

IMPLEMENTASI VLAN PADA SOFTWARE DEFINED NETWORK MENGGUNAKAN PROTOKOL OPENFLOW

Mochammad Ridhwan Nurfaal¹, Ali Akbar Rismayadi²

¹Universitas Adhirajasa Reswara Sanjaya
Jalan Sekolah Internasional No.1-2, Antapani – Bandung, 022-7100124
e-mail: ridwansacil9g@gmail.com

²Universitas Adhirajasa Reswara Sanjaya
Jalan Sekolah Internasional No.1-2, Antapani – Bandung, 022-7100124
e-mail: ali@ars.ac.id

Abstrak

Jaringan untuk saat ini menjadi hal yang sangat dibutuhkan, karena menjadi infrastruktur dalam perkembangannya. Teknologi jaringan pada saat ini memiliki keterbatasan yaitu kompleksitas yang mengarahkan ke arah statis, kebijakan yang kurang konsisten, ketidakmampuan untuk diukur dan ketergantungan terhadap *vendor*. Banyak cara untuk menyesuaikan kinerja jaringan menjadi lebih optimal dan lebih efisien, salah satu cara dalam jaringan yang sangat menarik yang dapat diprogram oleh suatu perangkat lunak yaitu *Software Defined Network*. Pengerjaan proyek akhir ini yaitu dapat membuktikan kinerja dari VLAN berbasis SDN menggunakan *opendaylight controller*. Kinerja jaringan ini akan dibuktikan dengan cara ping seluruh *host* ke server dan dilihat juga dari sisi *Quality Of Service* adalah *Throughput*, *Delay*, *Packet loss* dan *Jitter*. Hasil analisa performa antara jaringan berbasis *Software Defined Network* diperoleh nilai rata-rata dari QOS SDN *throughput* 50,552 Kb/s, untuk *packet loss* 0%, untuk *delay* 51,54 ms, untuk *jitter* 51,52 ms.

Kata kunci : *Software Defined Network*, VLAN, *Opendaylight*, *Openflow*, QOS

Abstract

The network for now is a much needed thing, because it becomes an infrastructure in its development. Network technology currently has limitations including complexity that leads towards static, inconsistent policies, inability to be measured and reliance on vendors. There are many ways to adjust the performance of the network to be more optimal and more efficient, there is one way in which a very interesting network that can be programmed by Software Defined Network software. The final project work is to prove the performance of SDN-based VLAN using the.opendaylight controller. This network performance will be proven by pinging the entire host to the server and viewed also in terms of Quality Of Service namely Throughput, Delay, Packet loss and Jitter. The results of performance analysis between Software Defined Network based networks obtained an average value of QOS SDN throughput of 50,552 Kb/s, for packet loss of 0%, for a delay of 51.54 ms, for jitters of 51.52 ms.

Keywords : *Software Defined Network*, VLAN, *Opendaylight*, *Openflow*, QOS

1. LATAR BELAKANG

Jaringan untuk saat ini menjadi hal yang sangat dibutuhkan, karena menjadi infrastruktur dalam perkembangannya. Banyak cara untuk menyesuaikan kinerja jaringan menjadi lebih optimal dan lebih efisien, salah satu cara yang dapat diprogram oleh suatu perangkat lunak yaitu *Software Defined Network*. *Switch* memiliki dua bagian disebut dengan bagian kontrol dan bagian data, tugas SDN disini memisahkan kontrol dan data dalam jaringan tersebut (Hunaifi, 2019). *Software Defined Network* adalah

konsep baru dalam mengelola, mendesain dan mengimplementasikan jaringan untuk mendukung kebutuhan, serta inovasi dalam jaringan komputer yang semakin kompleks. Konsep dasar SDN ini adalah dengan melakukan pemisahan antara *control plane* dan *data plane* maka dibuat *Software Defined Network* yang dikembangkan di UC Berkley dan Stanford University pada tahun 2008 dan pada tahun 2011 teknologi SDN dan *Openflow* dipromosikan oleh *Open Networking Foundation* (Roni Fernando Simarmata, Rohmat Tulloh, 2018).

Jaringan konvensional yang cenderung bersifat tertutup serta terdistribusi diubah dengan teknologi baru SDN menjadi bersifat terbuka (*Open Source*), (Tulloh, 2017). Dalam implementasi penggunaan jaringan selain kelebihan dari *sharing resource*, terkadang ada permasalahan dalam penggunaan pada jaringan tersebut, seperti kurangnya sebuah manajemen, konfigurasi keamanan dan kebijakan keamanan (*security policy*) yang berkaitan dengan komunikasi data dan informasi pada jaringan, apalagi jika *user* yang terhubung dengan jaringan tersebut cukup banyak. Sehingga terjadi permasalahan seperti koneksi yang lambat dan kerentanan dalam keamanan jaringan. Dengan meningkatnya pengguna maka manajemen jaringan dibutuhkan untuk meningkatkan performa dan keamanan jaringan tersebut (Fuadi, 2016). Untuk mengatasi hal tersebut, penerapan *Virtual Local Area Network (VLAN)* adalah sebuah solusi yang dapat digunakan. *VLAN* dapat mengelompokkan pengguna yang ada dalam jaringan menjadi lebih kecil, walaupun terpisah secara fisik namun seperti berada dalam satu jaringan LAN yang sama (Abdurrahman, 2016).

Jaringan Komputer

Menurut (Yudianto, 2007) dalam buku (Program Studi Sistem Informasi, 2019) jaringan komputer merupakan suatu sistem yang terdiri dari komputer yang dirancang untuk dapat berbagi sumber daya, berkomunikasi dan dapat mengakses informasi. Tujuan dari jaringan komputer agar mampu mencapai tujuannya dan setiap jaringan komputer dapat meminta dan memberikan layanan (Program Studi Sistem Informasi, 2019).

Jaringan Client-Server

Tipe jaringan client-server menghubungkan *host* yang digunakan sebagai *server* yang memberikan layanan komputer dan *client* dapat *me-request* layanan atau dapat menggunakan fasilitas yang disediakan *server*. *Server* di jaringan tipe *client-server* disebut dengan *dedicated server* karena berperan sebagai server penyedia fasilitas kepada (Hadi, 2016).

Local Area Network

Local Area Network adalah jaringan komputer yang mencakup dalam satu ruang, satu gedung. Sebagai contoh : jaringan dalam satu kampus merupakan lokasi perusahaan yang tergolong LAN, sekolah ataupun perkantoran yang membutuhkan hubungan atau koneksi antara dua computer

atau lebih antar ruangan. LAN umumnya menggunakan media transmisi berupa kabel, namun ada juga yang tidak menggunakan kabel yang disebut *wireless lan* atau LAN tanpa kabel. Kecepatan LAN berkisar dari 10Mbps sampai 1Gbps (Nidhom, 2019).

Virtual Local Area Network

Teknologi VLAN membagi jaringan secara logika menjadi beberapa VLAN ID. VLAN memungkinkan banyak VLAN ID di dalam suatu jaringan. VLAN ini bertujuan untuk membatasi *broadcast traffic* hanya untuk penerima yang memiliki VLAN ID. Jaringan dikonfigurasi secara virtual dan tidak tergantung pada lokasi perangkat secara fisik. VLAN dipilih karena mengatur jaringan berdasarkan teknik pengklasifikasiannya, yaitu melalui pengalamatan MAC, *port*, dan sebagainya yang menjadikan jaringan VLAN fleksibel. Proses pengklasifikasian disimpan ke dalam suatu tabel yang berisi pengalamatan suatu VLAN (*tagging*), dimana apabila pengalamatan berdasarkan *port* yang digunakan, maka *port* tersebut harus dapat diindikasikan oleh *database* (Tulloh, 2017).

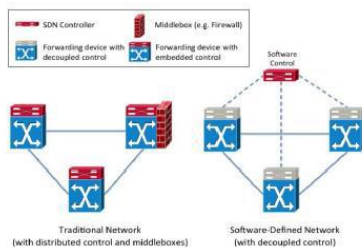
Topologi Linier

Topologi Linier atau biasanya disebut topologi bus beruntut. Topologi ini menggunakan satu kabel utama untuk menghubungkan tiap titik pada setiap komputer (Sitanggang, 2019).

Software Defined Network

Software Defined Network adalah arsitektur jaringan yang dimana kontrol jaringan dapat dipisahkan dari forwarding dan secara langsung dapat diprogram. Migrasi kontrol ini yang memungkinkan dapat mengakses ke perangkat komputasi, hal ini yang menjadi dasar infrastruktur untuk diubah menjadi aplikasi dan layanan jaringan, yang dapat memperlakukan jaringan sebagai virtual (Hunaifi et al., 2019).

SDN menawarkan konsep virtualisasi topologi jaringan dan memungkinkan administrator melakukan pengaturan pada *control plane*. Hal ini juga membuat SDN bersifat *managable*, dimana apabila ada suatu *maintenance*, seorang admin tidak perlu untuk mengurus tiap-tiap *switch*, cukup dengan mengkonfigurasi pada *controller*-nya saja. Selain itu, SDN juga dapat menjanjikan *bandwidth* yang lebar, *cost-effective*, dan *manageable*.



Gambar II. *Software Defined Network*

OpenDaylight

Controller SDN adalah aplikasi SDN yang mengelola *flow control* untuk mengaktifkan *intelligence* networking. Controller SDN bekerja berdasarkan protokol seperti OpenFlow yang memungkinkan server dapat memberitahu kemana paket akan dikirimkan. Perangkat meneruskan paket data yang diterima berdasarkan aturan yang *disetting* dari controller. Controller menentukan apa yang harus dilakukan dengan paket dan mengirimkan aturan baru untuk perangkat, sehingga dapat meneruskan paket data di masa depan dengan cara yang sama. Terdapat banyak jenis controller yang tersedia secara open source salah satunya adalah OpenDaylight. OpenDaylight merupakan sebuah infrastruktur *tensible*, *scalable* dan *multi-protocol*, dibangun untuk penyebaran SDN di jaringan heterogen, *multi-vendor*, serta *modern* (Hidayat & Rosyid, 2017).

Protokol Openflow

OpenFlow adalah protokol yang digunakan untuk komunikasi antara switch dengan kontroler dalam SDN Arsitektur. OpenFlow menyediakan protokol terbuka untuk memprogram *flowtable* pada switch dan router yang berbeda, bukan hanya itu saja OpenFlow juga dapat membuat administrator jaringan berpartisipasi langsung ke dalam *traffic* data untuk membuat atau meneliti aliran data, dengan ini peneliti dapat mengontrol aliran data mereka sendiri dengan memilih rute paket data yang mereka ikuti dan memproses yang akan diterima oleh receiver. (Hunaifi et al., 2019).

Open Virtual Switch

Open vSwitch merupakan perangkat lunak kode terbuka yang dirancang untuk digunakan sebagai vswitch (*virtual switch*) dalam lingkungan server virtual. Open vSwitch dapat diinstall pada perangkat komputer atau perangkat keras khusus. Open vSwitch dikembangkan oleh Nicira Network dan telah banyak digunakan karena

kestabilannya dan dapat diprogram. Open vSwitch dijalankan sebagai modul pada kernel Linux menggantikan modul *bridge*. Open vSwitch mendukung banyak manajemen jaringan seperti NetFlow, sFlow, CLI dan lain sebagainya. Open vSwitch mendukung protokol OpenFlow, dimana Open vSwitch sebagai *data plane* dikontrol oleh controller sebagai *control plane*. Model komunikasi *data plane* dengan *control plane*. Data plane dapat dikontrol untuk melakukan QoS, *tunneling*, dan *filtering rules*. Dukungan pengontrolan jarak jauh membuat Open vSwitch dapat digunakan untuk melakukan proses migrasi kebijakan jaringan. Karena fleksibilitasnya, data plane pada Open vSwitch dapat dipartisi secara logik (Effendi et al., 2017).

GNS3

Menurut (Welsh, 2013) pengertian diatas bermakna bahwa GNS3 adalah Simulator Jaringan Grafis yang memungkinkan pengguna menjalankan beberapa sistem yang diemulasi termasuk router Cisco, router Juniper, router Vyatta, virtual machine Linux, dan virtual machine Windows. Untuk membuat GNS3 melakukan simulasi ini bukanlah hal yang mudah, terutama jika ingin menjelajah lebih dari sekadar topologi jaringan sederhana.

Docker Container

Docker Container merupakan sebuah image yang dapat dikemas dan dibaca tulis, container berjalan diatas image. Pada setiap perubahan yang disimpan pada container akan menyebabkan terbentuknya layer baru di atas base image. Kita dapat melakukan instalasi aplikasi didalamnya dan melakukan penyimpanan.

2. METODE PENELITIAN

2.1. Studi Literatur

Metode pengumpulan data yang dilakukan dengan mempelajari literatur yang berkaitan dengan protokol *openflow*, pembahasan mengenai masalah *transfer* data yang terjadi di dalamnya, serta pengumpulan informasi beberapa contoh metode yang biasa digunakan pada penelitian sebelum-sebelumnya.

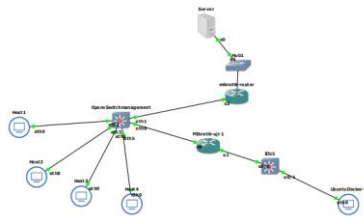
2.2. Analisis

Pada tahap ini, dilakukan analisis terhadap penggunaan VLAN pada konfigurasi protokol Openflow. Untuk menentukan *Quality of Service* jaringan *software defined network* menggunakan aplikasi *wireshark*. Penulis akan melakukan analisis terhadap jaringan menggunakan SDN protokol *openflow*. Penulis akan

melakukan pengujian dengan melakukan ping *host* ke *server* menggunakan *bandwidth* secara berkala setiap beberapa detik akan menambah *bandwidth*, pengujian ini dilakukan selama 30 menit.

2.3. Desain

Topologi yang digunakan adalah topologi *linear* dengan 4 *host* dan 1 *router* yang terhubung dengan *open vswitch* untuk konfigurasi VLAN.



Gambar III. 1. Topologi Jaringan

2.4. Pengujian

Pengujian yang dilakukan untuk memastikan konfigurasi jaringan menggunakan *SDN* sesuai dengan yang telah di konfigurasi. Host yang sudah terkoneksi akan dilakukan untuk menentukan QoS adalah *throughput*, *delay*, *jitter* dan *packet loss*.

a. Throughput

Throughput adalah kinerja jaringan yang terukur. *Throughput* merupakan jumlah total kedatangan paket yang berhasil yang diamati pada *destination* selama *interval* waktu tertentu dibagi oleh durasi *interval* waktu tersebut. Rumus yang digunakan untuk mencari *throughput* adalah

$$Throughput = \frac{\text{paket data diterima}}{\text{lama pengamatan}}$$

Tabel III. 1 Kategori *Throughput*

Kategori <i>Throughput</i>	<i>Throughput</i>	Indeks
Sangat Bagus	100%	4
Bagus	75%	3
Sedang	50%	2
Jelek	<25%	1

b. Packet loss

Packet loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena collision dan congestion pada jaringan. *Packet loss* dapat dihitung dengan rumus sebagai berikut :

$$\frac{\text{Packet} - \text{Loss}}{\text{Paket data dikirim}} \times 100 =$$

Tabel III. 2 Kategori *packet loss*

Kategori <i>packet loss</i>	<i>Packet Loss</i>	Indeks
Sangat Bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

c. Delay

Delay merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama. Untuk menghitung *Delay* dapat dihitung dengan rumus, sebagai berikut :

$$Delay = \frac{\text{Total delay}}{\text{Total paket diterima}}$$

Tabel III. 3 Kategori *Delay*

Kategori <i>Delay</i>	<i>Delay</i>	Indeks
Sangat Bagus	<150ms	4
Bagus	15 s/d 300 ms	3
Sedang	300 s/d 450 ms	2
Jelek	>450 ms	1

d. Jitter

Jitter adalah variasi *delay* yang disebabkan oleh variasi-variasi panjang antrian dalam waktu mengolah data. Untuk menghitung *jitter* menggunakan persamaan, sebagai berikut :

$$Jitter = \frac{\text{Total variasi delay}}{\text{Total paket diterima}}$$

Tabel III. 4 Kategori *Jitter*

Kategori <i>jitter</i>	<i>Jitter</i>	Indeks
Sangat Bagus	0 ms	4
Bagus	0 s/d 75 ms	3
Sedang	75 s/d 125 ms	2
Jelek	125 s/d 225 ms	1

3. HASIL DAN PEMBAHASAN

3.1. Pengujian

a. *Throughput*

$$\text{Throughput} = \frac{\text{packet data diterima}}{\text{lama pengamatan}}$$

Penulis mendapatkan hasil *throughput* yaitu :

$$\text{Throughput} = \frac{11411282 \text{ Bytes}}{1805.862 \text{ seconds}}$$

$$\text{Throughput} = 6.319 \text{ Bytes/s}$$

$$\text{Throughput} = 6,319 \text{ KB/s}$$

$$\text{Throughput} = 50,552 \text{ Kb/s}$$

Hasil dari *throughput* yaitu 50,552 Kb/s dengan indeks 4, karena *throughput* dapat dilihat pada *packet loss* menunjukkan 0. Maka dari itu, *throughput* dikategorikan sangat bagus.

b. *Packet Loss*

$$\text{Packet Loss} = \frac{\text{Paket data dikirim} - \text{paket data diterima}}{\text{Paket data dikirim}} \times 100 =$$

Penulis mendapatkan hasil *packet loss* yaitu:

$$\text{Packet Loss} = \frac{35038 - 35038}{35038} \times 100$$

$$\text{Packet Loss} = 0\%$$

Hasil dari *packet loss* yaitu 0% dengan indeks 4, karena *packet loss* menunjukkan hasil 0%. Maka dari itu, *packet loss* dikategorikan sangat bagus.

c. *Delay*

$$\text{Delay} = \frac{\text{Total delay}}{\text{Total paket diterima}}$$

Penulis mendapatkan hasil *delay* yaitu :

$$\text{Delay} = \frac{1805,862108 \text{ seconds}}{35037}$$

$$\text{Delay} = 0,051541573 \text{ s}$$

$$\text{Delay} = 51,541573 \text{ ms}$$

Hasil dari *delay* yaitu 51,541573 ms dengan indeks 4, karena *delay* menunjukkan hasil <150ms. Maka dari itu, *delay* dikategorikan sangat bagus.

d. *Jitter*

$$\text{Jitter} = \frac{\text{Total variasi delay}}{\text{Total paket diterima}}$$

Penulis mendapatkan hasil *jitter* yaitu :

$$\text{Jitter} = \frac{1805,169165 \text{ seconds}}{35036}$$

$$\text{Jitter} = 0,051523267 \text{ s}$$

$$\text{Jitter} = 51,523267 \text{ ms}$$

Hasil dari *jitter* yaitu 51,523267 ms dengan indeks 3, karena *jitter* menunjukkan hasil 0 s/d 75 ms. Maka dari itu, *jitter* dikategorikan bagus.

Tabel IV QoS Dari SDN *Openflow*

Jenis QoS	Hasil penelitian	Indeks	Kategori
<i>Throughput</i>	50,552 Kb/s	4	Sangat Bagus
<i>Packet Loss</i>	0%	4	Sangat Bagus
<i>Delay</i>	51,541573 ms	4	Sangat Bagus
<i>Jitter</i>	51,523267 ms	3	Bagus

4. PENUTUP

4.1. Kesimpulan

Dari hasil implementasi dan analisa pada penelitian ini, didapatkan kesimpulan dari penelitian yang dilakukan oleh penulis yaitu *throughput* dengan hasil 50,552 Kb/s dikategorikan sangat bagus, *packet loss* dengan hasil 0% dikategorikan sangat bagus, *delay* dengan hasil 51,541573 ms dikategorikan sangat bagus dan *jitter* dengan hasil 51,523267 ms dikategorikan bagus. Dari hasil penelitian jaringan *Software Defined Network* menggunakan protokol *openflow* termasuk dalam kategori sangat bagus menurut TIPHON. Maka jaringan berbasis *Software Defined Network* menggunakan protokol *openflow* dapat

menggantikan jaringan konvensional yang masih digunakan pada saat ini.

4.2. Saran

- a. Skenario pengujian untuk penelitian selanjutnya dapat ditambahkan untuk dapat mengetahui lebih lanjut karakteristik VLAN pada jaringan SDN.
- b. Dapat digunakan controller lain selain *OpenDaylight* pada penelitian selanjutnya untuk dapat mengetahui kinerja dan pengaruh dari jenis *controller* yang berbeda pada VLAN di jaringan SDN.
- c. Dapat dilakukan penelitian dengan topologi yang berbeda, misalnya penambahan jumlah *switch* maupun VLAN ID agar dapat mengetahui pengaruhnya pada performansi jaringan tersebut.

DAFTAR PUSTAKA

- Abdurrahman, H. (2016). Analisis dan perancangan jaringan iain langsa berbasis VLAN. *Jurnal TIMES*, *V*(2), 58–62.
- Effendi, M. R., Hamidi, E. A. Z., & Saepulloh, A. (2017). Implementasi GRE Tunneling Menggunakan Open vSwitch Pada Jaringan Kampus. *TELKA - Telekomunikasi, Elektronika, Komputasi Dan Kontrol*, *3*(2), 103–111. <https://doi.org/10.15575/telka.v3n2.103-111>
- Fuadi, K. (2016). *Analisis dan Implementasi Virtual Local Area Network (VLAN) untuk Optimalisasi Keamanan Jaringan Local Area Network*. 1–52. http://digilib.uin-suka.ac.id/22214/2/12650016_BAB-I_IV-atau-V_DAFTAR-PUSTAKA.pdf
- Hadi, A. (2016). *Administratif Jaringan Komputer*. Prenada Media. https://books.google.co.id/books?id=a9tUDwAAQBAJ&dq=client-server+jaringan+adalah&hl=id&source=gbs_navlinks_s
- Hidayat, M. H., & Rosyid, N. R. (2017). Analisis Kinerja dan Karakteristik Arsitektur Software-Defined Network Berbasis OpenDaylight Controller. *Citee*, *2085–6350*, 194–200.
- Hunaifi, N., Akbar, R. F., Studi, P., Informasi, S., Jaringan, M., & Pendahuluan, I. (2019). Analisis Kinerja Jaringan Berbasis Software Definition Network Dengan Protokol Openflow Di Rri Bandung. *Infotronik: Jurnal Teknologi Informasi Dan Elektronika*, *4*(1), 24–32. <https://doi.org/10.32897/infotronik.2019.4.1.4>
- Nidhom, A. M. (2019). *Komputer Terapan Jaringan Serta Pengaplikasiannya*. Ahlimedia Book. https://books.google.co.id/books?id=PL2xDwAAQBAJ&hl=id&source=gbs_navlinks_s
- Program Studi Sistem Informasi, U. M. C. (2019). *Jaringan Komputer: Konsep dan Studi Kasus*. Seribu Bintang. https://books.google.co.id/books?id=g8SZDwAAQBAJ&dq=landasan+teori+jaringan+komputer&hl=id&source=gbs_navlinks_s
- Roni Fernando Simarmata, Rohmat Tulloh, Y. S. H. (2018). Simulasi Jaringan Software Defined Network Menggunakan Protokol Routing Ospf Dan Ryu Controller. *E-Proceeding of Applied Science*, *4*(3), 2887–2896.
- Sitanggang, R. (2019). *SISTEM INFORMASI LAPORAN PENJUALAN KOMPUTER BERBASIS LAN*. *4*(1).
- Tulloh, R. (2017). Analisis Performansi VLAN Pada Jaringan Software Defined Network (SDN). *Jurnal Infotel*, *9*(4), 406. <https://doi.org/10.20895/infotel.v9i4.319>
- Welsh, “RedNectar” Chris. (2013). *GNS3 Network Simulation Guide*. Packt Publishing Ltd. https://books.google.co.id/books?id=OQq8AQAQBAJ&dq=gns3+ebook&hl=id&source=gbs_navlinks_s