

# IMPLEMENTASI VPLS PADA JARINGAN *SOFTWARE DEFINED NETWORK* (SDN) DENGAN MENGGUNAKAN ONOS CONTROLLER BERBASIS RASPBERRY-PI 3

## IMPLEMENTATION OF VPLS ON *SOFTWARE DEFINED NETWORK* USING ONOS CONTROLLER BASED ON RASPBERRY-PI 3

Rizal Cerdas Kurniawan S.<sup>1</sup>, Rohmat Tulloh, S.T., M.T.<sup>2</sup>, Dr. Indrarini Dyah Irawai S.T., M.T.<sup>3</sup>

<sup>1</sup>Prodi D3 Teknologi Telekomunikasi <sup>2</sup>Fakultas Ilmu Terapan, <sup>3</sup>Universitas Telkom

[1rizalcerdaskurniawan@gmail.com](mailto:rizalcerdaskurniawan@gmail.com), [2rohmatth@telkomuniversity.ac.id](mailto:rohmatth@telkomuniversity.ac.id),

[3indrarini@telkomuniversity.ac.id](mailto:indrarini@telkomuniversity.ac.id)

### Abstrak

Pada jaman sekarang jaringan komputer konvensional masih memiliki kekurangan yang signifikan. Seperti perangkat yang masih sulit dikonfigurasi karena menggunakan konfigurasi tingkat rendah (*low level configuration*) dan konfigurasi yang memakan waktu lama apalagi untuk skala jaringan yang besar. Dengan adanya teknologi *Software Defined Network* (SDN), maka seluruh perangkat jaringan akan terpusat pada satu *controller* yang dapat mengontrol, mengatur dan memonitor jaringan dengan lebih efektif.

Dalam proyek akhir ini teknologi SDN diterapkan menggunakan *Open Network Operating System* (ONOS) yang mampu memisahkan *control plane* dan *data plane* pada perangkat jaringan. ONOS dipasang pada perangkat Raspberry-Pi 3 untuk meminimalisir penggunaan perangkat *Personal Computer* (PC) pada jaringan SDN. ONOS juga *support* untuk fitur *Virtual Private LAN Service* (VPLS). VPLS adalah *software* pada ONOS yang mampu membuat suatu jaringan diatas jaringan lain Layer 2 dengan bantuan protokol *OpenFlow*. *Software* tersambung pada host *network* dengan mengkoneksikan *overlay network* yang terhubung dengan protokol *OpenFlow data plane*. VPLS dapat memaksimalkan bandwidth yang ada saat terjadi komunikasi antar perangkat jaringan yang jauh dan seolah – olah berada pada jaringan lokal.

Pada Proyek Akhir ini telah dirancang sistem yang mampu memudahkan administrator jaringan dalam mengontrol, mengatur dan memonitor jaringan dengan menggunakan satu *controller* terpusat berbasis pada Raspberry-Pi 3 dengan implementasi layanan VPLS. Hasil pengujian menunjukkan QoS pada Proyek Akhir ini memiliki hasil yang baik berdasarkan pada standard *Telecommunication and Internet Protocol Harmonization Over Network* (TIPHON) dengan pengukuran tanpa *background traffic* dan dengan *background traffic* sebesar 200 Mb, 400 Mb, 600 Mb, dan 800 Mb. Nilai *throughput* tanpa *background traffic* 3,623 Mb/s dan nilai *throughput* pada *background traffic* tertinggi 3,473 Mb/s. Nilai *packet loss* tanpa *background traffic* yaitu 2,1 % dan nilai *packet loss* pada *background traffic* tertinggi mencapai 7,2 %. Nilai rata – rata *delay* tanpa *background traffic* adalah 2,75 s dan nilai rata-rata *delay* pada *background traffic* tertinggi mencapai 3,59 s. Nilai rata – rata *jitter* pada tanpa *background traffic* yaitu 0,0352 s dan nilai rata – rata *jitter* pada *background traffic* tertinggi adalah 0,0414.

**Kata kunci :** *Software Defined Network*, VPLS, *data plane*, *control plane*, *OpenFlow*, Raspberry-Pi 3, ONOS, TIPHON.

### Abstract

Today, conventional computer networks still have significant shortcomings. Such a device is still difficult to configure because it uses a low level configuration and a configuration that takes a long time especially for large scale networks. With the *Software Defined Network* (SDN) technology, all network devices will be centered on one controller that can control, manage and monitor the network more effectively.

In this final project SDN technology is applied using the *Open Network Operating System* (ONOS) which is able to separate the control plane and the data plane on a network device. ONOS is installed on Raspberry-Pi 3 devices to minimize the use of *Personal Computer* (PC) devices on SDN networks. ONOS also supports *Virtual Private LAN Service* (VPLS) features. VPLS is *software* on ONOS that is able to create a network on another Layer 2 network with the help of the *OpenFlow* protocol. *Software* is connected to the host network by connecting

to network overlays that are connected to the OpenFlow data plane protocol. VPLS can maximize the existing bandwidth when there is communication between remote network devices and as if they were on a local network.

This Final Project has designed a system that is able to facilitate network administrators in controlling, managing and monitoring the network using a centralized controller based on Raspberry-Pi 3 with the implementation of VPLS services. The test results show QoS in this Final Project has good results based on the Telecommunication and Internet Protocol Harmonization Over Network (TIPHON) standard with measurements without background traffic and with background traffic of 200 Mb, 400 Mb, 600 Mb, and 800 Mb. The throughput value without background traffic is 3.623 Mb / s and the highest throughput value for background traffic is 3.473 Mb / s. The value of packet loss without background traffic is 2.1% and the value of packet loss on the highest background traffic reaches 7.2%. The average value of delay without background traffic is 2.75 s and the average value of delay on the highest background traffic reaches 3.59 s. The average value of jitter on no background traffic is 0.0352 s and the average value of jitter on the highest background traffic is 0.0414.

**Keyword :** Software Defined Network, VPLS, data plane, control plane, OpenFlow, Raspberry-Pi 3, ONOS, TIPHON.

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Infrastruktur jaringan yang hampir seluruhnya dikelola oleh vendor menyebabkan semakin kompleksnya jaringan yang dibangun, sehingga dibutuhkan sistem yang dapat mengelola dan mengimplementasikan jaringan agar dapat mendukung kebutuhan yang beragam. *Software Defined Network* (SDN) adalah konsep baru dalam mengontrol, mengimplementasi serta mengelola suatu jaringan yang mendukung kebutuhan dan inovasi di bidang telekomunikasi yg semakin lama semakin berkembang dan kompleks. Konsep dasar dari SDN yaitu dengan memisahkan antara *control plane* dan *data plane*. SDN mampu membuat jaringan baik berskala kecil maupun berskala besar mampu terkontrol menggunakan 1 *controller* yang terpusat. Beberapa kontroler yang ada pada SDN diantaranya adalah POX, RYU, OpenDaylight, dan ONOS. Untuk mendukung jaringan dengan skala besar, *Network Operating System* harus memenuhi persyaratan yang menuntut *Scalability*, *Performance*, dan *Availability*. Untuk mengatasi tantangan tersebut, ONF memperkenalkan *Open Network Operating System* (ONOS). ONOS adalah *controller* berbasis Java yang memanfaatkan *Open Service Gateway initiatives* (OSGi) untuk mempertahankan aplikasi dengan memasang, memulai, menghentikan, memperbarui, dan melepas pemasangan dari jarak jauh tanpa perlu reboot.

Pada penelitian [9], telah dilakukan analisis perbandingan beberapa *controller* SDN seperti POX, RYU, Maestro, Floodlight, dan ONOS. *Controller* POX dan Maestro tetap bekerja dengan sangat baik bahkan ketika menggunakan banyak perangkat switch (20-120) yang terlibat dalam jaringan. Namun jika menggunakan perangkat switch yang sedikit *controller* ONOS memiliki kinerja yang lebih baik, ONOS juga dapat bertindak sebagai pesawat kontrol SDN untuk jaringan lokal (LAN) dan jaringan pusat data. Pada penelitian [17], telah dilakukan simulasi jaringan virtual berbasis SDN menggunakan POX, namun pada penelitian tersebut hanya melakukan pengukuran QoS melalui perintah PING dan iperf. Pada penelitian [7], telah dilakukan perancangan dan implementasi routing EBGp menggunakan ONOS controller dan dilakukan pengukuran QoS menggunakan standard ITU-T G.1010. Dari hasil studi literatur, maka dapat dibuat tabel perbandingan penelitian sebelumnya seperti ditunjukkan pada Tabel 1.1.

Tabel 1.1 Perbandingan Penelitian Sebelumnya

Penelitian	Simulasi	Impementasi	Controller	Aplikasi
M. W. Putra, E. S. Pramukantoro, W. Yahya, 2018 [9]	✓	-	Floodlight, Maestro, RYU, POX dan ONOS	Mininet
Ummah, 2016 [18]	✓	-	POX	Jaringan Virtual ( <i>Virtual Network Description</i> )
M. N. Yaqin, Yaqin, R. Tulloh, I. D. Irawati, 2019 [7]	✓	✓	ONOS	Routing EBGp

Berdasarkan pada Tabel 1.1, dengan perbandingan penelitian sebelumnya diatas maka pada penelitian ini dilakukan perancangan dan implementasi jaringan SDN menggunakan ONOS *controller* yang dipasang pada Raspberry-Pi 3 dengan aplikasi *Virtual Private LAN Service* (VPLS). Perangkat jaringan yang digunakan terdiri dari 3 switch *OpenFlow*, 4 Laptop, dan 1 Raspberry-Pi 3 sebagai *controller*. Penelitian ini diharapkan dapat memaksimalkan QoS (*throughput, delay, jitter, packet loss*) berdasarkan standar *Telecommunications and Internet Protocol Harmonization Over Network* (TIPHON).

### 1.2 Tujuan

Adapun tujuan proyek akhir ini adalah sebagai berikut :

- a. Dapat mengimplementasikan jaringan *Software Defined Network* (SDN) dengan menggunakan ONOS *Controller* berbasis Raspberry-Pi 3.
- b. Dapat menerapkan *Virtual Private LAN Service* (VPLS) pada jaringan SDN.
- c. Dapat melakukan pengukuran parameter kinerja berdasarkan standar *Quality of Service* (QoS) TIPHON pada jaringan SDN.

### 1.3 Rumusan Masalah

Adapun rumusan masalah yang akan di bahas dari proyek akhir ini adalah :

- a. Bagaimana mengimplementasikan jaringan *Software Defined Network* (SDN) menggunakan ONOS *Controller* berbasis Raspberry-Pi 3?
- b. Bagaimana menerapkan *Virtual Private LAN Service* (VPLS) pada jaringan SDN ?
- c. Bagaimana melakukan pengukuran parameter kinerja berdasarkan standar *Quality of Service* (QoS) TIPHON pada jaringan SDN ?

### 1.4 Batasan Masalah

Batasan masalah yang akan di bahas pada proyek akhir ini adalah sebagai berikut :

1. Menggunakan ONOS *controller* sebagai *control plane*.
2. Menggunakan Raspberry-Pi sebagai *controller* dengan sistem operasi Raspbian.
3. Perangkat yang digunakan untuk implementasi jaringan ini adalah 4 buah laptop, 3 switch *OpenFlow*, 2 converter USB to Ethernet dan 1 buah *controller* (Raspberry-Pi 3).

### 1.5 Metodologi

Metodologi yang dilakukan pada proyek akhir ini adalah sebagai berikut:

1. Studi literatur  
Pencarian informasi dan pendalaman materi melalui berbagai sumber seperti buku, jurnal, internet untuk menunjang selesainya Proyek Akhir ini.
2. Perancangan Sistem  
Melakukan perancangan sistem sesuai dengan model sistem yang diinginkan.
3. Implementasi Perancangan  
Mengimplementasi Proyek Akhir sesuai dengan perancangan jaringan yang diinginkan.
4. Analisa sistem  
Mengamati dan menguji hasil dari sistem yang dirancang serta menyimpulkan dan menganalisa masalah yang ada.
5. Kesimpulan  
Menyimpulkan hasil dari seluruh tahapan yang telah dilakukan dalam proses pengerjaan Proyek Akhir dengan berbagai masukan dan saran dari dosen pembimbing maka dapat diambil kesimpulan dari hasil yang telah dilakukan.

## 2. DASAR TEORI

### 2.1 Software Defined Network

*Software Defined Network* (SDN) adalah konsep baru dalam mengontrol, mengimplementasi serta mengelola suatu jaringan yang mendukung kebutuhan dan inovasi di bidang telekomunikasi yg semakin lama semakin berkembang dan kompleks. Konsep dasar dari SDN yaitu dengan memisahkan antara *control plane* dan *data plane* [19]. SDN mampu membuat jaringan baik berskala kecil maupun berskala besar mampu terkontrol menggunakan 1 *controller* yang terpusat. Beberapa *controller* yang ada pada SDN diantaranya adalah POX, RYU, OpenDaylight, dan ONOS.

### 2.2 OpenFlow

*OpenFlow* adalah protokol pada SDN yang menghubungkan *controller* dengan perangkat *data plane* pada SDN. Posisinya berada di antara *controller* dan *forwarding (data plane)*. *OpenFlow* memungkinkan pengaturan *routing* dan pengiriman paket ketika melalui sebuah switch. Dalam sebuah jaringan, setiap switch hanya berfungsi meneruskan paket yang melalui suatu port tanpa mampu membedakan tipe protokol data yang dikirimkan. *OpenFlow* memungkinkan untuk mengakses dan memanipulasi *forwarding plane* secara langsung dari perangkat jaringan seperti switch dan router baik secara fisik maupun virtual[18]. Pada arsitektur SDN *protocol OpenFlow* terletak pada *Southbound API* yang menghubungkan antara *Control Layer* dengan *Infrastructure Layer*.

### 2.3 Open Network Operating System (ONOS)

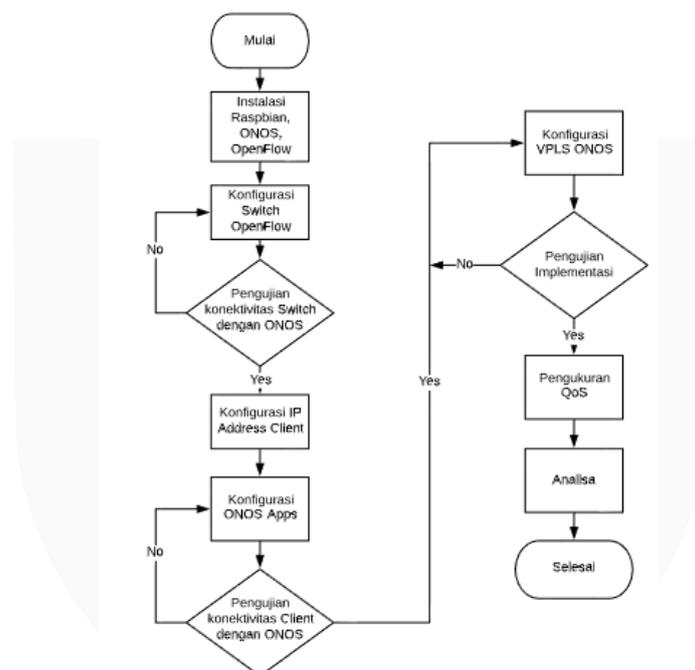
ONOS adalah sistem operasi jaringan SDN *open source* terkemuka untuk membangun solusi SDN generasi selanjutnya yang ditargetkan secara khusus di *Service Provider* dan *Mission Critical Network*. ONOS dibangun untuk memberikan ketersediaan tinggi (*High Availability*), *scale-out*, dan performansi kinerja jaringan yang dibutuhkan. *controlLer* ini berbasis java [3]. Dengan pengontrol *cloud* ONOS, pengguna dapat dengan mudah membuat aplikasi jaringan baru tanpa perlu mengubah sistem data plane. ONOS mendukung konfigurasi dan kontrol jaringan secara real-time, dan menghilangkan kebutuhan untuk menjalankan *routing* serta mengalihkan protokol kontrol di dalam jaringan. ONOS adalah proyek *open source* yang didistribusikan di bawah lisensi Apache 2.0 menggunakan bahasa pemrograman berbasis Java dan menggunakan Apache Karav untuk menghidupkan setiap fitur yang ada pada ONOS [10].

#### 2.4 Virtual Private LAN Service (VPLS)

VPLS adalah *multipoint* VPN layer 2 yang mengizinkan banyak daerah untuk dihubungkan dalam satu *single bridge domain* yang sama melalui jaringan *Internet Protocol* (IP). Seluruh daerah *client* dalam VPLS *instance* dapat seolah-olah berada pada satu jaringan LAN yang sama walaupun sebenarnya terpisah secara geografis. VPLS menggunakan *interface* Ethernet ke *client*-nya. Layanan VPLS disediakan untuk operator, penyedia layanan, dan perusahaan besar yang membutuhkan ketersediaan layanan Ethernet dengan kinerja tinggi dan jaminan *Quality of Service* (QoS) [12].

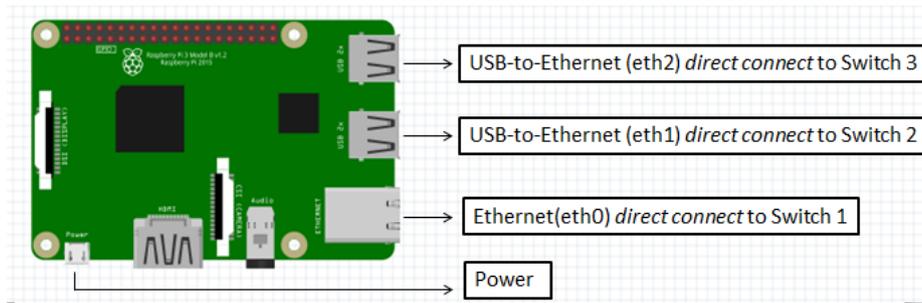
### 3. PERANCANGAN SISTEM

#### 3.1 Metode Penelitian



Gambar 3. 1 Flowchart Metode Penelitian

Pada Gambar 3.1 diatas menunjukkan *flowchart* untuk perancangan sistem pada Proyek Akhir, dimulai dari *installasi* Sistem Operasi Raspbian pada Raspberry-Pi 3 yang digunakan sebagai perangkat *controller* pada jaringan *Software Defined Network* (SDN). Tahap selanjutnya melakukan *installasi* ONOS sebagai *controller* yang digunakan pada Raspberry-Pi 3. Langkah berikutnya adalah melakukan instalasi *OpenFlow* pada setiap switch yang digunakan pada jaringan SDN lalu menghubungkan switch – switch tersebut dengan Raspberry-Pi 3 menggunakan ethernet.

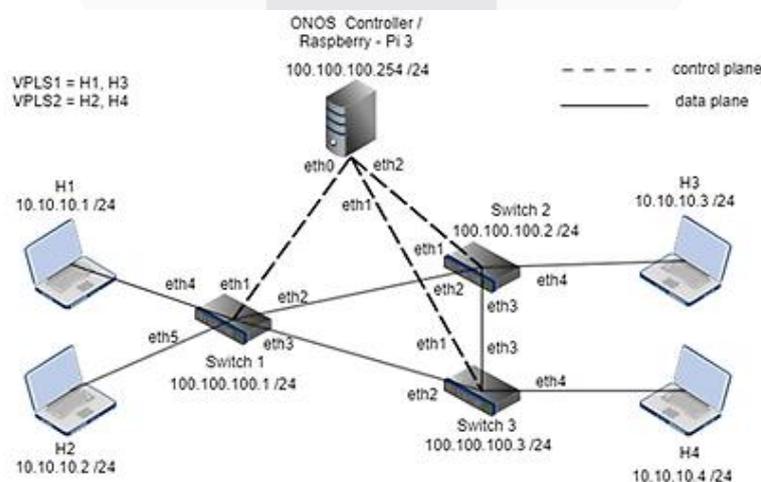


Gambar 3. 2 Rangkaian Skematik Rasperry-Pi 3

Pada Gambar 3.2 menunjukkan rangkaian skematik dari Rasperry-Pi 3 yang digunakan, terlihat bahwa Rasperry-Pi 3 yang digunakan pada Proyek Akhir ini menggunakan 2 buah *converter* USB-to-Ethernet agar dapat menghubungkan Rasperry-Pi 3 dengan switch *OpenFlow* yaitu Switch 2 dan Switch 3 serta 1 port Ethernet untuk menghubungkan ke Switch 1, seluruh port tersebut akan dihubungkan dengan Switch menggunakan kabel UTP. Untuk dapat menghubungkan seluruh switch *OpenFlow* dengan *controller* ini dilakukanlah konfigurasi *bridge* pada sisi *controller*. Tahap berikutnya melakukan konfigurasi *OpenFlow* pada switch sehingga setiap switch yang terhubung dengan *controller* Rasperry-Pi 3 dapat terdeteksi pada topologi ONOS yang artinya ONOS sebagai *controller* sudah dapat mengontrol jaringan SDN yang terbentuk melalui switch – switch tersebut. Langkah selanjutnya menghubungkan setiap switch dengan PC sesuai dengan topologi perencanaan Proyek Akhir, kemudian melakukan konfigurasi IP Address pada setiap PC menjadi seakan – akan dalam satu jaringan lokal dengan *prefix network* yang sama(/24).

Setiap PC yang terhubung dengan port yang dikonfigurasi dengan switch *OpenFlow* maka dapat dikenali oleh *controller* ONOS. Langkah selanjutnya melakukan konfigurasi ONOS Apps, pada Proyek Akhir ini dilakukan terdapat 3 aplikasi yang diaktifkan pada *controller* ONOS yaitu *OpenFlow*, *Forwarding*, dan VPLS. Langkah berikutnya melakukan pengujian tes PING antar PC. Tahap selanjutnya adalah melakukan konfigurasi VPLS pada terminal ONOS untuk memisahkan menjadi 2 VPLS ID dalam hal ini dilakukan menggunakan VPLS1 dan VPLS2, sehingga PC yang memiliki VPLS ID yang berbeda maka tidak bisa saling berkomunikasi. Sebaliknya, PC yang memiliki VPLS ID yang sama maka dapat saling berkomunikasi. Tahap berikutnya melakukan pengujian dengan melakukan tes PING baik antar PC yang memiliki VPLS ID yang sama maupun antar PC yang memiliki VPLS ID yang berbeda. Tahap terakhir terakhir adalah mengukur QoS dengan standarisasi TIPHON. Tahap pengujian dilakukan menggunakan layanan VoIP untuk melakukan pengukuran QoS mulai dari *throughput*, *delay*, *packet loss*, *jitter*. Dengan memiliki 2 VPLS ID, maka dilakukanlah pengujian menggunakan 1 PC sebagai *server* VoIP dan 3 PC lainnya akan digunakan sebagai *client*-nya, dalam pengujian setiap PC yang memiliki VPLS ID yang sama melakukan komunikasi VoIP, *software* yang digunakan yaitu Asterisk, Zoiper, dan iperf dalam pengujianya. Langkah terakhir adalah mengukur dan menganalisa hasilnya menggunakan Wireshark.

### 3.2 Topologi Jaringan



Gambar 3. 3 Topologi Jaringan

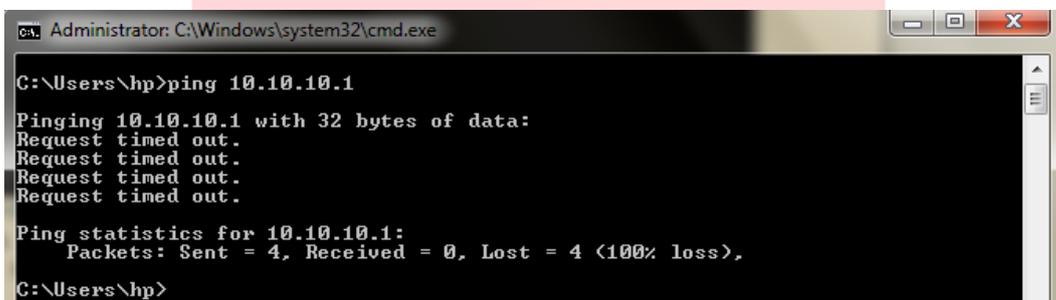
Pada tahap berikut dilakukan perancangan topologi jaringan yang diimplementasikan pada Proyek Akhir. Perangkat yang digunakan yaitu 1 Raspberry-Pi 3 yang digunakan sebagai *controller* ONOS, 2 buah *converter* USB-To-Ethernet, 3 buah switch *OpenFlow*, 10 kabel UTP, 4 buah PC. Berikut ini adalah rangkuman perangkat yang digunakan :

- |                                     |                    |
|-------------------------------------|--------------------|
| a. <i>Switch OpenFlow</i>           | : 3 buah           |
| b. Kabel UTP                        | : 10 buah          |
| c. <i>Converter</i> USB-to-Ethernet | : 2 buah           |
| d. <i>Controller</i>                | : 1 buah           |
| e. PC                               | : 4 buah           |
| f. Software pengukuran QoS          | : Wireshark, Iperf |

## 4. HASIL PENGUJIAN

### 4.1 Pengujian VPLS

Pada pengujian VPLS ini dilakukan dengan cara melakukan PING antar host baik yang memiliki VPLS ID yang sama maupun antar host yang memiliki VPLS ID yang berbeda. Jika belum dilakukan konfigurasi VPLS, otomatis seluruh host dapat berkomunikasi. Pada Gambar 4.1 menunjukkan bahwa PING antara h2 dengan h1 menghasilkan Request Time Out karena h2 dan h1 memiliki VPLS ID yang berbeda, sedangkan pada Gambar 4.2 PING antara h2 dengan h4 menghasilkan Reply TTL yang artinya h2 dan h4 dapat berkomunikasi, karena h2 dan h4 masuk dalam VPLS ID yang sama.



```

Administrator: C:\Windows\system32\cmd.exe

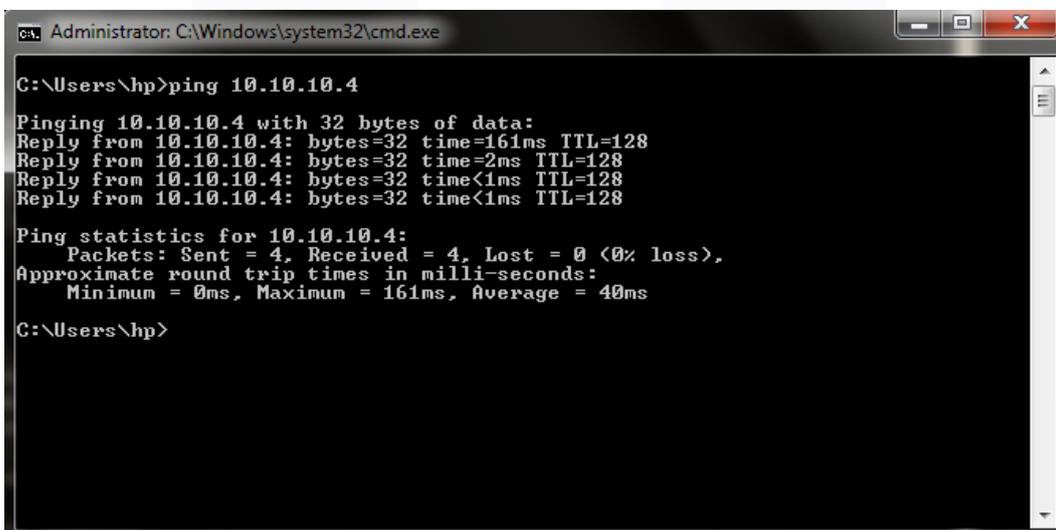
C:\Users\hp>ping 10.10.10.1

Pinging 10.10.10.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.10.10.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\hp>
  
```

Gambar 4. 1 PING Antara Host 2 Dengan Host 1



```

Administrator: C:\Windows\system32\cmd.exe

C:\Users\hp>ping 10.10.10.4

Pinging 10.10.10.4 with 32 bytes of data:
Reply from 10.10.10.4: bytes=32 time=161ms TTL=128
Reply from 10.10.10.4: bytes=32 time=2ms TTL=128
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128
Reply from 10.10.10.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.10.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 161ms, Average = 40ms

C:\Users\hp>
  
```

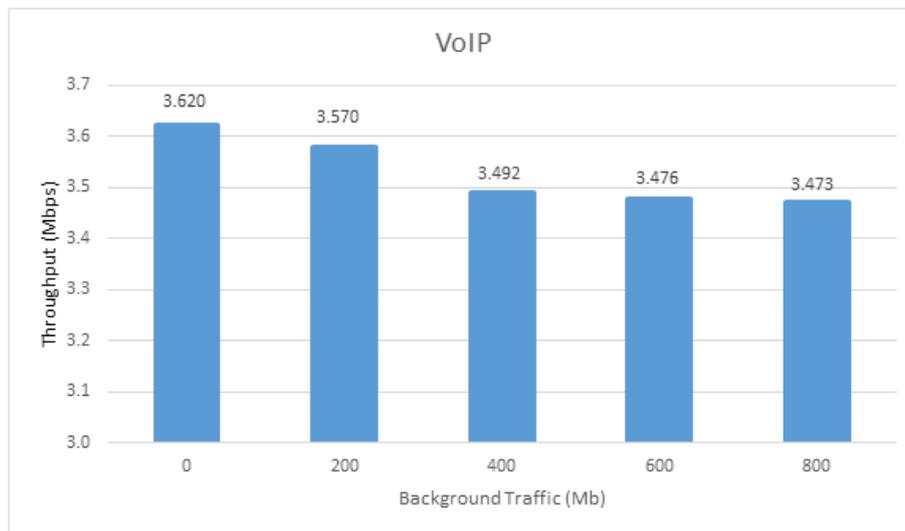
Gambar 4. 2 PING Antara Host 2 Dengan Host 4

### 4.2 Analisa Pengujian Performansi (VoIP)

Pada pengukuran performansi jaringan Proyek Akhir ini menerapkan layanan VPLS pada jaringan SDN dengan menggunakan komunikasi VoIP dengan iperf sebagai pembangkit *traffic*, Asterisk sebagai VoIP server, Zoiper sebagai *software client* VoIPnya serta Wireshark untuk mengukur dan menganalisa QoS-nya. Pengukuran dilakukan dengan *background traffic* 200Mb, 400Mb, 600Mb, dan 800Mb dengan protocol UDP. Adapun parameter – parameter QoS yang diukur, sebagai berikut:

- a. *Throughput*

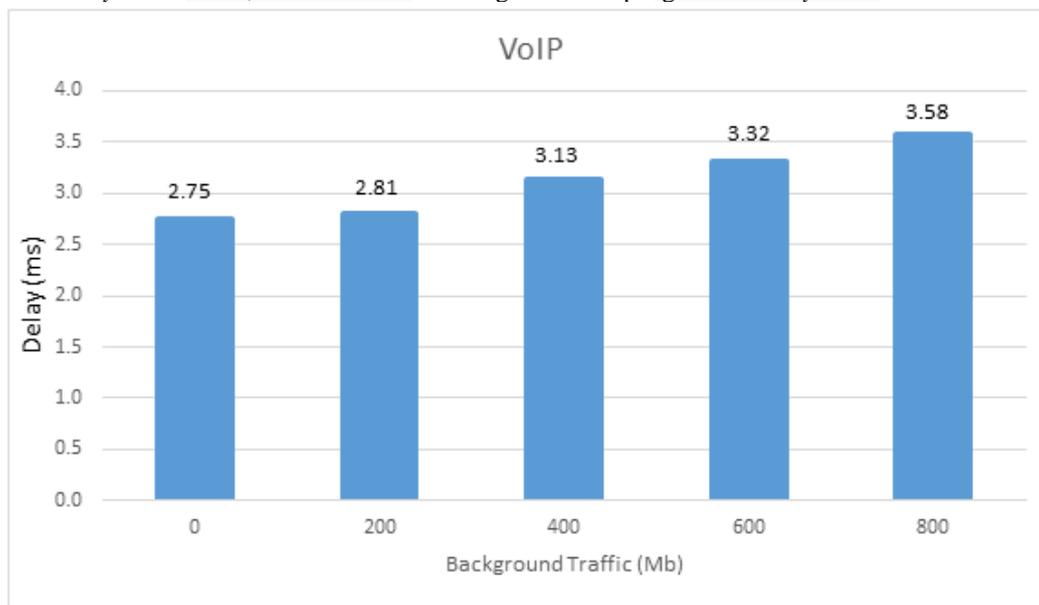
Pengukuran yang pertama adalah pengukuran *throughput*, pengukuran *throughput* dilakukan untuk mengetahui kecepatan sebenarnya jaringan pada saat melakukan pengiriman data. Pengukuran dilakukan dengan melakukan panggilan antara h1 dengan h3 yang sudah saling terhubung dan dapat berkomunikasi melalui layanan VPLS yang sudah dibangun pada jaringan SDN. Pada Gambar 4.3 menunjukkan bahwa *throughput* yang dihasilkan pada pengukuran ini masih sesuai dengan standarisasi TIPHON dimana nilai *throughput* tanpa *background traffic* mencapai 3,62 Mbps dan nilai *throughput* pada *background traffic* tertinggi mencapai 3,473 Mbps. Rata – rata dari keseluruhan *throughput* ini adalah 3,528 Mbps. Berikut adalah grafik hasil pengukuran *throughput* pada VOIP.



Gambar 4. 3 Grafik Throughput VoIP

b. *Delay*

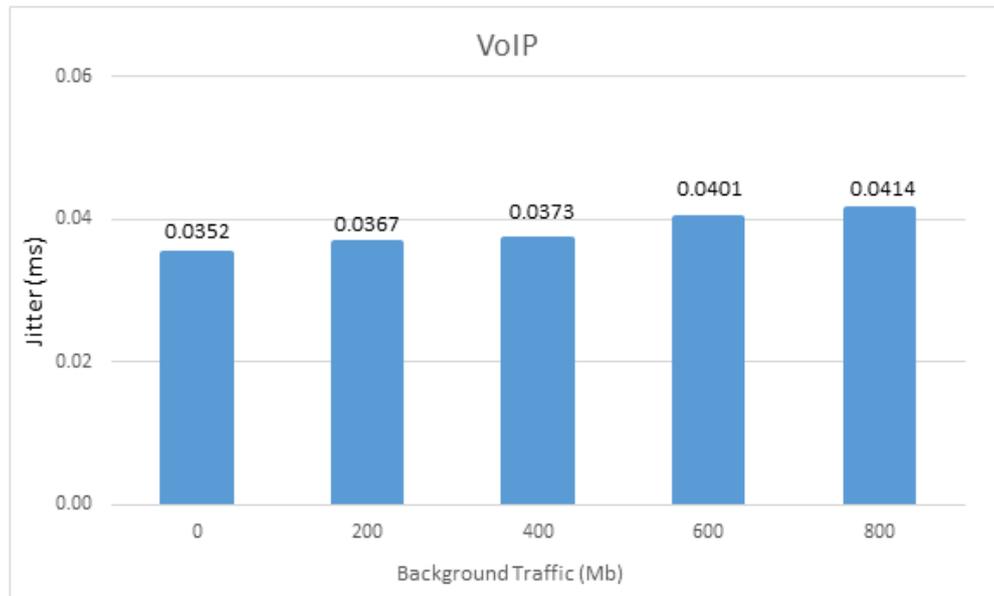
Pada Gambar 4.4 menunjukkan hasil *delay* dari pengukuran tanpa *background traffic* yaitu senilai 2,75 ms dan pengukuran dengan *background traffic* tertinggi senilai 3,58 ms. Hasil ini menunjukkan bahwa *delay* yang ditunjukkan pada pengukuran ini masih sesuai dengan standarisasi TIPHON. Rata – rata dari keseluruhan pengukuran *delay* ini adalah 3,11 ms. Berikut adalah grafik hasil pengukuran *delay*.



Gambar 4. 4 Grafik Delay VoIP

c. *Jitter*

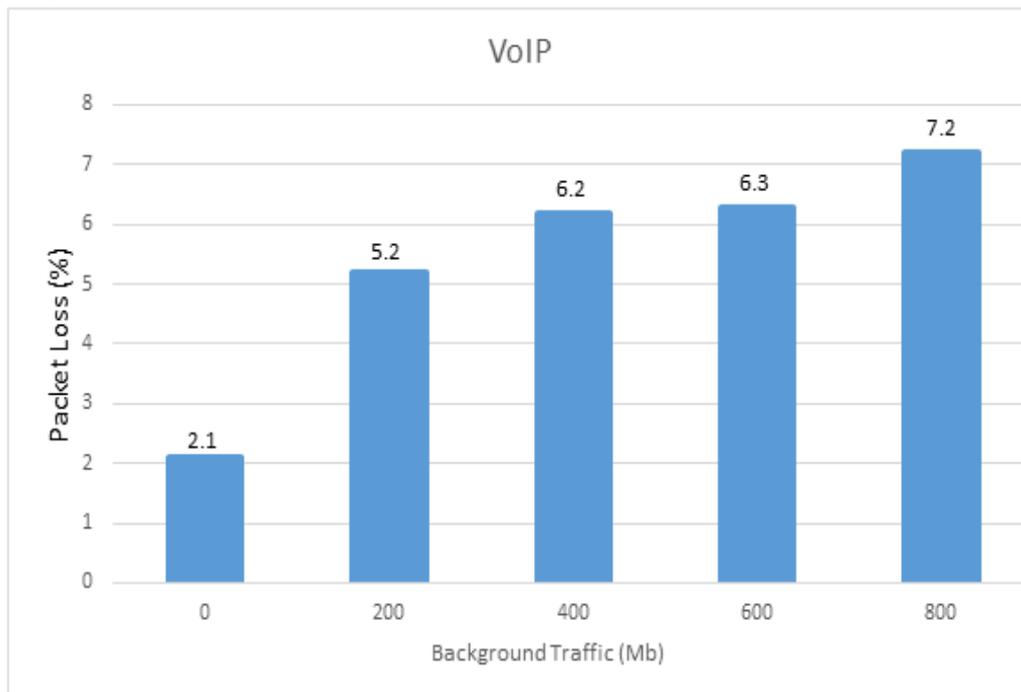
Pada Gambar 4.5 menunjukkan bahwa nilai *jitter* pada pengukuran ini masih sesuai dengan standarisasi TIPHON, pada pengukuran ini semakin tinggi nilai *background traffic* maka nilai *jitter* juga akan semakin meningkat, nilai *jitter* tanpa *background traffic* menunjukkan nilai 0,0352 ms sedangkan nilai *jitter* pada *background traffic* tertinggi bernilai 0,0414 ms. Rata – rata dari keseluruhan pengukuran *jitter* ini adalah 0,0381 ms. Berikut adalah grafik hasil pengukuran *jitter*.



Gambar 4. 5 Grafik Jitter VoIP

d. *Packet Loss*

Pada Gambar 4.6 menunjukkan bahwa nilai *packet loss* dari pengukuran ini masih sesuai dengan standarisasi TIPHON, dimana pada pengukuran ini nilai *packet loss* pada pengukuran tanpa *background traffic* senilai 2.1 % dan nilai *packet loss* pada pengukuran *background traffic* tertinggi adalah 7.2%. Berikut adalah grafik hasil pengukuran *packet loss*.



Gambar 4. 6 Grafik *Packet Loss* VoIP

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil Proyek Akhir yang telah dilakukan, maka kesimpulannya adalah sebagai berikut :

- a. Implementasi VPLS pada jaringan SDN yang memisahkan *control plane* dengan *data plane* dapat dilakukan dengan menggunakan *controller* ONOS yang diaplikasikan pada Raspberry-Pi 3.
- b. Hasil pengujian performansi implementasi VPLS pada jaringan SDN dengan menggunakan ONOS *controller* yang diaplikasikan pada Raspberry-Pi 3 menunjukkan bahwa nilai dari keempat parameter QoS masih sesuai dengan nilai standar TIPHON yang baik.
- c. Pada Proyek Akhir ini dilakukan perbandingan hasil pengukuran QoS dengan *background traffic* 200Mb, 400Mb, 600Mb, 800Mb untuk mengetahui batas kinerja jaringan terhadap beban *traffic*. Dan hasilnya adalah pada pengukuran dengan beban *traffic* tertinggi, nilai QoS masih menunjukkan nilai yang baik sesuai dengan standar TIPHON.
- d. Implementasi VPLS dapat dilakukan pada jaringan SDN menggunakan ONOS *controller* untuk membagi jaringan *private* antar Host.

### 5.2 Saran

Saran yang dapat diusulkan pada Proyek Akhir ini adalah

- a. Mengembangkan jaringan SDN menggunakan *controller* ONOS berbasis Raspberry-Pi 3.
- b. Mengimplementasikan aplikasi – aplikasi lain pada ONOS *controller*.
- c. Mengembangkan jaringan SDN pada sisi keamanan *controller*.
- d. Mengembangkan layanan VPLS pada jaringan SDN dengan menggunakan *controller* lain.
- e. Mengembangkan pengujian dan pengukuran QoS pada SDN menggunakan standar QoS yang lain.

### DAFTAR PUSTAKA

- [1] A. A. Prastyo, “Konfigurasi VPLS (Virtual Private LAN Service) ONOS 1.11.1 di GNS3,” 2017. [Online]. Available: <https://www.routecloud.net/blog/konfigurasi-vpls-virtual-private-lan-service-onos-1-11-1-di-gns3/>. [diakses pada 19 Desember 2019].
- [2] A. Triputranada, “Monitoring Suhu Ruangan Server Berbasis Raspberry-Pi Menggunakan Sensor Suhu DHT11,” J. Chem. Inf. Model., 2016.
- [3] D. S. R. Sari, R. Munadi, “Analisis Performansi Segment Routing Pada Software Defined Network Menggunakan Kontroler ONOS,” vol. 6, no. 2, pp. 3469–3476, 2019.
- [4] F. Ramadhan, R. Primananda, and W. Yahya, “Implementasi Routing Berbasis Algoritme Dijkstra Pada Software Defined Networking Menggunakan Kontroler Open Network Operating System,” J. Pengemb. Teknol. Inf. dan Ilmu Komputer; Vol 2 No 7, vol. 2, no. 7, pp. 2531–2541, 2017.
- [5] F. T. Elektro, “Tim Teaching MKP SDN Mininet,” 2016.
- [6] Linux Foundation. “Production Quality, Multilayer Open Virtual Switch,” 2016. [Online]. Available: <https://www.openvswitch.org/>. [diakses pada 19 Desember 2019].
- [7] M. Mulfi, S. N. Hertiana, I. Wahidfa. “Routing Connection VPLS Melalui Ethernet Over MPLS”. 2009.
- [8] M. N. Yaqin, R. Tulloh, I. D. Irawati, “Perancangan dan Implementasi Protocol Routing EBGp Pada Software Defined Network Menggunakan ONOS Controller,” 2019.
- [9] M. Nuruzzamanirridha, I. D. Irawati, Y. S. Hariyani, “Implementasi Jaringan Komputer Berbasis Software Defined Network Menggunakan Ryu Controller dan OpenvSwitch,” e-Proceeding of Applied Science : vol. 2, No.2 ,2016.
- [10] M. W. Putra, E. S. Pramukantoro, and W. Yahya, “Analisis Perbandingan Performansi Kontroler Floodlight , Maestro , RYU , POX Dan ONOS Dalam Arsitektur Software Defined Network ( SDN ),” J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 2, no. 10, pp. 3779–3787, 2018.
- [11] Open Networking Foundation, “Software-Defined Networking: The New Norm for Networks.
- [12] P. Monika, R. M. Negara, D. D. Sanjoyo, F. T. Elektro, U. Telkom, and L. I. Monitor, “Analisis Performansi Jaringan Software Defined Network Menggunakan Metode Intent Monitor And Reroute ( IMR ) Pada Controller ONOS,” 2019.

- [13] R. Munadi and R. Mayasari, "Implementation and Performance Analysis of Virtual Private LAN Service-TE Tunnel Network with OpenIMSCore as Multimedia Service Server," vol. 3, no. 3, pp. 4641–4648, 2016.
- [14] R. Muchlisin. "Pengertian, Layanan dan Parameter Quality of Service (QoS)". 2019. [Online]. Available : <https://www.kajianpustaka.com/2019/05/pengertian-layanan-dan-parameter-quality-of-service-qos.html>. [diakses pada 25 Juni 2020].
- [15] R. Tulloh, R. M. Negara, and A. N. Hidayat, "Simulasi Virtual Local Area Network. (VLAN) Berbasis Software Defined Network (SDN) Menggunakan POX Controller," J. INFOTEL - Inform. Telekomun. Elektron., vol. 7, no. 2, p. 129, 2015.
- [16] Raspberry Pi Foundation. "Trademark rules and brand guidelines Raspberry-Pi," 2013. [Online]. Available: <https://www.raspberrypi.org/trademark-rules/>. [diakses pada 19 Desember 2019].
- [17] T. I. Bayu and E. E. Tahan, "Software Defined Network (Sdn) Simulation Concept Using Raspberry Pi," J. Terap. Teknol. Inf., vol. 2, no. 2, pp. 1–11, 2018.
- [18] Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON), "Tr 101 329," Etsi, vol. 1, no. General aspects of Quality of Service (QoS), pp. 1–37, 1999.
- [19] Ummah, "Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking," Indones. J. Comput., vol. 1, no. 1, pp. 95–106, 2016.
- [20] V. Monita, I. D. Irawati, R. Tulloh, "Comparation of Routing Protocol Performance on Multimedia Service On Software Defined Network," Bulletin of Electrical Engineering and Informatics vol. 9, no. 4, pp. 1612 – 1619, 2020.