

# Pengujian QoS Pada Implementasi SDN Berbasis Mininet dan OpenDaylight Menggunakan Topologi Tree

I Putu Agus Eka Pratama<sup>[1]\*</sup>, Kevin Christopher Bakkara<sup>[2]</sup>

Program Studi Teknologi Informasi, Fakultas Teknik<sup>[1], [2]</sup>  
Universitas Udayana

Jl. Raya Kampus Unud, Bukit Jimbaran, Bali, Indonesia  
eka.pratama@unud.ac.id<sup>[1]</sup>, kevinbakkara18@gmail.com<sup>[2]</sup>

**Abstract**— The development of information technology and computer network from time to time is increasing along with the increase in user needs for both from the business, education, industrial, to data security side. Data of network traffic that is getting denser in communication and data exchange between users on computer networks can become a problem when using conventional computer network technology. For that, it needs a new technology that is implemented in computer networks, along with the measurement of Quality of Service (QoS) in it. Software-Defined Networking (SDN) is a solution for this, where the stages of network design, management and implementation, separate the data plane and the control plane. In this research, the implementation of SDN was carried out in the form of a simulation using both of Mininet and OpenDaylight with a Tree Topology, then the QoS measurements were carried out in it. The results of testing and measuring QoS on SDN simulations with Tree topology using Mininet and OpenDaylight, showed a Jitter value of 0.425 ms, a Packet Loss value of 0.266%, a Bandwidth value of 9.3925 Mbps, a UDP Throughput value of 2.348 bits/sec, and a TCP Throughput value of 2.335 bits/sec.

**Keywords**— Computer network, Mininet, OpenDaylight, Software Defined Network (SDN), Quality of Service (QoS)

**Abstrak**— Perkembangan teknologi informasi dan jaringan komputer dari waktu ke waktu, semakin meningkat seiring dengan peningkatan kebutuhan pengguna terhadap keduanya dari sisi bisnis, pendidikan, industri, hingga keamanan data. Trafik jaringan data yang semakin padat di dalam komunikasi dan pertukaran data antar pengguna dalam jaringan komputer, dapat menjadi sebuah masalah apabila menggunakan teknologi jaringan komputer konvensional. Untuk itu, diperlukan sebuah teknologi terbaru yang diimplementasikan di dalam jaringan komputer, beserta dengan pengukuran *Quality of Service (QoS)* di dalamnya. *Software Defined Networking (SDN)* adalah solusi untuk kebutuhan akan hal ini, di mana tahapan desain, pengelolaan, dan implementasi jaringan, memisahkan antara *data plane* dan *control plane*. Di dalam penelitian ini, implementasi dari SDN dilakukan dalam bentuk simulasi menggunakan *Mininet* dan *OpenDaylight* dengan topologi *Tree*, untuk kemudian dilakukan pengukuran QoS di dalamnya. Hasil pengujian dan pengukuran QoS pada simulasi SDN dengan topologi *Tree* menggunakan *Mininet* dan *OpenDaylight*,

menunjukkan nilai Jitter sebesar 0.425 ms, Packet Loss sebesar 0.266%, Bandwidth sebesar 9,3925 Mbps, Throughput UDP sebesar 2,348 bit/sec, dan Throughput TCP sebesar 2,335 bit/sec.

**Kata Kunci**—jaringan komputer, Mininet, OpenDaylight, Software Defined Network (SDN), Quality of Service (QoS)

## I. PENDAHULUAN

Teknologi informasi dan jaringan komputer terus berkembang dari waktu ke waktu. Perkembangan ini terjadi seiring dengan makin meningkatnya kebutuhan manusia akan teknologi informasi dan jaringan komputer di berbagai bidang kehidupan, antara lain di bidang bisnis, pendidikan, industri, hingga keamanan data. Adanya perkembangan teknologi informasi dan jaringan komputer, salah satunya menghasilkan peningkatan trafik jaringan.

Peningkatan trafik jaringan, disebabkan oleh makin banyaknya aplikasi dan layanan berbasis teknologi informasi di dalam jaringan komputer. Layanan-layanan ini dapat diakses dan digunakan oleh pengguna dari perangkat desktop dan mobile. Dalam hal ini, dari sudut pandang jaringan komputer, *Network Layer* menjadi tulang punggung keberlangsungan lalu lintas paket data dan trafik jaringan yang makin padat tersebut [1].

Trafik jaringan yang makin meningkat pesat, menjadikan jaringan komputer semakin padat. Hal ini dapat menyebabkan sejumlah permasalahan terhadap penyediaan aplikasi dan layanan online berbasis teknologi informasi yang disediakan pada jaringan komputer, apabila masih menggunakan model jaringan komputer konvensional tanpa disertai dengan upaya pengukuran kualitas layanan di dalamnya. Untuk itu, diperlukan sebuah implementasi teknologi terbaru di dalam jaringan komputer untuk menyelesaikan permasalahan ini.

*Software Defined Networking (SDN)* serta pengukuran *Quality of Service (QoS)* diusulkan sebagai solusi atas permasalahan ini. SDN merupakan sebuah konsep pendekatan baru untuk mendesain, mengelola, dan mengimplementasikan arsitektur jaringan komputer berbasis teknologi *Cloud Computing* (khususnya jenis layanan *IaaS Cloud*) [2], yang memisahkan antara sistem kontrol (*control plane*) dan sistem

forwarding (data plane) pada perangkat keras jaringan [3]. SDN memiliki arsitektur yang memungkinkan jaringan komputer untuk kebutuhan aplikasi maupun kebutuhan pengguna, menyederhanakan manajemen dan memungkinkan skalabilitas jaringan, yang diwujudkan melalui implementasi sederhana dari penambahan komponen dan layanan jaringan [4]. Struktur dari SDN bergantung kepada controller yang dipilih, sehingga memudahkan pengiriman paket data secara lebih cepat. Selain itu, dengan konsep jaringan berbasis software, network administrator dapat mengelola jaringan, mengintegrasikan teknologi baru, dan meningkatkan paket data secara lebih cepat dan lebih baik.

Di dalam penelitian ini, dilakukan simulasi SDN menggunakan open source software bernama Mininet [5] dan controller OpenDaylight [6] dengan topologi Tree. Selanjutnya pada simulasi dilakukan pengukuran dan analisa terhadap keempat parameter pada Quality of Service (QoS), yang meliputi: throughput, packet loss, bandwidth, dan jitter [7], dengan memanfaatkan open source software Iperf [8]. Analisa terhadap Throughput dilakukan melalui port number 5566 (TCP) dan port number 5577 (UDP) pada Transport Layer.

Adapun tujuan yang ingin dicapai di dalam penelitian ini, yaitu: 1.) Mensimulasikan usulan perancangan jaringan komputer yang lebih baik dengan memanfaatkan teknologi SDN, 2.) Memperoleh nilai hasil pengukuran keempat parameter QoS pada simulasi SDN dengan menggunakan Topologi Tree.

Terdapat dua buah pertanyaan penelitian yang menjadi rumusan masalah pada penelitian ini, yaitu: 1.) Bagaimana teknis untuk mengimplementasikan simulasi SDN dengan menggunakan topologi Tree berbasis Mininet dan OpenDaylight? 2.) Bagaimana melakukan pengujian dan pengukuran terhadap nilai dari keempat parameter QoS pada

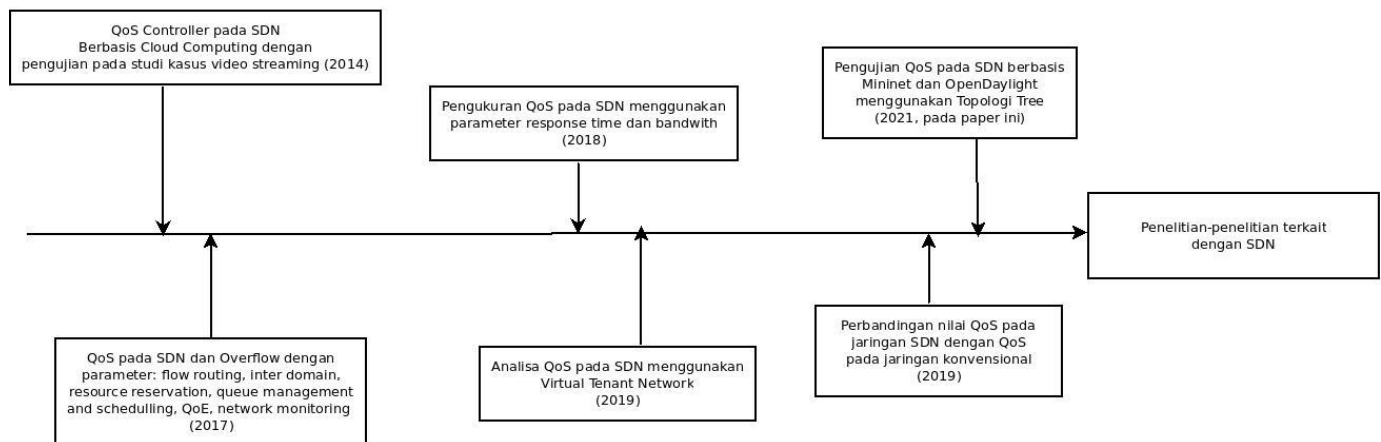
SDN yang disimulasikan dengan Topologi Tree?

Terdapat sejumlah penelitian yang melatar belakangi penelitian ini. Penelitian pertama oleh Pranawan untuk mengukur QoS pada SDN menggunakan metode Load Switching pada jaringan seluler untuk meningkatkan reliability [9]. Penelitian kedua dilakukan oleh Anam dan Adrian untuk menganalisa performansi SDN pada jaringan dengan menggunakan protokol Routing Open Shortest Path First (ROSPF) [10]. Penelitian ketiga dilakukan oleh Purnomo dan Arisandi untuk menganalisa QoS pada SDN dengan menggunakan Virtual Tenant Network (VTN) [11].

Penelitian keempat dilakukan oleh Karakus mengenai QoS pada SDN dan OpenFlow, yang meliputi: mekanisme multimedia flows routing, inter-domain routing, resource reservation, queue management and scheduling, Quality of Experience (QoE) aware, dan network monitoring [12]. Penelitian kelima dilakukan oleh Govindarajan, et al., untuk meneliti QoS Controller pada jaringan SDN Cloud Computing pada studi kasus layanan video streaming [13]. Penelitian keenam dilakukan oleh Nugroho untuk melakukan perbandingan nilai QoS pada jaringan SDN dengan jaringan komputer konvensional [14]. Terakhir, penelitian oleh Hosain, et al., melakukan pengukuran QoS pada SDN dengan menggunakan parameter response time dan bandwidth [15].

Penelitian-penelitian di atas turut membantu pemetaan penelitian yang telah dilakukan terkait dengan SDN. Kemudian, berdasarkan kepada penelitian-penelitian tersebut, terdapat sebuah gap penelitian yaitu belum adanya penelitian mengenai pengujian QoS pada simulasi SDN berbasis Mininet dan controller OpenDaylight pada Topologi Tree. Untuk itu, gap penelitian ini diambil untuk kemudian dikerjakan dan dipublikasikan melalui paper ini. Roadmap penelitian pada Gambar 1 menunjukkan hal tersebut.

### Roadmap Penelitian



Gambar 1. Roadmap Penelitian

## II. METODE PENELITIAN

### A. Alat dan Bahan

Pada penelitian ini digunakan sejumlah *hardware* dan *software* pendukung. Untuk kebutuhan *software* digunakan sistem operasi *Linux Ubuntu*, *Mininet*, *OpenDaylight*, *OpenJDK*, dan *Iperf*. Semua *software* yang digunakan di dalam penelitian ini berlisensi *open source*.

Untuk kebutuhan *hardware* digunakan komputer pribadi dengan spesifikasi: ASUS A456UR Intel(R) Core (TM) i5-6200U CPU @ 2.30GHz (4CPUs), memori DDR4 4096MB, HDD 1TB. Selain itu juga ditambahkan dengan koneksi internet untuk kemudahan penggunaan repositori *online*.

### B. Metodologi Penelitian

Metodologi yang digunakan di dalam penelitian ini adalah *Design Science Research Methodology* (DSRM) [16]. DSRM memiliki delapan langkah terurut yang digunakan di dalam langkah-langkah penelitian. Kedelapan langkah tersebut meliputi: 1.)Studi literatur terkait dengan topik penelitian dan state of the art terhadap sejumlah penelitian sebelumnya, 2.)Penentuan tujuan penelitian, 3.)Rumusan masalah yang disajikan ke dalam bentuk pertanyaan penelitian, 4.)Usulan solusi, 5.)Implementasi dari usulan solusi, 6.)Pengujian, 7.)Dokumentasi dan analisa hasil pengujian, dan 8.)Publikasi hasil penelitian.

### C. Flowchart Penelitian

Di dalam penelitian ini, terdapat urutan langkah-langkah penelitian yang dilakukan pada setiap tahapan. Urutan langkah-langkah yang dilakukan di dalam penelitian ini, disajikan ke dalam sebuah *flowchart* penelitian, sebagaimana ditampilkan pada Gambar 2.

Berdasarkan kepada *flowchart* penelitian yang ditampilkan pada Gambar 2. di atas, terdapat enam tahapan di dalam penelitian ini, yaitu: 1.)Tahap pendahuluan, 2.)Tahap implementasi, 3.)Tahap pengujian, 4.)Tahap analisa hasil pengujian, 5.)Tahap kesimpulan, dan 6.)Tahap akhir penelitian. Pada tahap pendahuluan, dilakukan penentuan studi kasus yang diangkat di dalam penelitian, dilanjutkan dengan melakukan identifikasi masalah, formulasi masalah, studi literatur untuk memperoleh *research gap*, untuk kemudian dilakukan pengajuan usulan solusi, diikuti dengan pengajuan pertanyaan penelitian (*research question*) dan hipotesa penelitian.

Pada tahap implementasi, dilakukan implementasi dalam bentuk instalasi dan konfigurasi pada sistem operasi Linux menggunakan *Mininet*, *OpenDaylight*, dan *Iperf*. Pada tahap pengujian, dilakukan pengujian QoS pada implementasi SDN disertai dengan pengamatan dan pencatatan hasil pengujian. Pada tahap analisa hasil pengujian, dilakukan analisa dari data-data hasil pengujian yang telah dilakukan. Pada tahap kesimpulan, dilakukan penarikan kesimpulan dari hasil analisa pengujian yang dilakukan, disertai dengan pembuktian hipotesa dan jawaban atas pertanyaan-pertanyaan penelitian yang diajukan. Pada tahap akhir dilakukan dokumentasi dan publikasi hasil penelitian ke dalam bentuk paper di jurnal ilmiah.

## III. HASIL DAN PEMBAHASAN

### A. Implementasi

Sebelum melakukan pengujian, terlebih dahulu dilakukan instalasi dan konfigurasi. Pertama dilakukan instalasi *OpenJDK* menggunakan perintah berikut:

```
sudo apt-get install openjdk-8jdk
```

Selanjutnya, konfigurasi *OpenJDK* dilakukan dengan menambahkan baris-baris berikut:

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
```

```
export PATH=$JAVA_HOME/bin:$PATH
```

Instalasi *Mininet* beserta dengan kelengkapan *Virtual Machine (VM)*, *Open vSwitch*, *OpenFlow*, *Wireshark*, dan *POX*, dilakukan dengan menggunakan perintah berikut:

```
sudo apt-get install git -y --fix-missing
```

```
git clone git://github.com/mininet/mininet
```

```
cd mininet
```

```
git tag
```

```
git checkout -b 2.2.1 2.2.1
```

```
mininet/util/install.sh -a
```

Instalasi *OpenDaylight* dilakukan dengan menggunakan perintah berikut:

```
wget
```

```
https://nexus.opendaylight.org/content/repositories/public/org.opendaylight/integration/distribution-karaf/0.6.1-Carbon/distribution-karaf-0.6.1-Carbon.zip
```

```
unzip distribution-karaf-0.6.1-Carbon.zip
```

```
cd /distribution-karaf-0.6.1-Carbon/bin
```

```
./karaf
```

```
feature:install odl-l2switch-switch-ui
```

```
feature:install odl-dlux-core
```

```
feature:install odl-dluxapps-nodes
```

```
feature:install odl-dluxapps-topology
```

```
feature:install odl-dluxapps-yangui
```

```
feature:install odl-dluxapps-yangvisualizer
```

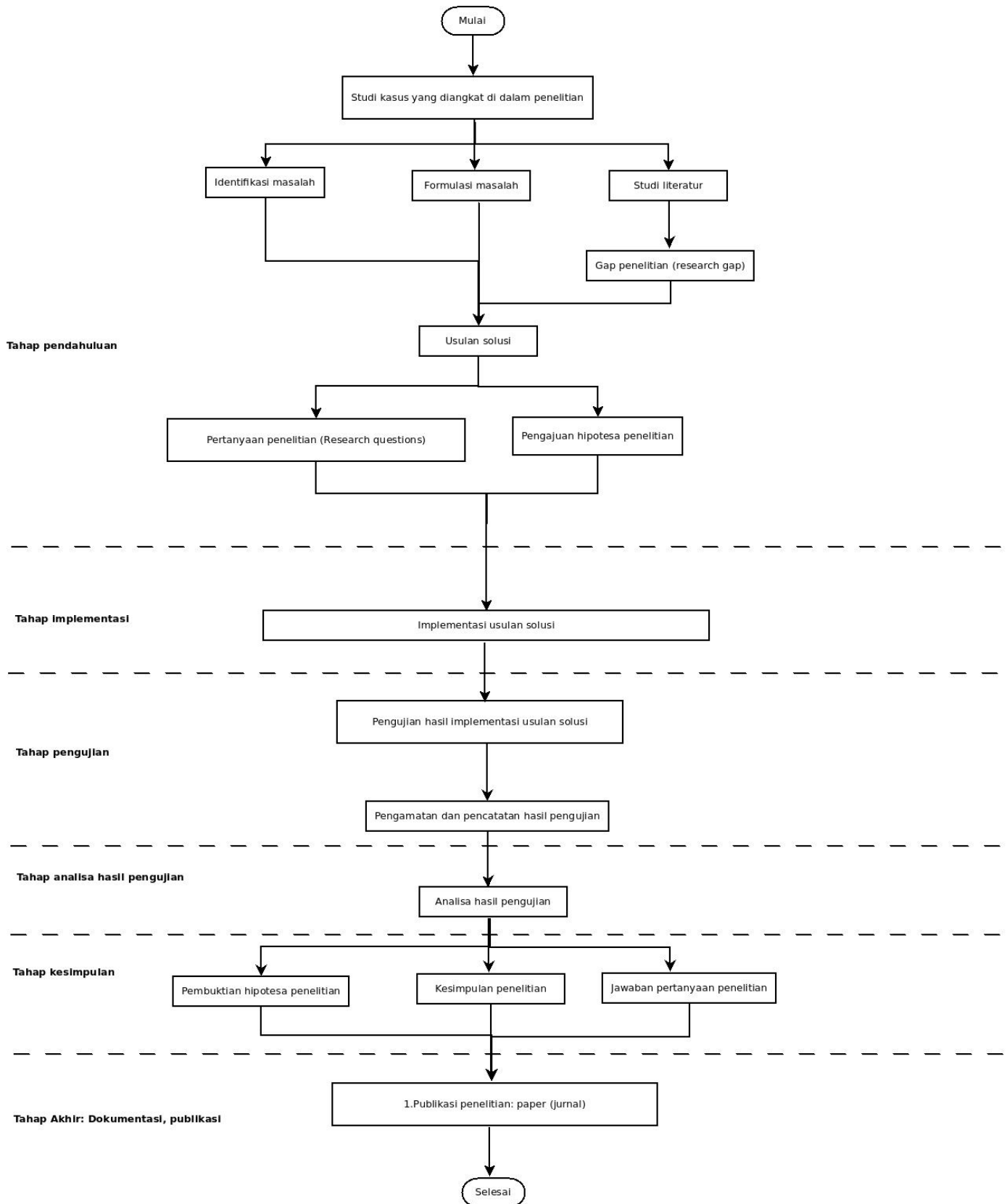
```
feature:install odl-dluxapps-yangman
```

Selanjutnya, instalasi *Iperf* yang akan digunakan pada pengukuran QoS, dilakukan dengan menggunakan perintah berikut:

```
sudo apt-get install iperf3
```

Pembuatan 16 *host* pada *Mininet* dengan range IP address dari 10.0.0.1 sampai 10.0.0.16 beserta dengan Topologi *Tree*, dilakukan dengan menggunakan perintah berikut:

```
sudo mn -controller=remote,ip=172.20.10.15 -topo=tree,4,2
```



Gambar 2. Flowchart Penelitian

Pengecekan terhadap koneksi ke setiap host yang terbentuk, dilakukan dengan menggunakan perintah berikut:

```
pingall
```

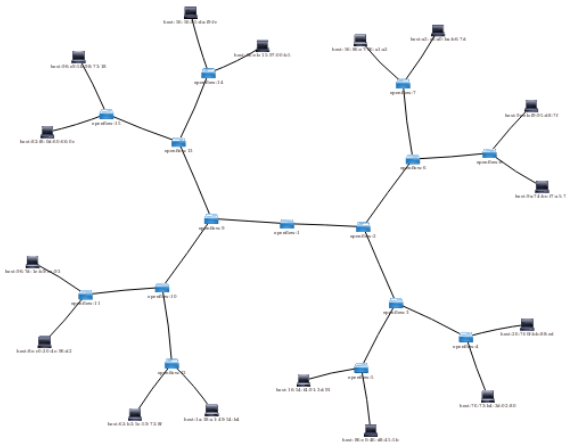
Hasil Topologi *Tree* pada Mininet dengan 16 host di dalamnya, ditampilkan pada Gambar 3.

```

kevin@kevin-X456UR: ~
*** Cleanup complete.
kevin@kevin-X456UR:~$ sudo mn --controller=remote,ip=172.20.10.15 --topo=tree,4,2
*** Creating network
*** Adding controller
Unable to contact the remote controller at 172.20.10.15:6633
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
*** Adding switches:
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15
*** Adding links:
(s1, s2) (s1, s9) (s2, s3) (s2, s6) (s3, s4) (s3, s5) (s4, h1) (s4, h2) (s5, h3) (s5, h4) (s6, s7) (s6, s8) (s7, h5) (s7, h6) (s8, h7) (s8, h8) (s9, s10) (s9, s13) (s10, s11) (s10, s12) (s11, h9) (s11, h10) (s12, h11) (s12, h12) (s13, s14) (s13, s15) (s14, h13) (s14, h14) (s15, h15) (s15, h16)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16
*** Starting controller
c0
*** Starting 15 switches
s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 ...
*** Starting CLI:
mininet> pingall
    
```

Gambar 3. Topologi Tree pada Mininet

Sedangkan visualisasi dari Topologi *Tree* pada Mininet di atas untuk 16 host di dalamnya, disajikan melalui *OpenDaylight* pada Gambar 4 berikut:



Gambar 4. Visualisasi melalui OpenDaylight

**B. Pengujian**

Pengujian QoS pada protokol TCP di *Transport Layer* antara *host 1* dan *host 2*, dilakukan dengan menggunakan perintah berikut:

```
Iperf3 -c 10.0.0.2 -t 20 -p 5566
iperf3 -s -I 1 -p 5566
```

Pengujian QoS pada protokol UDP di *Transport Layer* antara *host 1* dan *host 2*, dilakukan dengan menggunakan perintah berikut:

```
Iperf3 -c 10.0.0.2 -t 20 -p 5577 -u -b 10M
```

```
iperf3 -s -I 1 -p 5577
```

**C. Hasil Pengujian**

Berdasarkan kepada pengujian yang telah dilakukan dengan menggunakan *Iperf*, maka diperoleh nilai dari keempat parameter pada QoS (*Jitter*, *Packet Loss*, *Bandwith*, *Throughput*) pada protokol TCP dan UDP. Nilai-nilai hasil pengujian QoS tersebut, ditampilkan pada Tabel 1 berikut:

TABEL 1. HASIL PENGUJIAN QOS

Host	Jitter (ms)	Packet loss (%)	Bandwith (Mbps)	Throughput UDP (bit/s)	Throughput TCP (bit/s)
1	0.887	0.292	0,52	2.385	2.383
2	0.796	0.111	9,96	2.386	2.392
3	0.618	0.253	10,00	2.386	2.379
4	0.445	0.532	10,00	2.385	2.389
5	0.528	0.334	9,96	2.381	2.385
6	0.476	0.244	10,00	2.387	2.389
7	0.427	0.175	9,96	2.386	2.388
8	0.500	0.160	10,00	2.384	2.384
9	0.397	0.247	10,00	2.386	2.382
10	0.405	0.173	9,96	2.419	2.380
11	0.308	0.302	10,00	2.383	2.386
12	0.188	0.305	9,96	2.254	2.224
13	0.179	0.243	10,00	2.224	2.223
14	0.193	0.344	10,00	2.382	2.384
15	0.211	0.449	10,00	2.065	2.225
16	0.245	0.439	9,96	2.381	2.067
<b>Rata-Rata</b>	<b>0.425</b>	<b>0.266</b>	<b>9,3925</b>	<b>2.348</b>	<b>2.335</b>

**D. Pembahasan**

Berdasarkan kepada nilai-nilai dari hasil pengujian QoS pada SDN yang ditampilkan pada Tabel 1. di atas terhadap semua *host* dengan menggunakan *Iperf*, maka dapat dilakukan perhitungan nilai rata-rata dari setiap parameter QoS tersebut (*Jitter*, *Packet Loss*, *Bandwith*, *Throughput* UDP dan TCP). Perhitungan nilai rata-rata dilakukan dengan cara menjumlahkan semua nilai, untuk kemudian dibagi dengan 16 (16 adalah total jumlah *host*).

Dari hasil perhitungan tersebut, maka diperoleh nilai rata rata dari setiap parameter QoS, yaitu nilai rata-rata Jitter sebesar 0.425 ms, nilai rata-rata Packet Loss sebesar 0.266 %, nilai rata-rata Bandwith sebesar 9.3925 Mbps, nilai rata-rata Throughput (UDP) sebesar 2.348 bit/sec, serta nilai rata-rata Throughput (TCP) sebesar 2.335 bit/sec.

**IV. KESIMPULAN**

Berdasarkan kepada hasil pengujian dan pembahasan, maka dapat diperoleh 3 poin kesimpulan berikut: 1.)Teknis implementasi simulasi SDN menggunakan topologi Tree berbasis *Mininet* dan *OpenDaylight* pada sistem operasi Linux dapat dilakukan dengan baik, dengan mengikuti langkah-langkah instalasi dan konfigurasi sistem, 2.)Teknis pengujian dan pengukuran nilai dari keempat parameter QoS pada SDN yang disimulasikan dapat dilakukan dengan baik, menggunakan *Iperf* terhadap 16 host di dalamnya, 3.)Nilai rata-rata dari hasil pengujian QoS pada simulasi SDN menggunakan *Mininet* dan *OpenDaylight* untuk keempat elemen (*Jitter*, *Packet Loss*, *Bandwith*, *Throughput* UDP dan TCP) dilakukan dengan menjumlahkan nilai, untuk kemudian dibagi dengan total host, di mana nilai rata-rata menunjukkan bahwa secara umum bernilai baik. Dari ketiga poin kesimpulan ini, dapat dibuktikan bahwa pemisahan antara

sistem kontrol (*control plane*) dan sistem *forwarding (data plane)* beserta dengan peran dari controller pada SDN melalui simulasi yang dilakukan, mampu menjadikan pengiriman dan pertukaran paket data di dalam jaringan menjadi lebih cepat. Ke depannya, penelitian ini dapat dilanjutkan dengan melakukan pengujian di lingkungan SDN yang dikombinasikan ke dalam lingkup ekosistem Cloud Computing, sehingga terdapat tantangan di dalam melakukan pengukuran QoS pada lingkungan Cloud Computing.

#### UCAPAN TERIMA KASIH

Terima kasih penulis ucapkan kepada Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana serta kepada rekan-rekan komunitas Linux dan Open Source Indonesia selama proses penelitian ini, untuk diskusi terkait dokumentasi, pengembangan, dan teknis implementasi.

#### DAFTAR PUSTAKA

- [1] S. Riyadi, A.S. Budiman, "Perencanaan Pembangunan WAN Menggunakan EIGRP Dynamic Routing," Jurnal SISFOKOM (Sistem Informasi dan Komputer), Vol. 09, No. 03, 2020.
- [2] A. Siamak, W. Philipp, and Y. Ramin, "SDN-Based Cloud Computing Networking," 2013 15th International Conference on Transparent Optical Networks, 2013.
- [3] A. Al-Najjar, S. Layeghy, and M. Portmann, "Pushing SDN to the end-host, network load balancing using OpenFlow," 2016 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops. 2016.
- [4] Grgurevic, Ivan, K. Zvonko, and P. Anthon, "Simulation Analysis of Characteristics and Application of Software-Defined Networks," Jurnal. University of Zagreb, Zagreb, Croatia. 2015.
- [5] Mininet, "Mininet An Instant Virtual Network on Your Laptop (or Other PC)," Mininet Project Contributor, 2020. [Online]. Available: <https://mininet.org>. [Diakses: 8 April 2020]
- [6] OpenDaylight, "OpenDaylight a modular open platform for customizing and automating networks of any size and scale," OpenDaylight Project, 2020. [Online]. Available: <https://www.opendaylight.org> [Diakses: 10 Mei 2020]
- [7] ETSI, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON): General aspects of Quality of Service (QoS)," France. 2009.
- [8] IPERF, "Iperf The Ultimate Speed Test Tool for TCP, UDP and SCTP," IPERF Project, 2020. [Online]. Available: <https://iperf.fr>. [Diakses: 20 Juli 2020]
- [9] Y.B.A. Pranawa, R.M. Ijtihadie, dan W. Wibisono, "Implementasi Wireless Quality of Service dengan Metode Load Switching Jaringan Seluler Menggunakan Software Defined Network untuk Meningkatkan Network Reliability pada Jaringan Dinamis," Jurnal Teknik ITS, Vol.6, No.1, 2017.
- [10] K. Anam, R. Adrian, "Analisis Performa Jaringan Software Defined Network Berdasarkan Penggunaan Cost Pada Protokol Routing Open Shortest Path First," The 9th Conference on Information Technology and Electrical Engineering (CITEE 2017), 2017.
- [11] R. Purnomo, dan R. Arisandi, "Analisis QoS Dengan Virtual Tenant Network Pada Software Define Networking," Jurnal Teknologi Informatika dan Komputer, Vol.5, No.2, 2019.
- [12] M. Karakus, and A. Durrezi, "Quality of Service (QoS) in Software Defined Networking (SDN): A Survey," Journal of Network and Computer Applications, Vol. 80, No. 15, 2017.
- [13] K. Govindarajan, et al., "Realizing the Quality of Service (QoS) in Software-Defined Networking (SDN) based Cloud infrastructure," 2014 2nd International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 2014.
- [14] H.P. Nugroho, M. Irfan, A. Faruq, "Software Defined Networks: a Comparative Study and Quality of Services Evaluation," Scientifics Journal of Informatics, Vol. 6, No. 2, 2019.
- [15] M.A. Hossain, et al., "Quality of Service in Software Defined Networking," Global Journal of Computer Science and Technology: Network, Web, and Security, Vol. 18, Issue 3, 2018.
- [16] Peffer, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S. (2007) : A Design Science Research Methodology for Information System Research. Volume 24 Issue 3, Winter 2007-8, pp. 45-78.