ANALISIS KINERJA ARSITEKTUR *SPINE AND LEAF DATA CENTER* MENGGUNAKAN SDN KONTROLER FLOODLIGHT

ANALYSIS OF SPINE AND LEAF DATA CENTER ARCHITECTURE USING FLOODLIGHT SDN CONTROLLER

Rio Niki Putra[1], Anton Siswo Raharjo Ansori[2], Aliwarman Tarihoran[3]

[1,2,3] Universitas Telkom, Bandung
[1]**rioniki@student.telkomuniversity.ac.id**, [2]**raharjo@telkomuniversity.co.id**,
[3]**aliwarmantarihoran@telkomuniversity.ac.id**

Abstrak

Internet adalah salah satu teknologi yang dibutuhkan manusia hampir pada setiap aktivitas. Meningkatnya pengguna internet menyebabkan peningkatan perangkat yang dibutuhkan dan meningkat pula perangkat yang perlu dikelola. Penerapan arsitektur *Software Defined Network* dinilai dapat dijadikan solusi pada masalah yang dihadapi oleh jaringan konvensional.

Konsep yang diterapkan oleh Software Defined Network yaitu pemisahan *Control Plane dan Data Plane.* Pemisahan *Control Plane* dan *Data Plane* bertujuan agar terpusatnya kontrol jaringan. Terpusatnya kontrol dinilai dapat mempermudah pengelolaan perangkat jaringan yang semakin meningkat. *Controller* yang digunakan pada penelitian ini yaitu Floodlight. Diharapkan aristektur jaringan menggunakan SDN *controller* Floodlight dapat menghasilkan QoS yang lebih baik ketimbang arsitektur jaringan konvensional.

Penelitian yang dilakukan menghasilkan nilai QoS yang baik pada arsitektur jaringan menggunakan SDN *Controller* Floodlight dan arsitektur jaringan konvensional. Parameter QoS yang diujikan yaitu *Throughput, Delay,* dan *Jitter*. Jika dibandingkan, maka Arsitektur jaringan menggunakan SDN *Controller* Floodlight menghasilkan nilai yang lebih baik ketimbang arsitektur jaringan konvensional.

Kata kunci : SDN *Controller*, Floodlight, *Spine and Leaf*, QoS.

Abstract

Internet is one of the technologies that humans need in almost every activity. The increase in internet users causes an increase in the number of devices needed and an increase in the number of devices that need to be managed. The application of the Software Defined Network architecture is considered to be a solution to the problems faced by conventional networks.

Concept applied by the Software Defined Network is the separation of the Control Plane and the Data Plane. The separation of Control Plane and Data Plane aims to centralize network control. Centralized control is considered to be able to facilitate the management of increasing network devices. The controller used in this research is Floodlight. It is expected that the network architecture using the Floodlight SDN controller can produce better QoS than conventional network architectures.

The research carried out resulted in a good QoS value on the network architecture using SDN Controller Floodlight and conventional network architecture. The QoS parameters tested are Throughput, Delay, and Jitter. When compared, the network architecture using SDN Controller Floodlight produces better value than conventional network architectures.

Keywords: SDN Controller, Floodlight, Spine and Leaf, QoS.

1. Pendahuluan

Salah satu teknologi yang dibutuhkan manusia adalah internet, internet dibutuhkan hampir di segala aktivitas manusia. Akibatnya, pengguna Internet meningkat, tetapi infrastruktur jaringan utama yang konstan membuat pembangunan jaringan semakin rumit [1]. Peningkatan jumlah pengguna internet ini pun membutuhkan perangkat-perangkat dengan kinerja baik, cepat, dan mudah dikelola.

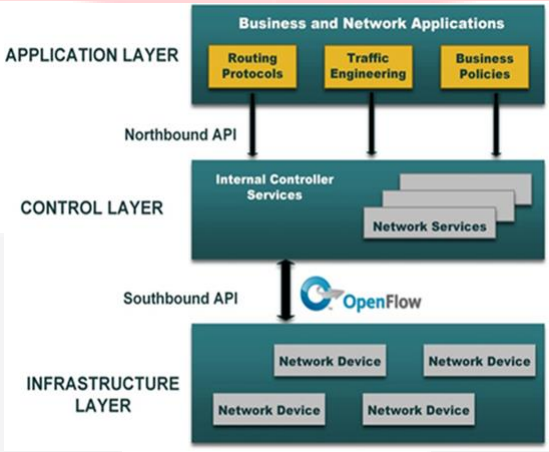
Data Center mempunyai peran penting untuk mewujudkan kinerja yang maksimal. *Data Center* ini adalah sumber daya yang meliputi *Server*, *Storage*, Network. Pengembangan desain dari *Data Center* tradisional dinilai mampu mewujudkan kinerja yang maksimal. Implementasi SDN memungkinkan administrator jaringan untuk mengelola dan memprogram jaringan dengan lebih mudah [2].

SDN merupakan pengembangan dari jaringan tradisional. Prinsip SDN yaitu memisahkan antara *Control Plane* dengan *Data Plane*. *Control Plane* bertindak sebagai pengontrol/pengendali laju dari *Data Plane*, dan *Data Plane* terdapat pada perangkat jaringan

2. Dasar Teori

2.1 Software Define Network (SDN)

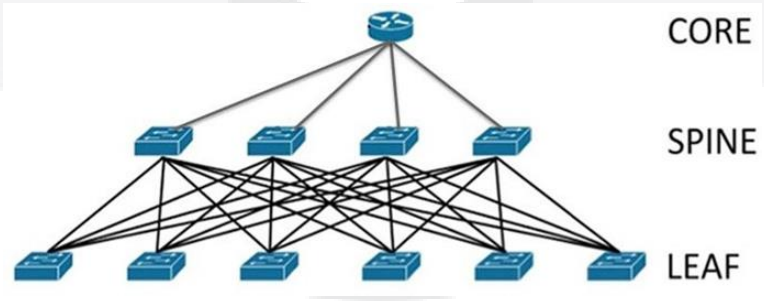
Software Defined Network (SDN) adalah suatu gagasan baru dalam jaringan internet yang memiliki konsep memisahkan antara *Control Plane* dan *Data Plane*. *Control Plane* dapat diibaratkan sebagai otak yang mengatur trafik data [2], dan *Data Plane* adalah perangkat keras tersebut (*Hardware*).



Gambar 1. Jaringan SDN

2.2 *Spine And Leaf*

Spine and Leaf yaitu arsitektur yang didalamnya terdapat 2 tingkatan yaitu lapisan *Spine* dan lapisan *Leaf*. Lapisan *Leaf* (*low-tier switch*) terdiri dari *switch* yang terhubung dengan perangkat seperti router, dan lapisan *Spine* (top tier *switch*) bertanggung jawab menghubungkan semua *switchLeaf* [6].



Gambar 2. *Spine and Leaf*

2.3 Floodlight *Controller*

Floodlight adalah salah satu *Controller* yang berbasis bahasa pemrograman Java yang dikembangkan oleh Developer pada komunitas di Big *Switch* Network [5]. Floodlight *Controller* menggunakan protokol *open flow* untuk mengatur arus lalu lintas di jaringan yang ditentukan software (SDN). Floodlight bersifat *open source* yang artinya bersifat terbuka untuk dikembangkan

Gambar 3. Floodlight *Controller*

2.4 Virtual Local Area Network (VLAN)

VLAN adalah protokol yang memungkinkan beberapa jaringan dibuat pada segmen jaringan yang sama, VLAN pun memungkinkan untuk mengelola suatu *host* agar terhubung seolah-olah *host* tersebut berada pada jaringan yang sama meskipun secara fisik berbeda.
VLAN diklasifikasikan bersumber pada metode (tipe) yang digunakan untuk mengelompokannya, baik memakai *port*, MAC *addresses*, dsb [8].

2.5 *Data Center*

Data Center yaitu kumpulan sumber daya (*Server*, *Storage*, Network) yang saling terhubung oleh jaringan. *Data Center* memiliki peran penting pada jaringan yaitu menghubungkan sumber daya yang ada [4].

2.6 Mininet

Mininet adalah suatu emulator jaringan yang didalamnya terdapat perangkat-perangkat jaringan seperti *host*, *switch*, *link*, *Controller*[. Perangkat jaringan yang terdapat pada mininet yaitu sebuah peerangkat-perangkat virtual. Mininet dapat dikatakan sebuah emulator yang mendukung jaringan SDN[7].

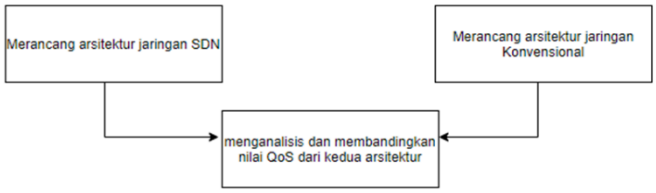
2.7 Quality of Service (QoS)

QoS (Quality of Service) merupakan metode penilaian terhadap kondisi jaringan dan merupakan upaya untuk mendefinisikan karakter suatu layanan[4]. QoS memiliki parameter parameter yaitu *throughput*, *delay*, *jitter*, *dan packet loss*.

3. Perancangan Sistem

3.1 Perancangan Desain Sistem

Secara keseluruhan gambran umum tahapan dari proses perancangan system direpresentasikan sebagai berikut.



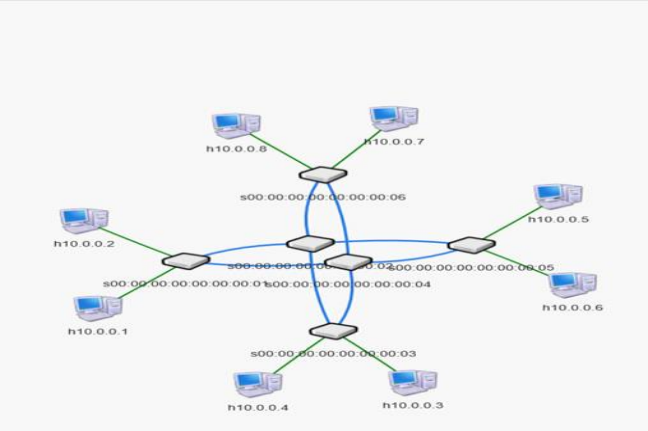
Gambar 4. Gambaran Umum

3.1.1 Arsitektur Sistem

Terdapat 2 arsitektur sistem pada penelitian ini yaitu arsitektur jaringan konvensional dan arsitektur jaringan SDN *Controller* mengunakan topologi *Spine and Leaf*. Pada arsitektur jaringan tradisional prinsip yang digunakan yaitu *three hirarcy model*. Arsitektur tradisional memiliki 3 lapisan yaitu *Core*, *Aggregation*, *Access*.

3.1.2 Rancangan Topologi

Topologi yang digunakan untuk kedua arsitektur yaitu topologi *Spine and Leaf*, 2 *switch* berperan sebagai *Spine*, dan 4 *switch* berperan sebagai *Leaf*. Setiap *switch Leaf* terhubung dengan 2 *host* yang meggunakan vlan id berbeda



Gambar 5. Topologi *Spine and Leaf*

3.2 Perancangan Perangkat Pendukung

Pada penelititan tugas akhir ini dibutuhkan perangkat pendukung. Perangkat pendukung tersebut dibagi menjadi dua, yaitu perangkat lunak dan perangkat keras.

3.2.1 Perangkat Keras

Perangkat keras yang digunakan pada penelitian ini sebagai berikut:

Tabel 1. Perangkat Keras

Komponen	Spesifikasi
Processors	Amd Ryzen 5 3500U
Memory	8GB
HDD	500GB

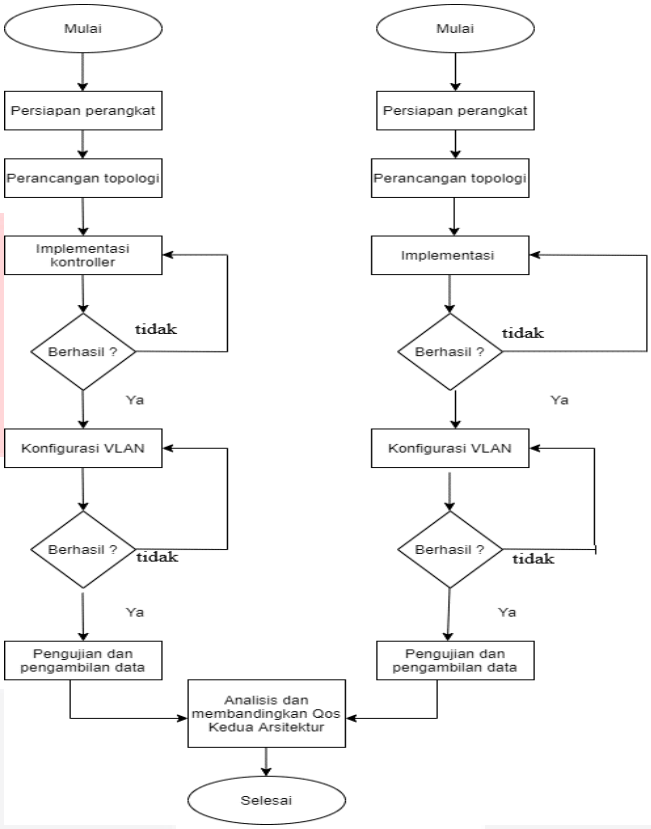
3.2.2 Perangkat Lunak

Berikut merupakan perangkat lunak yang digunakan pada penelitian tugas akhir ini:

Tabel 2. Perangkat Lunak

Software	Keterangan
Linux Ubuntu	*Operating System*
Mininet	Emulator perancangan jaringan
D-ITG	Untuk memonitoring jaringan
Iperf3	Untuk membuat *Background Traffic*
Floodlight	SDN *Controller*
Vmware	*Software Virtualisasi* untuk membuat VM

3.3 Diagram Alir

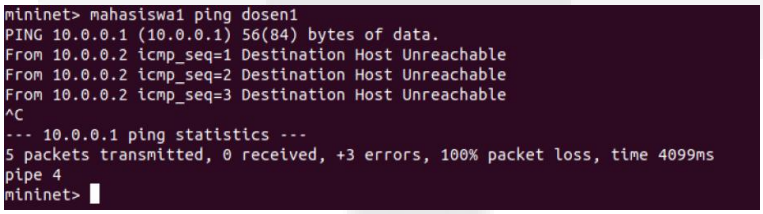


Gambar 6. Diagram Alir

4. Skenario Pengujian

4.1 Pengujian VLAN Jaringan SDN

Pada jaringan SDN terdapat 8 host, yang masing masing diberi nama dosen1, dosen2, mahasiswa1, dan mahasiswa 2. Untuk dosen memiliki vlan_id 100, dan mahasiswa memiliki vlan_id 200. Dosen1 dan mahasiswa1 terhubung pada switch1, dosen2 dan mahasiswa2 terhubung pada switch 3, dosen3 dan mahasiswa3 terhubung pada switch5, dosen4 dan mahasiswa4 terhubung pada switch6. Dosen1 memiliki ip 10.0.0.1, mahasiswa1 memiliki ip 10.0.0.2, dosen2 memiliki ip 10.0.0.3, mahasiswa2 memliki ip 10.0.0.4, dosen3 memiliki ip 10.0.0.5, mahasiswa3 memiliki ip 10.0.0.6, dosen4 memiliki ip 10.0.0.7, dan mahasiswa4 memiliki ip 10.0.0.8.



```
mininet> mahasiswa1 ping dosen1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
^C
--- 10.0.0.1 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 4099ms
pipe 4
mininet>
```

Gambar 7. Uji ping berbeda vlan

Selanjutnya dilakukan pengujian antara mahasiswa1 dan mahasiswa2 yang memiliki VLAN id yang sama

```
mininet> mahasiswa1 ping mahasiswa4
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=0.422 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=0.134 ms
```

Gambar 8. Uji ping sesame vlan

4.2 Pengujian VLAN Jaringan Konvensional

Pada jaringan konvensional yang dibuat pada emulator mininet, jumlah *host*, ip *address*, dan vlan_id sama dengan konfigurasi yang dilakukan pada jaringan SDN.

```
mininet> mahasiswa1 ping dosen1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
^C
--- 10.0.0.1 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3062ms
```

Gambar 9. Uji ping berbeda vlan konvensional

Pada pengujian ping mahasiswa1 kepada dosen1 muncul keterangan "Destination *Host* Uncreachable" yang berarti tujuan tidak dapat diakses. Tujuan tidak dapat diakses dikarenakan 2 *host* tersebut berbeda vlan_id, walaupun 2 *host* tersebut terhubung pada *switch* yang sama.

```
mininet> mahasiswa1 ping mahasiswa4
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
64 bytes from 10.0.0.8: icmp_seq=1 ttl=64 time=0.443 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=64 time=0.066 ms
```

Gambar 10. Uji ping sesama vlan konvensional

4.3 Pengujian Quality of Service

Pengujian QoS pada jaringan SDN dan Konvensional dilakukan menggunakan D-ITG dan Iperf3. Pada pengujian QoS, mahasiswa1 dijadikan sender, dan mahasiswa4 dijadikan receiver. Protokol yang digunakan yaitu protokol UDP (*User Datagram Protocol*). Waktu (t) yang digunakan yaitu 15s, Bandwitch (c) yang digunakan yaitu 100, 200, 300, 400 Mb, dan ditambah dengan background traffic sebesar 100mb

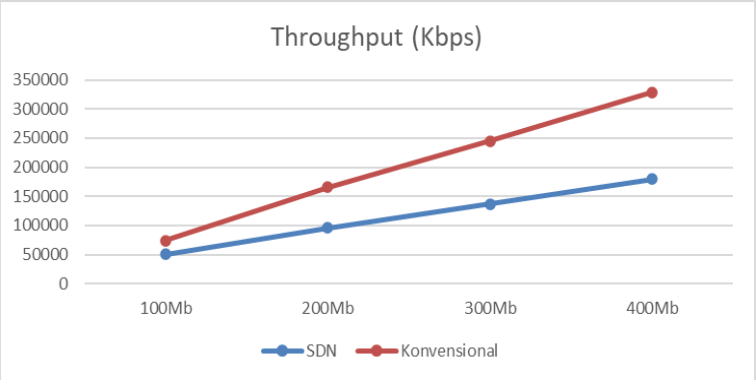
4.3.1 Hasil Pengujian

Tabel 3. Hasil pengujian QoS

Parameter	100Mb		200Mb		300Mb		400Mb	
	SDN	Konvensional	SDN	Konvensional	SDN	Konvensional	SDN	Konvensional
Throughput (Kbps)	50092.152	24335.017	95766.128	69448.849	137228.564	108246.157	179869.53	148260.934
Delay (ms)	0.287	0.739	0.284	0.930	0.328	0.806	0.087	1.010
Jitter (ms)	0.0950	0.891	0.0933	1.004	0.0839	1.001	0.087	1.0102

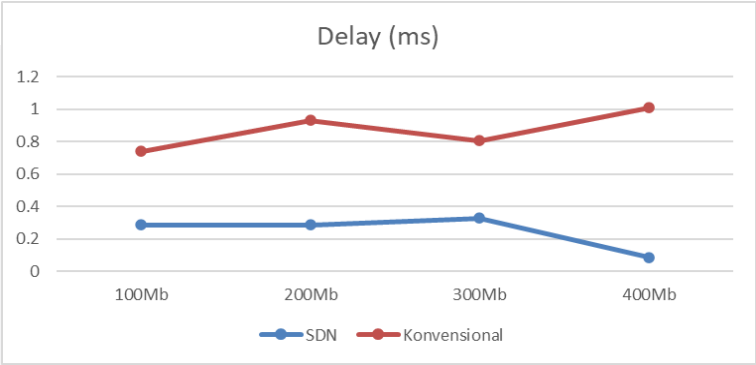
4.4 Analisis Dan Perb*and*ingan

Dari setiap pengujian dengan *Bandwitch* yang berbeda, jaringan dengan menggunakan SDN *Controller* menghasilkan nilai *throughput* yang selalu lebih besar dari jaringan konvensional, yang berarti jaringan SDN menghasilkan nilai *throughput* lebih baik ketimbang jaringan konvensional. Nilai *throughput* pada jaringan SDN masuk pada kategori terbaik menurut standarisasi TIPHON.



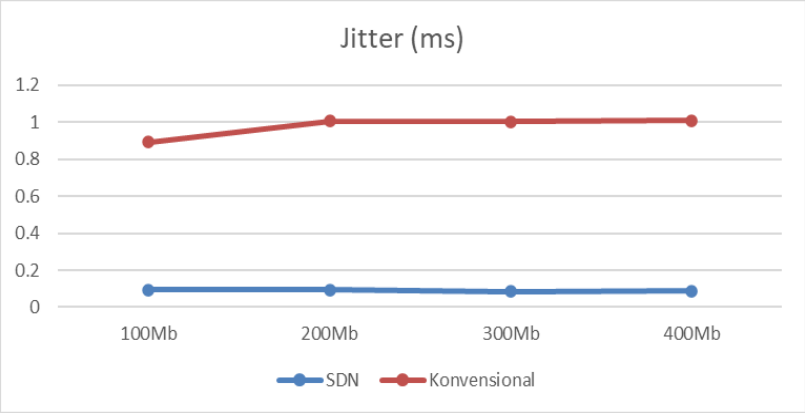
Gambar 10. Grafik Perb*and*ingan *Throughput*

Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Jika nilai *delay* semakin kecil, maka semakin baik. Pada parameter *delay* ini jaringan SDN lebih baik ketimbang jaringan konvensional karena nilai *delay* SDN pada setiap pengujiian *Bandwitch* yang berbeda selalu menghasilkan nilai yang lebih kecil ketimbang nilai *delay* jaringan konvensional. Dan menurut standarisasi TIPHON nilai *delay* dari jaringan SDN termasuk pada kategori sangat bagus.



Gambar 11. Grafik Perb*and*ingan *Delay*

Rata rata nilai *jitter* jaringan SDN pada setiap pengujian dengan besaran *Bandwitch* yang berbeda sangat mendekati 0, dan lebih kecil ketimbang jaringan konvensional, yang berarti pada parameter *jitter*, jaringan SDN lebih baik ketimbang jaringan konvensional. Dan menurut standarisasi TIPHON, nilai *jitter* pada jaringan SDN termasuk kategori Bagus.



Gambar 12. Grafik Perb*and*ingan *Jitter*

5. Kesimpulan dan Saran

5.1 Kesimpulan

Pada penelitian ini mendapatkan kesimpulan bahwa pengujian QoS SDN memperoleh hasil yang lebih baik ketimbang jaringan Konvensional. Parameter QoS yang dijadikan perbandingan yaitu *Throughput*, *Delay*, dan *Jitter*. Hasil dari semua pengujian *throughput*, *delay* dan *jitter* jaringan SDN controller mendapatkan nilai rata-rata lebih baik dibandingkan jaringan konvensional.

Pada nilai Throughput, jaringan menggunakan SDN Controller Floodlight menunjukan nilai yang lebih besar, yaitu 50092.152 Kbps untuk 100mb, 95766.128 Kbps untuk 200mb, 137228.564 Kbps untuk 300mb dan 179869.530 Kbps untuk 400mb ketimbang jaringan konvensional yang mendapatkan nilai rata-rata 24355.0179 Kbps untuk 100mb, 69948.849 Kbps untuk 200mb, 108246.157 Kbps untuk 300mb dan 400mb mendapatkan 148620.934 Kbps, yang berarti jaringan menggunakan SDN Controller lebih unggul.

Jaringan menggunakan SDN *Controller* Floodlight menghasilkan nilai delay dan jitter lebih kecil ketimbang jaringan konvensional, dapat disimpulkan pada parameter delay dan jitter, jaringan dengan menggunakan SDN Controller Floodlight lebih baik ketimbang jaringan konvensional.

5.2 Saran

Beberapa saran yang dapat dipertimbangkan untuk penelitian selanjutnya:

1. Penambahan jumlah *switch* dan *host*, dan penambahan VLAN id.
2. Pengembangan pada *Controller* Floodlight.
3 Menggunakan protokol selain VLAN.
4 Menggunakan topologi yang berbeda.

Referensi:

[1] M. Wahyudi Putra, E. Sakti Pramukantoro, Widhi Yahya, "Analisis Perb*and*ingan Performansi Kontroler Floodlight, Maestro,RYU, POX dan ONOS dalam Arsitektur Software Defined Network (SDN)," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 10, Oktober 2018.

[2] Melissa, Susmini Indriani, " Analisis Kinerja Arsitektur Software-Defined Network Berbasis OpenDaylight *Controller*," Jurnal Teknik Komputer Unikom, Volume 7, No.1, 2018

[3] Ahmad Heryanto, Afrilia, Skripsi " SOFTWARE DEFINED NETWORK MENGGUNAKAN SIMULATOR MININET" (Palembang: Unsri,2016), Hal 2.

[4] Luthfi Setioutomo, E. Purwa Laksana, " Desain Arsitektur Jaringan Management Environment Dengan Software Defined Network," Jurnal Maestro, Volume 3, No.1, 2020

[5] Sizka L. Hanifa, Rikie Kartadie, " UJI PERFORMA KONTROLER SOFTWARE-DEFINE NETWORK FLOODLIGHT vs ONOS," JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), Volume 3, No.2, 2018.

[6] Cisco *Data Center Spine-and-Leaf* Architecture: Design Overview White Paper. Diakses 27Nov 2020 dari: https://www.cisco.com/c/en/us/products/collateral/*switch*es/nexus-7000-series *switch*es/white-paper-c11-737022.html

[7] Raihan Radito, Primantara Hari Trisnawan, Kasyful Amron, "Implementasi Pencarian Jalur berdasarkan B*and*width dengan menggunakan Algoritme Dijkstra pada Arsitektur Jaringan Software-Defined Networking (SDN)," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 4, No. 5, Mei 2020

[8] Rohmat Tulloh " Analisis Performansi VLAN Pada Jaringan Software Defined Network (SDN)," Jurnal Infotel Vol.9 No.4 November 2017

[9] Aprianto Puji Adikusuma, Asmunin, "IMPLEMENTASI SIMPLE PORT KNOCKING PADA DYNAMIC ROUTING (OSPF) MENGGUNAKAN SIMULASI GNS3, " Jurnal Manajemen Informatika. Volume 5 Nomor 2 Tahun 2016

[10] M. Fadlulloh Romadlon Bik " IMPLEMENTASI DOCKER UNTUK PENGELOLAAN BANYAK APLIKASI WEB," Jurnal Manajemen Informatika. Volume 7 Nomor 2 Tahun 2017