak menggunakan VLAN. Terakhir untuk nilai *jitter*, berdasarkan gambar 16 dan 17 nilai rata-rata *jitter* terbaik juga ada pada jaringan tanpa VLAN yang dipengaruhi oleh nilai variasi *delay* atau jeda *delay* antar paket yang terjadi selama proses pengiriman data. Juga, jika dilihat kembali secara seksama seperti pada *delay*, dengan host yang lebih banyak performa dari VLAN lebih dapat diandalkan.

Tabel 1 Rata-rata hasil pengujian 30 dan 60 host

PARAMETER	30 HOST			60 HOST		
	Tanpa VLAN	VLAN 10	VLAN 20	Tanpa VLAN	VLAN 10	VLAN 20
Throughput (Mbps)	244,9	216,9	214,4	240,2	240,4	220,1
Packet loss (%)	0	0	0	0	0	0
Delay (s)	0,0000613	0,0000640	0,0000648	0,0000617	0,0000618	0,0000640
Jitter (ms)	0,0000554	0,0000712	0,0000752	0,0000633	0,0000575	0,0000656

Tabel 2 Perbandingan antara tanpa VLAN dan VLAN

PARAMETER	Tanpa VLAN	VLAN
Throughput (Mbps)	242,52	222,95
Packet Loss (%)	0	0
Delay (s)	0,0000615	0,0000636
Jitter (ms)	0,0000593	0,0000675

Dari data yang ditampilkan pada Tabel 2 diperoleh kesimpulan bahwa nilai rata-rata keseluruhan antara VLAN dan tanpa VLAN untuk nilai QoS *throughput, packet loss, delay, dan jitter* nilai yang diperoleh oleh VLAN lebih buruk dibandingkan tanpa VLAN dengan nilai berturut-turut 222,95 Mbps, 0%, 0,0000636 s, dan 0,0000675 ms. Sedangkan untuk tanpa VLAN nilai *throughput, packet loss, delay, dan jitter* berturut-turut 242,52 Mbps, 0%, 0,0000615 s, dan 0,0000593 ms. Namun, seiring bertambahnya host performa VLAN lebih dapat diandalkan seperti pada tabel 1. Juga didapatkan asumsi bahwa penggunaan iperf dan wireshak secara bersamaan dengan generate paket yang besar serta penyimpanan pada sistem yang kurang memadai menyebabkan paket yang di*generate* menjadi lebih kecil saat pengujian VLAN.

KESIMPULAN

a. VLAN dapat diimplementasikan dengan baik pada *Software Defined Network* dengan Ryu Controller yang diberi background traffic 100, 200, 300, dan 400 Mb baik dengan 30 host maupun 60 host.
b. Nilai *Qos Throughput, Packet Loss, delay* dan *Jitter* keseluruhan VLAN kurang baik dibandingkan tanpa VLAN dengan nilai berturut-turut 222,9 Mbps, 0%, 0,0000636 s, dan 0,0000675 ms. Namun, seiring bertambahnya *host,* VLAN lebih dapat diandalkan.

DAFTAR PUSTAKA

[1] Ahmed, Khandakar & Blech, Jan & Gregory, Mark & Schmidt, Heinrich (Heinz). (2018). Software Defined Networks in Industrial Automation. *Journal of Sensor and Actuator Networks*. 7. 33. 10.3390/jsan7030033.

[2] Sharma, Karthik Neelakanta. (2015). *A Scalable and Fault Tolerant OpenFlow Controller*. Tesis. Masters in Computer Science.The University of Waikato. 2-3

[3] Hidayat, M. H. & Rosyid, N. R. (2017). Analisis kinerja dan karakteristik Arsitektur *Software-Defined-Network* berbasis OpenDaylight Controller. Teknologi Jaringan, Departemen Teknik Elektro dan Informatika: Universitas Gajah Mada

[4] Ryu. (2017). *Component-Based Software Defined Networking Framework: Build Sdn Agilely*. https://Ryu-sdn.org/index.html. Diakses pada 10 April 2021

[5] Mininet. (2022). *Mininet Overview*. http://mininet.org/overview/. Diakses pada 10 April 2022

[6] Sood, Manu & Sharma, K.K. (2014). Mininet as a Container Based Emulator for Software Defined Networks. *International Journal of Advanced Research in Computer Science and Software Engineering*. Vol. 4

[7] MiniNAM. (2020). MiniNAM: *A Network Animator for Mininet*. https://github.com/uccmisl/MiniNAM. Diakses pada 10 April 2021

[8] Kabir, Md. (2020). *Design a VLAN (Virtual Local Area Network) Based Network*. 10.13140/RG.2.2.29163.57120.

[9] Novita, Romasella dkk (2021). Analisis Keamanan Wifi Menggunakan Wireshark. *JES (Jurnal Elektro Smart)* Vol. 1, No. 1.

[10] Sofana, Iwan. (2011). Teori dan Modul Praktikum Jaringan Komputer. Bandung: Modula.

[11] Sujatha Krishanmoorthy et al (2020). A Study on Optimization of Network latency and Pocket loss Rate. *IOP Conf*. Ser.: Mater. Sci. Eng. 937 012054

[12] Paulson, Eberechukwu et.al (2019). On the latency and jitter evaluation of software defined networks. *Bulletin of Electrical Engineering and Informatics*. 8. 10.11591/eei.v8i4.1578.

[13] Stefan Mahlknecht, Sajjad A. Madani and Jawad Kazmi (2010). *Wireless Sensor Networks: Modelling and Simulation*, Discrete Event Simulations, Aitor Goti (Ed.), ISBN: 978-953-307-115-2, InTech, Available from: http://www.intechopen.com/books/discrete-event-simulations/wireless-sensor-networks-modelling-and-simulation

[14] Mehrnaz Moudi & Mohamed Othman (2020) On the relation between network throughput and delay curves, *Automatika*, 61:3, 415-424, DOI: 10.1080/00051144.2020.1774731

[15] iPerf—*The Network Bandwidth Measurement* Tool. https://iperf.fr/