

PENERAPAN ARSITEKTUR SOFTWARE-DEFINED NETWORKING BERBASIS OPENFLOW PADA SIMULASI JARINGAN VIRTUAL

Himawan Eka Putra¹, Susmini Indriani Lestaringati²

^{1,2} Jurusan Teknik Komputer, Unikom, Bandung

Hima.cia008@email.unikom.ac.id, susmini.indriani@email.unikom.ac.id

ABSTRAK

Penelitian ini bertujuan untuk menganalisis kinerja dari jaringan yang menggunakan arsitektur SDN dan dibandingkan dengan kinerja jaringan yang menggunakan arsitektur tradisional. Penelitian ini dilakukan dengan cara melakukan simulasi jaringan dengan arsitektur SDN dan tradisional menggunakan software emulator Mininet. Untuk mengukur kinerja dari kedua arsitektur jaringan dapat dilihat dari nilai QoS yang dihasilkan menggunakan software Iperf. Hasil pengujian kinerja jaringan SDN membuktikan bahwa tidak ada penurunan yang signifikan dibandingkan dengan arsitektur tradisional pada jaringan komputer. Hal ini dapat dilihat dari parameter nilai QoS delay maksimum jaringan SDN dengan topologi tree sebesar 3,822 ms dan konvensional pada topologi yang sama sebesar 1,867 ms. Pada jaringan SDN topologi Star nilai delay maksimum yang dihasilkan sebesar 1,292 ms dan pada jaringan konvensional pada topologi yang sama sebesar 0,225 ms. Nilai jitter maksimum pada jaringan SDN dengan topologi Tree dan Star sebesar 4,85 ms dan 1,15 ms. Pada jaringan konvensional dengan topologi Tree dan Star nilai jitter maksimum yang dihasilkan sebesar 1,21 ms dan 0,18 ms. Nilai throughput maksimum pada jaringan SDN dengan topologi Tree dan Star adalah sebesar 21,6 Gbit/s dan 21,2 Gbit/s.

Kata Kunci: QoS, SDN, Simulasi, Star, Tree

ABSTRACT

This study aims to analyze the performance of networks using SDN architecture and compared with network performance using traditional architecture. This research was carried out by doing network simulation with SDN and traditional architecture using Mininet emulator software. To measure the performance of both architectures, it can be seen from the QoS values generated using Iperf software. The results of the SDN network performance experiment prove that there is no significant difference compared to traditional architecture on computer networks. This can be seen from the parameter QoS maximum delay network value of SDN with tree topology of 3,822 ms and conventional in the same topology of 1,867 ms. In Star topology SDN network, the maximum delay generated is 1,292 ms and in conventional networks the same topology is 0.225 ms. Maximum jitter value in SDN network with Tree and Star topology of 4.85 ms and 1.15 ms. In conventional networks with Tree and Star topologies, the maximum jitter produced is 1.21 ms and 0.18 ms. The maximum throughput value in the SDN network with Tree and Star topology is 21.6 Gbit/s and 21.2 Gbit/s.

Keywords: QoS, SDN, Simulasi, Star, Tree

I. PENDAHULUAN

Jaringan komputer memiliki sifat yang dinamis dan kompleks, tidak mengherankan apabila pengelolaanya menjadi sebuah tantangan. Operator jaringan harus mengkonfigurasi perangkat jaringan secara manual melalui *Command Line Interface* (CLI). Hal ini menjadi sangat kompleks pada sebuah jaringan komputer yang besar karena akan banyak perangkat jaringan yang harus dikonfigurasi. Kondisi jaringan yang dinamis juga membuat operator jaringan harus sering menyesuaikan dan mengkonfigurasi perangkat jaringan. Tidak menutup kemungkinan apabila operator jaringan melakukan kesalahan karena kondisi jaringan yang memang begitu besar. Oleh karena itu operator

jaringan memerlukan cara yang lebih baik untuk mengonfigurasi dan mengelola jaringan mereka. Sayangnya, *software* dan pengembangan dari perangkat jaringan berbeda-beda tiap vendor dan biasanya sifatnya tertutup tergantung dari perusahaan. Hal ini tentu saja menyulitkan operator jaringan dalam mengembangkan sebuah protokol baru yang sesuai dengan jaringan mereka kelola.

Software Defined Network (SDN) adalah sebuah paradigma di mana sebuah program pusat, atau disebut *controller*, akan mengatur sistem jaringan secara keseluruhan [1]. Konsep *Software Defined Network* sendiri adalah memisahkan antara *control plane* dengan *data plane*. *Control plane* akan dikendalikan secara terpusat melalui *controller* dan peran perangkat jaringan hanya mengirimkan paket data saja. Dengan menggunakan *Software*

Defined Network dan *OpenFlow*, operator akan dengan mudah untuk menerapkan ide-ide baru mereka melalui sebuah *software* secara terbuka. Kendali sistem jaringan yang terpusat juga memudahkan operator jaringan, karena tidak perlu mengkonfigurasi semua perangkat jaringan secara individual.

Dengan penerapan arsitektur jaringan *Software Defined Network*, diharapkan mampu digunakan untuk menganalisis performa jaringan yang menggunakan arsitektur ini dan membantu dalam proses pengembangannya.

II. TINJAUAN PUSTAKA

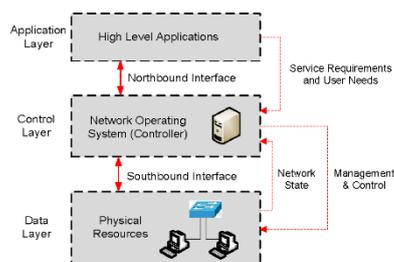
A. Software-Defined Networking

Software Defined Network (SDN) adalah sebuah konsep yang memungkinkan/memperbolehkan operator jaringan untuk mengelola *router* dan *switch* secara fleksibel menggunakan *software* yang berjalan di *server* eksternal [1]. Sedangkan [2] mendefinisikan SDN adalah sebuah arsitektur jaringan yang baru dimana kontrol jaringan dipisahkan dari *forwarding* dan diprogram secara langsung.

Beberapa aspek penting dari *Software Defined Network* adalah:

1. Adanya pemisahan secara fisik/eksplisit antara *forwarding/data-plane* dan *control-plane*.
2. Antarmuka standard (*vendor-agnostic*) untuk memprogram perangkat jaringan.
3. *Control-plane* yang terpusat (secara logika) atau adanya sistem operasi jaringan yang mampu membentuk peta logika (*logical map*) dari seluruh jaringan dan kemudian memrepresentasikannya melalui *Application Programming Interface* (API).
4. Virtualisasi dimana beberapa sistem operasi jaringan dapat mengontrol bagian-bagian (*slices* atau *substrates*) dari perangkat yang sama.

Arsitektur SDN terdiri dari tiga layer yaitu sebagai berikut, seperti tampak pada Gambar II.1:



Gambar III.1 Arsitektur SDN

1. *Application Plane*: berada pada lapisan teratas, berupa aplikasi yang dapat secara langsung dan eksplisit mendefinisikan *network requirement* dan *network behavior* yang diinginkan.
2. *Controller Plane*: yaitu entitas kontrol yang memiliki tugas yaitu mentranslasikan *network requirement* yang telah didefinisikan oleh *Application Layer* menjadi instruksi-instruksi yang sesuai untuk *Infrastructure Layer*, dan memberikan *abstract view* yang dibutuhkan bagi *Application Layer* (*abstract view* meliputi informasi statistik dan *event* yang terjadi di jaringan).
3. *Data Plane*: terdiri dari elemen jaringan yang dapat menerima instruksi dari *Control Layer*.
4. *Northbound API*: SDN menggunakan *northbound API* untuk berkomunikasi dengan *application layer*. *Northbound API* juga membantu operator dalam mengimplementasikan *software* baru.
5. *Southbound API*: SDN menggunakan *southbound API* untuk menyampaikan informasi ke *switch* dan *router*. *OpenFlow* merupakan salah satu protokol yang digunakan pada *southbound API* dan menjadi bagian umum dari SDN.

Control Plane merupakan otak dari jaringan SDN, dapat dijalankan secara terpisah dari *Data Plane*. Sedangkan *Data Plane* merupakan perangkat-perangkat keras jaringan yang terprogram secara khusus dan dikendalikan penuh oleh *Control Plane*. Pada SDN tersedia *open interface* yang memungkinkan sebuah entitas *software* atau aplikasi untuk mengendalikan konektivitas dari sumber daya jaringan, mengendalikan aliran lalu lintas paket data, serta melakukan inspeksi atau memodifikasi lalu lintas paket data tersebut.

B. OpenFlow

OpenFlow adalah sebuah protokol standar yang digunakan dalam komunikasi antar *control-plane* dan *data-plane* [3]. OpenFlow mendefinisikan infrastruktur *flow-based forwarding* dan *Application Programming Interface* (API) standar yang memungkinkan controller untuk mengarahkan fungsi dari switch melalui saluran yang aman (*secure channel*) [4].

Untuk bisa menggunakan *OpenFlow* diperlukan *controller* SDN yang mendukung jalannya protokol *OpenFlow*. *POX controller* adalah salah satu *controller* SDN yang mendukung protokol *OpenFlow*. *POX* adalah *controller* yang berbasis bahasa *Python*, dan dapat dijalankan pada sistem operasi *Windows*, *MacOS* dan *Linux*. *POX* masih dalam tahap pengembangan, dan merupakan

perkembangan dari *controller* terdahulu yaitu NOX (berbasis bahasa C). Dalam jaringan SDN, *controller* harus diaktifkan dahulu sebelum menjalankan jaringan itu sendiri.

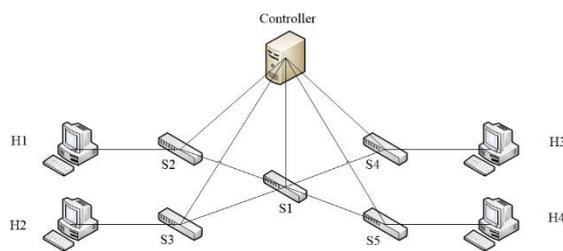
C. Mininet

Mininet adalah suatu *software emulator* yang memungkinkan untuk melakukan percobaan pada jaringan yang luas dengan hanya menggunakan satu mesin. Mininet digunakan untuk mengemulasi *data plan* atau *infrastructure layer* dalam perancangan jaringan SDN [5].

III. PERANCANGAN SISTEM

1. Pemodelan Jaringan dengan Topologi Star

Pada topologi ini digunakan satu *controller* dan lima *switch* yang masing-masing *switch* tersebut terhubung pada satu *switch* pusat. Pada setiap *switch* S2 sampai S5 terdapat satu *host* PC yang terhubung.



Gambar 3.1 Sistem Jaringan dengan Topologi Star

Pada Tabel 3.1 akan menjelaskan pengalaman IP pada masing-masing perangkat jaringan yang digunakan dalam sistem jaringan Topologi Star:

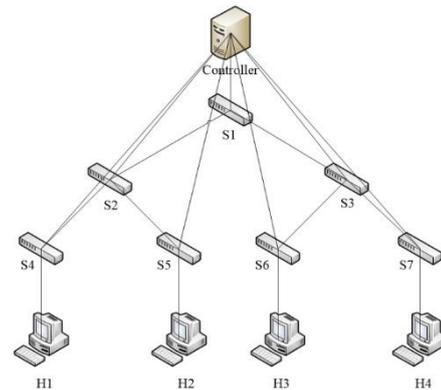
Tabel 3.1 Pengalaman IP pada Sistem Jaringan Topologi Star

Perangkat	Alamat IP	Subnet Mask
Controller	127.0.0.1	255.0.0.0
H1	10.0.1.2	255.255.0.0
H2	10.0.2.2	255.255.0.0
H3	10.0.3.2	255.255.0.0
H4	10.0.4.2	255.255.0.0

Pada topologi jaringan Star masing-masing *host* PC menggunakan alamat IP *private* dengan *subnet* 10.0.0.0/16. Sedangkan *controller* menggunakan alamat IP *local loopback* yaitu 127.0.0.1/8.

2. Pemodelan Jaringan dengan Topologi Tree

Pada topologi ini digunakan satu *controller* dan tujuh *switch* yang terhubung layaknya percabangan, kemudian pada masing-masing *switch* S4 sampai S7 terhubung dengan satu *host* PC.



Gambar 3.2 Sistem Jaringan dengan Topologi Tree

Pada Tabel 3.2 akan menjelaskan pengalaman IP pada masing-masing perangkat jaringan yang digunakan dalam sistem jaringan Topologi Tree:

Tabel 3.2 Pengalaman IP pada Sistem Jaringan Topologi Tree

Perangkat	Alamat IP	Subnet Mask
Controller	127.0.0.1	255.0.0.0
H1	10.0.1.2	255.255.0.0
H2	10.0.2.2	255.255.0.0
H3	10.0.3.2	255.255.0.0
H4	10.0.4.2	255.255.0.0

Pada topologi jaringan Star masing-masing *host* PC menggunakan alamat IP *private* dengan *subnet* 10.0.0.0/16. Sedangkan *controller* menggunakan alamat IP *local loopback* yaitu 127.0.0.1/8.

IV. PENGUJIAN DAN ANALISIS

Pengujian dilakukan untuk mengetahui perilaku jaringan virtual SDN serta performansinya, diukur dari *delay*, *jitter*, dan *throughput*. *Tools*/perintah yang digunakan untuk ini yaitu *ping* dan *Iperf*. Perintah *ping* digunakan untuk mencari nilai *delay* dan *jitter*, sedangkan perintah *Iperf* digunakan untuk mengukur *throughput*.

A. Pengukuran Delay

Nilai rata-rata *delay* yang diperoleh dari hasil simulasi jaringan SDN menggunakan topologi tree cenderung dinamis. Sedangkan nilai rata-rata *delay* yang diperoleh dari hasil simulasi jaringan konvensional dengan topologi tree lebih konsisten. Jumlah ping dalam satu skenario percobaan sebanyak sepuluh kali ping. Tabel 4.1 menunjukkan hasil pengujian dari pengukuran nilai rata-rata *delay* antara jaringan SDN dan konvensional.

Tabel 4.3 Hasil Pengujian Delay pada Topologi Tree

No.	Skenario Ping	Delay (ms)	
		Jaringan SDN	Jaringan Konv.
1	h1 ping h2	0,656	1,762
2	h1 ping h3	3,224	1,259
3	h1 ping h4	3,822	1,857
4	h2 ping h1	1,029	1,479
5	h2 ping h3	3,176	1,262
6	h2 ping h4	2,304	1,457
7	h3 ping h1	1,876	1,171
8	h3 ping h2	0,734	1,867
9	h3 ping h4	1,206	1,358
10	h4 ping h1	2,011	1,670
11	h4 ping h2	2,001	1,452
12	h4 ping h3	0,769	1,477

Nilai rata-rata *delay* yang diperoleh dari hasil simulasi jaringan SDN menggunakan topologi Star cenderung lebih besar dibandingkan jaringan konvensional dengan topologi yang sama. Hal ini terlihat pada Tabel 4.2 yang menunjukkan *delay* maksimum pada jaringan SDN dengan topologi star yaitu sebesar 1,292 ms terjadi saat host 3 melakukan ping pada host 4. Dan pada skenario ping lainnya nilai *delay* yang didapat terlihat konstan tidak lebih dari 1,3 ms. Sedangkan pada jaringan konvensional dengan topologi star, nilai *delay* maksimum yang didapat hanya sebesar 0,225 ms pada saat hos 2 melakukan ping pada host 1.

Tabel 4.4 Hasil Pengujian Delay pada Topologi Star

No.	Skenario Ping	Delay (ms)	
		Jaringan SDN	Jaringan Konv.
1	h1 ping h2	0,153	0,102
2	h1 ping h3	1,020	0,148
3	h1 ping h4	1,086	0,167
4	h2 ping h1	0,515	0,225
5	h2 ping h3	1,124	0,165
6	h2 ping h4	1,243	0,160
7	h3 ping h1	1,176	0,134
8	h3 ping h2	1,267	0,112
9	h3 ping h4	1,292	0,140
10	h4 ping h1	1,126	0,142
11	h4 ping h2	1,166	0,163
12	h4 ping h3	1,227	0,168

Jika dibandingkan secara keseluruhan, topologi Star menghasilkan nilai delay yang lebih kecil dibandingkan dengan topologi Tree. Pada topologi Tree nilai delay minimum didapatkan pada pengujian jaringan SDN yaitu sebesar 0,656 ms, sedangkan pada jaringan konvensional adalah sebesar 1,171 ms. Pada pengujian topologi Star nilai delay minimum yang didapatkan adalah sebesar

0,102 ms yang dihasilkan dalam jaringan konvensional, sedangkan pada jaringan SDN adalah sebesar 0,153 ms. Dari keseluruhan hasil pengujian, nilai delay yang diperoleh masih berada dalam kategori standar rekomendasi ITU-T yaitu antara 0 sampai 150 ms.

B. Pengukuran Jitter

Nilai rata-rata *jitter* yang didapatkan dari hasil simulasi jaringan SDN menggunakan topologi Tree terlihat lebih tinggi dibandingkan dengan rata-rata *jitter* pada simulasi jaringan konvensional dengan topologi yang sama. Nilai *jitter* maksimum pada jaringan SDN dengan topologi Tree didapatkan pada saat percobaan ke-10 yang terlampir dalam Tabel 4.3 yaitu sebesar 4,85 ms. Sedangkan pada jaringan konvensional dengan topologi Tree nilai *jitter* maksimum didapatkan pada percobaan ke-12 yaitu sebesar 1,21 ms. Nilai *jitter* tersebut masih berada dalam kategori standar rekomendasi ITU-T yaitu sebesar 0 sampai 50 ms.

Tabel 4.5 Hasil Pengujian Jitter pada Topologi Tree

No.	Skenario Ping	Jitter (ms)	
		Jaringan SDN	Jaringan Konv.
1	h1 ping h2	1,05	0,79
2	h1 ping h3	1,45	0,46
3	h1 ping h4	0,65	0,66
4	h2 ping h1	2,11	0,46
5	h2 ping h3	0,81	0,67
6	h2 ping h4	3,74	0,76
7	h3 ping h1	1,43	0,33
8	h3 ping h2	1,44	0,70
9	h3 ping h4	1,36	0,48
10	h4 ping h1	4,85	0,56
11	h4 ping h2	3,85	0,76
12	h4 ping h3	1,73	1,21

Nilai rata-rata *jitter* yang diperoleh dari hasil simulasi jaringan SDN menggunakan topologi Star tidak berbeda jauh dibandingkan jaringan konvensional dengan topologi yang sama. Hal ini terlihat pada Tabel 4.4 yang menunjukkan *jitter* maksimum pada jaringan SDN dengan topologi star yaitu sebesar 1,15 ms terjadi pada percobaan ke-6. Dan pada percobaan lainnya nilai *jitter* yang didapat terlihat konstan tidak lebih dari 1,2 ms. Sedangkan pada jaringan konvensional dengan topologi star, nilai *jitter* maksimum yang didapat hanya sebesar 0,18 ms pada saat percobaan ke-4.

Tabel 4.6 Hasil Pengujian Jitter pada Topologi Star

No.	Skenario Ping	Jitter (ms)	
		Jaringan SDN	Jaringan Konv.
1	h1 ping h2	0,06	0,05

2	h1 ping h3	0,60	0,07
3	h1 ping h4	0,91	0,10
4	h2 ping h1	0,46	0,18
5	h2 ping h3	0,96	0,12
6	h2 ping h4	1,15	0,09
7	h3 ping h1	1,01	0,08
8	h3 ping h2	1,01	0,04
9	h3 ping h4	0,88	0,07
10	h4 ping h1	0,74	0,08
11	h4 ping h2	0,80	0,07
12	h4 ping h3	1,13	0,09

Jika dibandingkan secara keseluruhan, topologi Tree menghasilkan nilai jitter yang lebih kecil dibandingkan dengan topologi Star. Hal ini disebabkan oleh beberapa faktor yang salah satunya adalah perbedaan jumlah *nodes* antara kedua topologi. Dari keseluruhan hasil pengujian, nilai delay yang diperoleh masih berada dalam kategori standar rekomendasi ITU-T yaitu antara 0 sampai 150 ms.

C. Pengukuran Throughput

Nilai *throughput* yang dihasilkan jaringan SDN dengan topologi tree memiliki nilai yang cukup tinggi dibandingkan dengan jaringan konvensional seperti yang terlampir dalam Tabel 4.5. Nilai *throughput* maksimum dihasilkan pada saat percobaan ke-1 pada jaringan SDN yaitu sebesar 21,6 Gbits/s.

Tabel 4.7 Hasil Pengujian Throughput pada Topologi Tree

No.	Skenario Ping	Throughput (Gbits/s)	
		Jaringan SDN	Jaringan Konv.
1	h2 ping h1	21,6	12
2	h3 ping h1	10,7	10
3	h4 ping h1	11,9	12
4	h1 ping h2	14,3	19,8
5	h3 ping h2	19,1	17,3
6	h4 ping h2	19,9	18,1
7	h1 ping h3	19,2	17
8	h2 ping h3	18,6	17,7
9	h4 ping h3	20,3	19,7
10	h1 ping h4	15	17,1
11	h2 ping h4	18,6	18,3
12	h3 ping h4	21,2	19,9

Nilai *throughput* yang dihasilkan jaringan SDN dengan topologi Star memiliki nilai yang cukup tinggi dibandingkan dengan jaringan konvensional seperti yang terlampir dalam Tabel 4.6. Nilai *throughput* maksimum dihasilkan pada saat percobaan ke-3 pada jaringan SDN yaitu sebesar 21,7 Gbits/s.

Tabel 4.8 Hasil Pengujian Throughput pada Topologi Star

No.	Skenario Ping	Throughput (Gbits/s)	
		Jaringan SDN	Jaringan Konv.
1	h2 ping h1	21,4	13,4
2	h3 ping h1	20,2	15,3
3	h4 ping h1	21,7	15,2
4	h1 ping h2	18,6	16,3
5	h3 ping h2	20,3	14,6
6	h4 ping h2	21,4	15,8
7	h1 ping h3	14,6	20,2
8	h2 ping h3	15,7	20,9
9	h4 ping h3	14,3	20,3
10	h1 ping h4	20,7	19,5
11	h2 ping h4	21	20,3
12	h3 ping h4	21,2	16,5

V. KESIMPULAN DAN SARAN

A. Kesimpulan

1. Kinerja jaringan yang menggunakan arsitektur SDN masih memenuhi standar dari ITU-T.
2. Bila dibandingkan dengan jaringan konvensional, jaringan dengan arsitektur SDN masih berada dalam level kinerja yang sama.
3. Hasil QoS yang didapat tidak menunjukkan penurunan yang signifikan.

B. Saran

1. Menambah jumlah node-node jaringan atau menambah skenario pengujian jaringan.
2. Membandingkan kinerja jaringan SDN dengan dua controller yang berbeda.

DAFTAR PUSTAKA

- [1] "Forwarding and Control Element Separation (Forces) Framework," IETF Working Group, [Online]. Available: <http://www.ietf.org/html.charters/forcescharter.html/>.
- [2] "Software-Defined Networking: The New Norm for Networks," Open Networking Foundation, 2012. [Online]. Available: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>. [Diakses 24 April 2018].
- [3] A. Mukhtarom, A. Basuki dan M. Aswin, "Pemerataan Utilisasi Jaringan Multipath dengan Mode Controller Proactive-Reactive pada Software Defined Networking," Jurnal EECCIS, vol. 11, no. 2, p. 61, 2017.

- [4] "SDN," Cisco, 5 April 2014. [Online]. Available: https://www.cisco.com/web/ANZ/cisco-live/attend/hot_topics/sdn.html. [Diakses 24 April 2018].
- [5] R. Tulloh, R. M. Negara dan A. N. Hidayat, "Simulasi Virtual Local Area Network (VLAN) Berbasis Software Defined Network (SDN) Menggunakan POX Controller," Jurnal Infotel, vol. 7, no. 2, p. 131, 2015.