

Simulasi Kinerja Berbagai Topologi Jaringan Berbasis *Software-Defined Network (SDN)*

Naufal Abyan Faruqi¹, Luthfi Nurwadi², Nanang Ismail³, Dodi Maryanto⁴

^{1,2,3}Teknik Elektro UIN Sunan Gunung Djati Bandung

⁴Gerbang Telekomunikasi dan Informasi, Cimahi Indonesia

naufal.al_faruqi@ymail.com¹, luthfinurwadi@gmail.com², nanang.is@uinsgd.ac.id³, dodi@gti.co.id⁴

Abstrak – Kebutuhan akan teknologi perangkat jaringan di berbagai perusahaan mengalami perkembangan yang sangat pesat. Performansi jaringan, penambahan konfigurasi yang semakin kompleks dan besar, bagian kontrol jaringan pun akan semakin rumit, tidak fleksibel dan sulit untuk diatur. *Software Defined Network (SDN)* adalah sebuah pendekatan baru untuk merancang, membangun dan mengelola jaringan komputer dengan memisahkan control plane dan data plane. Protokol yang paling menonjol pada SDN yaitu *OpenFlow*. *OpenFlow* adalah Protokol komunikasi antarmuka yang berada antara control dan forwarding layer. Makalah ini membahas simulasi jaringan berbasis SDN, menggunakan tools/aplikasi mininet. Mininet yaitu emulator jaringan opensource yang digunakan untuk penelitian SDN. Tujuan dari simulasi ini melakukan analisa nilai QoS pada berbagai topologi jaringan. Berdasarkan hasil pengujian, QoS yang dihasilkan memenuhi standar ITU – T meskipun mengalami ketidakstabilan.

Kata kunci : Delay, Jitter, *OpenFlow*, Throughput, SDN

1. Pendahuluan

Semakin berkembangnya dunia industri Teknologi Informasi (IT) membuat semakin banyak inovasi-inovasi dan penelitian yang bertujuan untuk mempermudah pengorganisasian pada jaringan komputer[1]. Pada perkembangan teknologi ini munculah konsep baru yaitu *Software – Defined Networking (SDN)*. *Software – Defined Networking (SDN)* adalah sebuah konsep pendekatan baru untuk mendesain, membangun dan mengelola jaringan komputer dengan memisahkan control plane dan data plane[2]. Konsep utama pada *Software – Defined Networking (SDN)* adalah sentralisasi jaringan dengan semua pengaturan berada pada control plane[3].

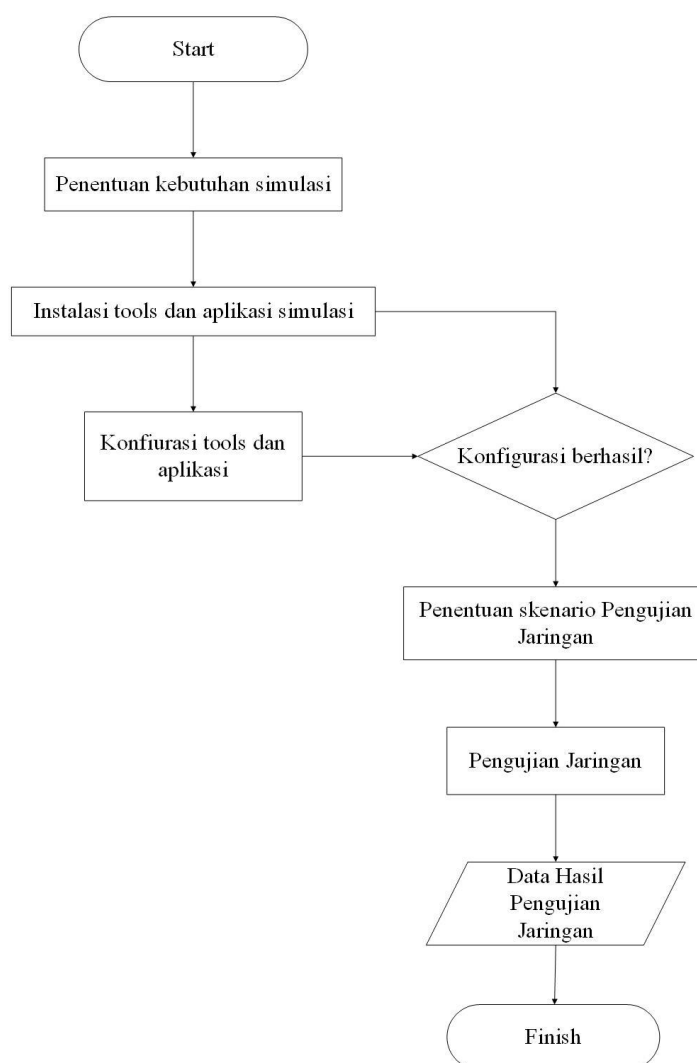
Pada jaringan konvensional administrator jaringan diharuskan menangani puluhan, ratusan atau bahkan ribuan perangkat jaringan didalam sebuah organisasi[1]. Permasalahan ini sering ditemukan dalam dunia industri.

Konsep SDN dan *OpenFlow* dapat memudahkan operator dan network administrator dalam mengelola jaringannya. *OpenFlow* dapat mengakses dan memanipulasi secara langsung forwarding plane (data plane) dari perangkat-perangkat jaringan tersebut, baik secara fisik maupun virtual[4].

Saat ini belum semua memakai protokol *OpenFlow*, Oleh karena itu untuk melakukan penelitian ini dibutuhkan simulator yang mendukung protokol *OpenFlow* yaitu Mininet.

2. Metoda Penelitian

Secara Umum langkah – langkah yang dilakukan dalam penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Diagram Alir Penelitian[4].

2.1 Perancangan Sistem

Penjelasan mengenai langkah –langkah dari perancangan sistem yaitu sebagai berikut:

1. Penentuan kebutuhan simulasi, sesuai dengan konsep utama SDN yang memisahkan antara control plane dan data plane.
2. Instalasi tool/aplikasi untuk simulasi. Instal Mininet dengan versi 2.2.1, instal paket protokol OpenFlow dengan versi 1.0.0 dan kemudian install controller POX dengan versi/branch dart 0.3.0. Semua terinstall pada OS Linux Ubuntu 14.04.3.
3. Konfigurasi tool/aplikasi Mininet agar dapat terhubung dengan server controller SDN. Bisa dikatakan terhubung apabila MAC address dari switch – switch yang terdeteksi oleh controller SDN.
4. Pengecekan konfigurasi. Seperti yang sudah dijelaskan di atas, pada server controller SDN menginvoke controller POX. Sedangkan pada Mininet menggunakan perintah remote controller yang artinya Mininet mengaktifkan controller namun bukan local controller Mininet. Apabila konfigurasi berhasil identitas – identitas perangkat – perangkat jaringan seperti MAC Address terdeteksi oleh controller POX.

5. Pada penentuan jenis topology jaringan yang digunakan dalam simulasi jaringan ini adalah topologi full mesh, ring, linier, tree, dan star. Dalam perancangan semua bentuk topologi tools yang digunakan dengan perintah command

```
$ sudo ./mininet/examples/miniedit.py
```

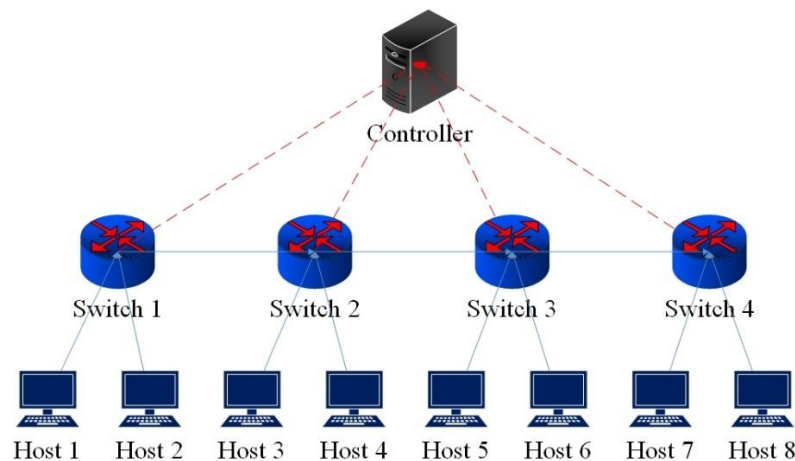
6. Miniedit adalah sebuah graphical user interface untuk perancangan skenario jaringan SDN.
7. Perlu adanya mengecek konektivitas agar dapat terbentuk komunikasi. Untuk mengecek konektivitas jaringan digunakan perintah pingall.
8. Setelah dilakukan pengujian jaringan maka akan didapatkan hasil dari mengujian jaringan berupa tabel (QoS) Quality of Service dimana paramter – parameternya berupa delay, jitter, throughput, dan packet loss.

2.2 Perancangan Jaringan dengan Topologi

Pada tahap ini dilakukan perancangan Jaringan pada sistem Jaringan SDN, dimana topologi yang digunakan adalah linier, star, tree, ring, fullmesh. Pemilihan berbagai topologi dikarenakan untuk membandingkan QoS pada setiap topologi Jaringan. Setiap bentuk topologi menyesuaikan dengan jumlah node yang kan di rancang. Berikut design yang digunakan dalam perancangan Topologi ini :

1. Topologi linier.

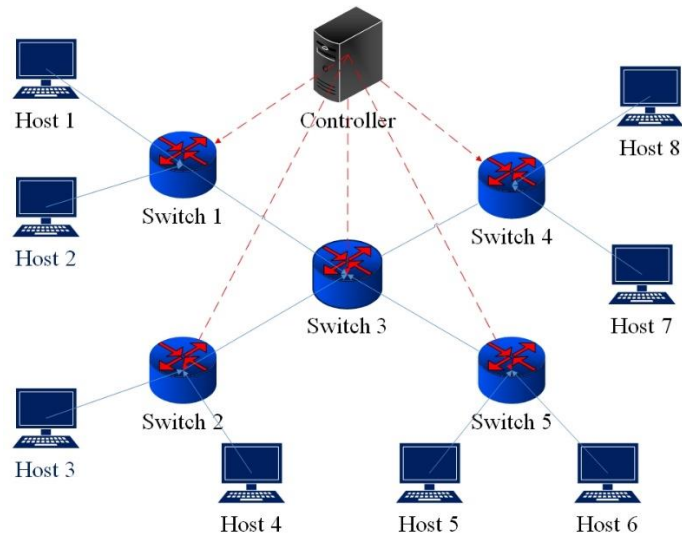
Pada Topologi ini terdapat 1 Controller, 4 Switch dan 8 Host yang dihubungkan ke Switch (Gambar 2).



Gambar 2 Topologi Linier

2. Topologi star:

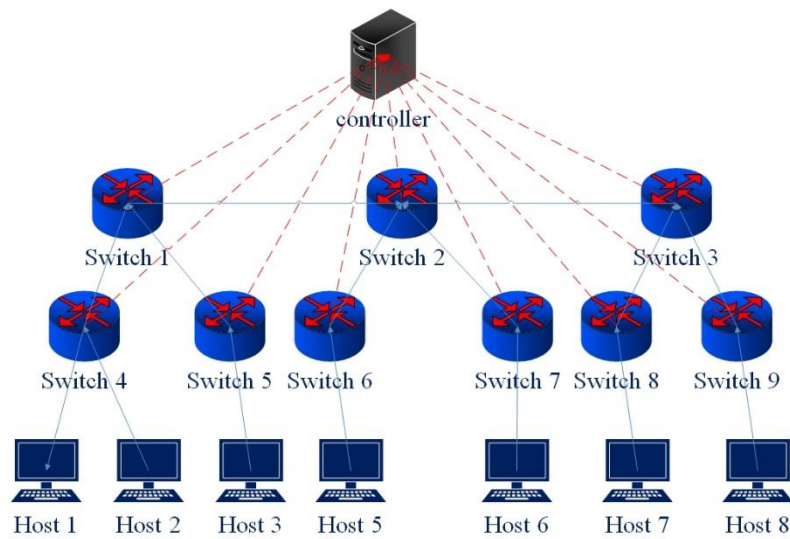
Pada Topologi ini digunakan 1 Controller, 5 Switch dan 8 Host yang terhubung ke Switch (Gambar 3).



Gambar 3. Topologi Star.

3. Topologi tree.

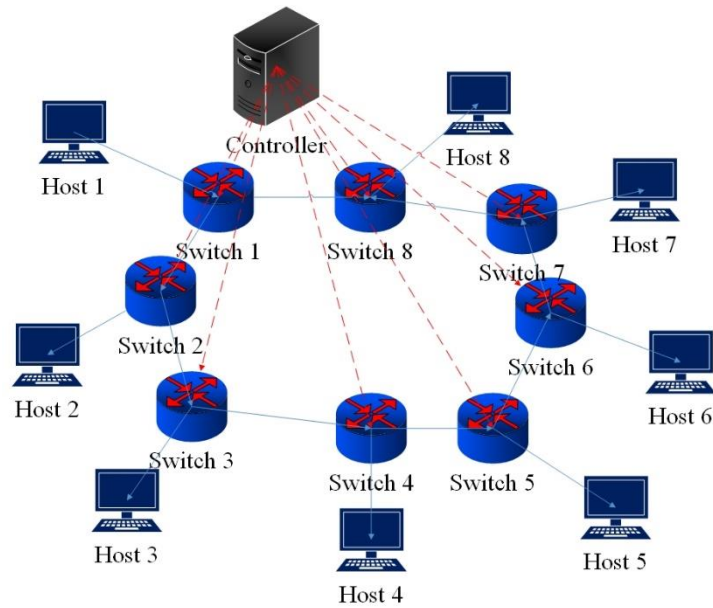
Pada Topologi ini terdapat 1 Controller, 9 Switch, dan 8 Host yang terhubung ke Switch (Gambar 4).



Gambar 4. Topologi tree

4. Topologi Ring.

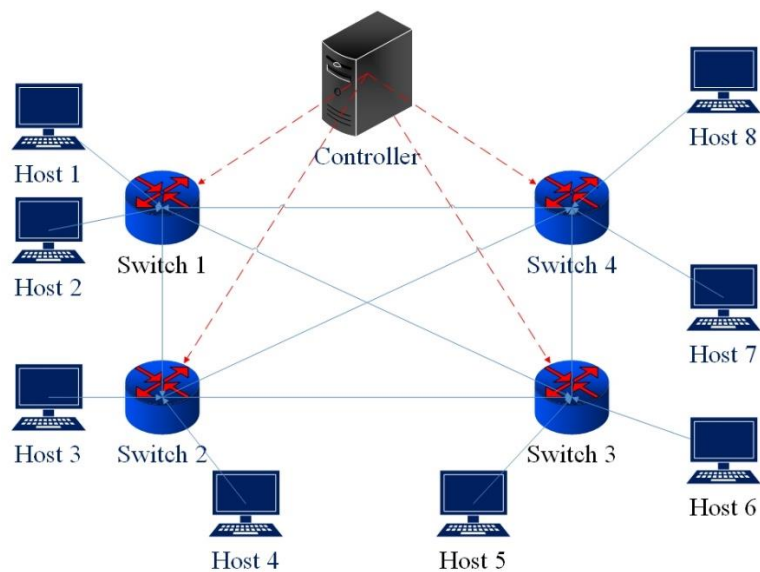
Pada Topologi ini digunakan 1 Controller, 8 Switch, dan 8 Host yang terhubung ke Switch (Gambar 5).



Gambar 5. Topologi Ring.

5. Topologi fullmesh.

Pada Topologi ini digunakan 1 Controller, 4 Switch yang terhubung satu sama lain, dan 8 Host yang terhubung ke Switch (Gambar 6).



Gambar 6. Topologi fullmesh.

3. Hasil dan Pembahasan

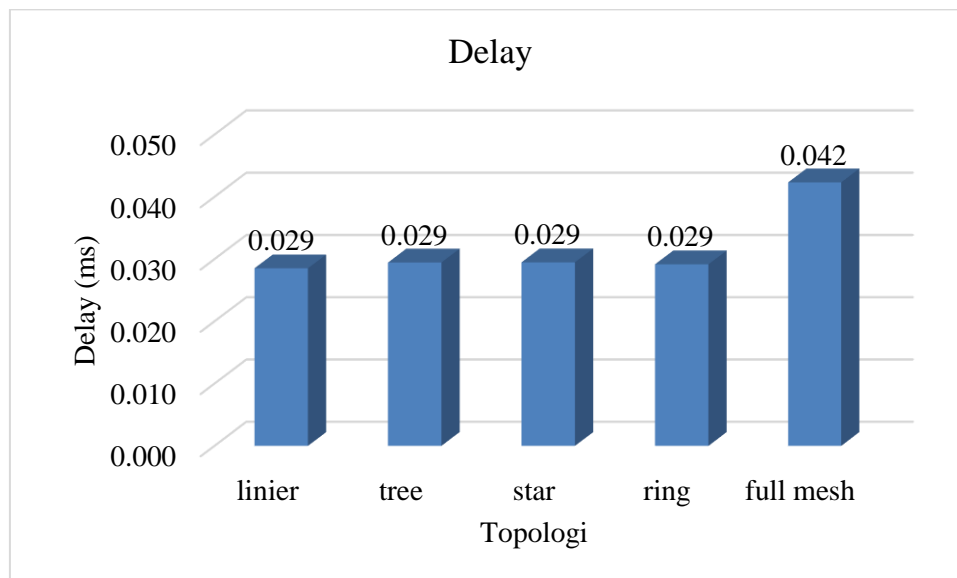
Pada Pembahasan kali ini dilakukan Pengujian untuk mengetahui perilaku Jaringan SDN serta performasinya, diukur dengan Parameter (QoS) *Quality of Service* menggunakan parameter berupa *delay*, *jitter*, dan *throughput*. Hasil dari pengukuran didapatkan dari monitoring *capture* aplikasi *wireshark*.

Table 1. Hasil Pengukuran Jaringan.

Topologi	Delay (ms)	Jitter (ms)	Troughput (bits/sec)	
			TCP	UDP
<i>linier</i>	0,02852	0,041	15,27	16,45
<i>tree</i>	0,02945	0,04	14,79	15,63
<i>star</i>	0,02948	0,163	14	14,25
<i>ring</i>	0,02916	0,23	5,2	6,87
<i>full mesh</i>	0,04232	0,82	9,34	9,53

3.1. Delay

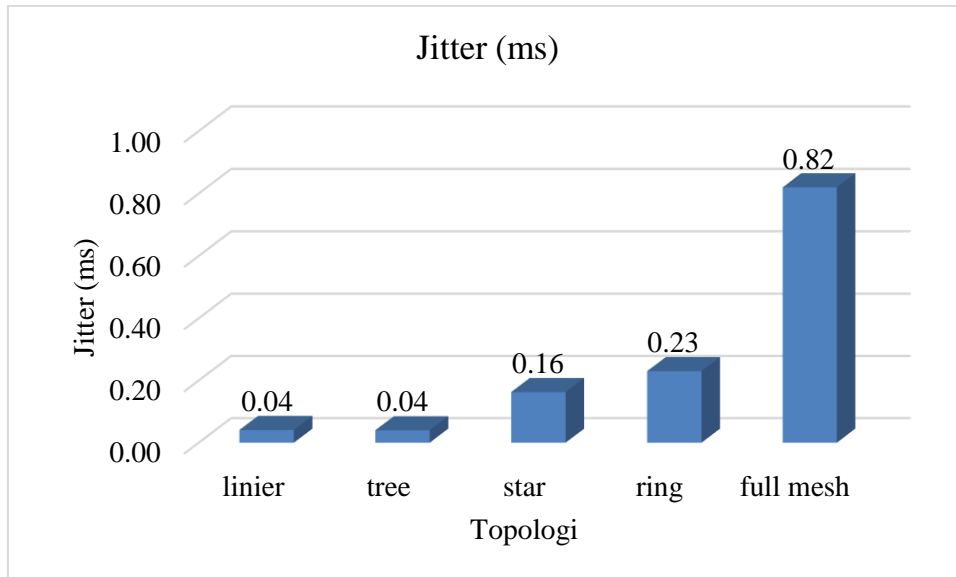
Pola keseluruhan nilai delay dari hasil pengukuran cenderung stabil, namun ada beberapa yang mengalami kenaikan dan penurunan. Kenaikan terjadi pada topologi full mesh. Hal ini terjadi karena pada saat penentuan data flow, terdapat penambahan perangkat seperti perangkat jaringan yang dituju dan port – port yang tersedia pada suatu switch jaringan tersebut. Serta pula menyebabkan bertambahnya waktu pengecekan alamat fisik dari perangkat jaringan dan port – port yang tersedia. Berikut grafik Delay yang ditunjukkan pada gambar 7.



Gambar 7. Grafik Delay.

3.2. Jitter

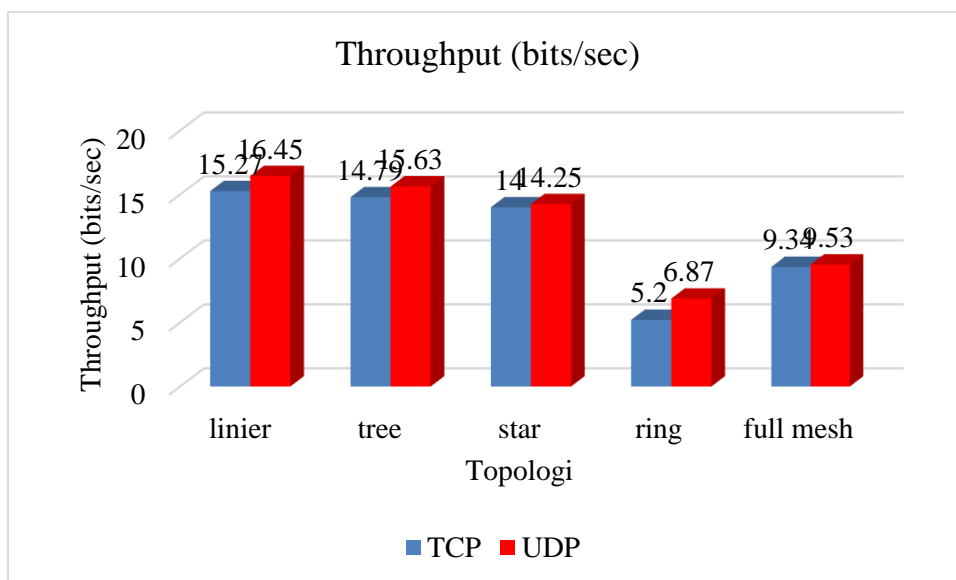
Pada keseluruhan nilai Jitter terdapat peningkatan dan penurunan. Dapat dilihat terjadi peningkatan pada topologi *star*, *ring* dan *full mesh*. Hal ini menunjukkan kompleksitas sebuah jaringan menyebabkan kenaikan nilai jitter. Namun ada pula penurunan pada topologi *tree* yang diakibatkan pengaruh dari waktu komunikasi itu sendiri. Nilai – nilai jitter tersebut masih memenuhi standar rekomendasi ITU – T yaitu dibawah 50 ms. Berikut Grafik Jiiter yang ditunjukkan pada Gambar 8.



Gambar 8. Grafik Jitter.

3.3 Throughput

Throughput untuk setiap TCP maupun UDP mengalami penurunan. Topologi *linier* adalah yang paling tinggi. Hal ini dikarenakan source setiap host pada topologi ini mudah menemukan jalur destination (tidak perlu mengkonfigurasi lagi data flow nya) sehingga transfer data lebih cepat. Sedangkan penurunan terjadi pada topologi ring dan full mesh. Hal ini dikarenakan kompleksitas jaringan pada jaringan tersebut membuat kecepatan transfer data rendah. Berikut Grafik yang ditunjukkan pada Gambar 9.



Gambar 9 Grafik Throughput.

4. Kesimpulan

Setelah dilakukan pengujian, berikut ini adalah beberapa hal yang dapat disimpulkan dalam laporan simulasi ini.

1. Topologi jaringan berbasis SDN dapat dirancang dengan tools emulator mininet dengan menuliskan command pada terminal miniedit.py.
2. Hasil pengukuran parameter Delay, Jitter, dan Throughput memenuhi standar ITU – T untuk semua topologi, meskipun mengalami ketidakstabilan.

Ucapan Terima Kasih

Ucapan terimakasih diberikan kepada seluruh jajarannya, dosen dan juga staff Jurusan Teknik Elektro Universitas Islam Negeri Sunan Gunung Djati Bandung. Teman – teman mahasiswa dan mahasiswi Jurusan Teknik Elektro Universitas Islam Negeri Sunan Gunung Djati Bandung yang telah mendukung penuh serta memberi bantuan terhadap penelitian yang dilakukan.

Daftar Pustaka

- [1] D. Achmadi, “Implementasi dan Analisa Quality of Services (QoS) Pada Jaringan Berbasis Software-Defined Networking (SDN) Menggunakan SDN Controller OpenDayLight,” Universitas Mercubuana, Jakarta, 2017.
- [2] Open Networking Foundation, “SDN Architecture Overview,” *Onf*, no. 1, hal. 1–5, 2013.
- [3] F. Adnantya, S. N. Hertiana, L. Vidya, dan Y. St, “Simulasi Dan Analisis Performansi Protokol Ruting Ebgp Pada Sdn (Software Defined Network) Simulation and Perfomance Analysis of Ebgp Routing Protocol on Sdn (Software Defined Network),” *e-Proceeding Eng.*, vol. 2, no. 2, hal. 2346–2353, 2015.
- [4] I. Ummah, “Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking,” *Indones. J. Comput.*, vol. 1, no. 1, hal. 95–106, 2016.