

# Perbandingan Unjuk Kerja Jaringan pada Arsitektur Software Defined Network dan Konvensional Menggunakan Router Mikrotik Rb-750 dan Emulator Mininet

R.W. Tri Hartono\*, T.B. Utomo, Andry Haidar, Usman B, Hanafi, Nadia I. Kirana

<sup>1</sup>Program Studi Teknik Telekomunikasi, Politeknik Negeri Bandung  
Jl. Gegerkalong Hilir Ds Ciwaruga, Kotak pos 1234, telp (022) 2013789  
rwtri\_h@yahoo.com\*

**Abstrak** – *Software-Defined Network (SDN)* menawarkan kemudahan dalam mengatur perangkat jaringan dengan hanya mengatur *software control plane* sehingga mampu mengurangi kompleksnya konfigurasi jaringan. Kemudahan yang ditawarkan ini belum tentu diikuti dengan peningkatan performa jaringan jika dibandingkan dengan arsitektur jaringan konvensional yang menggunakan protokol TCP/IP. Pada makalah ini dilakukan analisis perbandingan unjuk kerja jaringan dengan arsitektur SDN (*OpenFlow*) dan arsitektur konvensional (TCP/IP). Pengujian unjuk kerja dilakukan menggunakan router Mikrotik sebagai perangkat kerasnya dan Emulator Mininet sebagai perangkat lunaknya. Hasil dari pengukuran jaringan pada arsitektur *Software Defined Network* menunjukkan nilai delay yang lebih rendah dibandingkan dengan jaringan konvensional karena *Software Defined Network* menggunakan sistem terdistribusi (*controller*) dan sistem tersebut dapat langsung mengkalkulasi jalur dari setiap *node*.

**Kata kunci:** Mikrotik, QoS, Router, SDN, TCP/IP

## 1. Pendahuluan

*Software Defined Network* adalah konsep pendekatan baru pada arsitektur jaringan. Berbeda dengan arsitektur tradisional yang menggunakan protokol TCP/IP dimana *control plane* dan *data plane* berdampingan pada perangkat jaringan seperti *switch* dan *router*, sedangkan konsep SDN dengan protokol *OpenFlow* pada dasarnya memisahkan antara *control plane* dan *data plane*. *Control plane* dipindahkan keluar dari perangkat jaringan sehingga hanya *data plane* saja yang berada di dalam perangkat jaringan [1],[2].

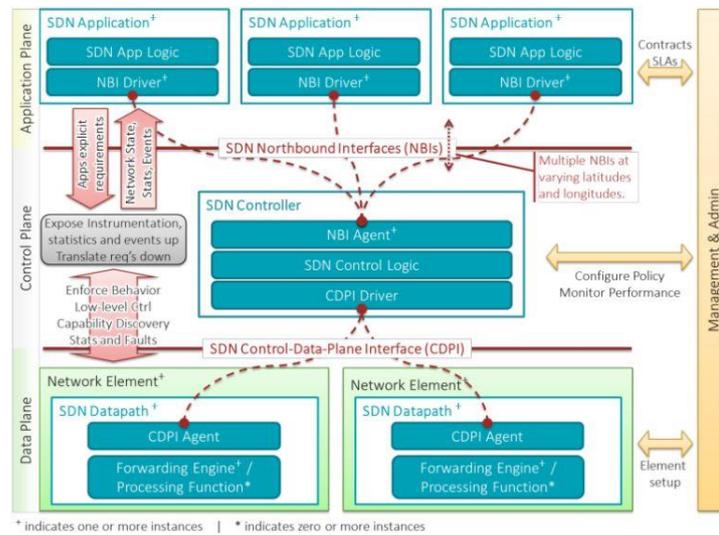
Dalam penelitian dilakukan analisa seberapa besar pengaruh penggunaan arsitektur SDN terhadap performa jaringan dengan parameter yang dianalisis adalah *latency*, *throughput*, *jitter* dan *packet loss*. Dari hasil pengamatan dan analisa diharapkan dapat diketahui perbandingan unjuk kerja dari jaringan pada arsitektur SDN (*OpenFlow*) dan arsitektur konvensional (TCP/IP).

### 1.1. Software Defined Network

*Software Defined Network* (SDN) memiliki kemampuan untuk mengelola, mendesain, dan mengimplementasikan sebuah arsitektur pada jaringan yang berfungsi untuk memisahkan sistem control (*control plane*) dan sistem forwarding (*forwarding plane*). *Software Defined Network* (SDN) juga merupakan solusi masalah jaringan untuk mengefisienkan layanan aplikasi yang dapat disediakan baik secara lokal ataupun terpusat bahkan dapat melalui *cloud* [3], [5].

Seluruh perangkat dapat diolah melalui satu media dimana media tersebut melewati satu protokol. *Software Defined Network* (SDN) dapat digunakan untuk konfigurasi *switch*, *router*, dan seluruh perangkat jaringan hanya dengan satu aplikasi saja. Pada konsep *Software Defined Network* (SDN), terdapat *open interface* yang dapat mengendalikan konektivitas yang tersedia pada sejumlah sumber daya jaringan dan dapat juga mengendalikan aliran trafik serta inspeksi

ataupun memodifikasi trafik tersebut [6]. Gambar 1 menunjukkan ilustrasi arsitektur *Software Defined Network* (SDN) berikut komponen dan interaksinya.



Gambar 1. Ilustrasi dari Arsitektur SDN

Arsitektur dari *Software Defined Network* (SDN) dapat didefinisikan pada 3 lapisan/bidang, yaitu:

**1. Lapisan Infrastruktural (data plane / infrastructure layer)**

Layer ini berisi elemen jaringan yang dapat mengatur seluruh *data path* dari SDN sesuai dengan instruksi dari *Control Data Plane Interface* (CDPI).

**2. Lapisan Kontrol (control plane / layer)**

Lapisan ini dapat mentranslasikan kebutuhan dari aplikasi disertai infrastruktur dengan memberi instruksi yang sesuai untuk *data path* dari SDN. Selain itu, lapisan ini dapat memberikan informasi yang tepat dan dibutuhkan oleh aplikasi SDN.

**3. Lapisan Aplikasi (application plane / layer)**

Layer ini terdapat pada lapisan terluar atau teratas pada SDN, fungsi dari lapisan ini yaitu dapat berkomunikasi dengan sistem melalui *NorthBound Interface* (NBI).

**1.2. Protokol OpenFlow**

Protokol *OpenFlow* merupakan protokol yang dapat berfungsi membantu *server* untuk memberitahu jaringan ke perangkat mana paket akan dikirimkan. Pada jaringan konvensional setiap perangkat memiliki aplikasi yang berfungsi untuk memberitahu apa yang harus dilakukan, protokol *OpenFlow* disini dapat membuat paket bergerak secara terpusat, sehingga seluruh perangkat dapat dikonfigurasi atau diprogram secara terpusat dari sebuah perangkat independen dari *data center*. [6], [8]

Pada jaringan konvensional data pada jalur data (*packet forwarding*) dan jalur kendali atau kontrol (*routing*) terdapat pada perangkat yang sama, berbeda dengan *OpenFlow* yang dapat memisahkan dan membedakan antara jalur kendali atau kontrol. Seluruh perangkat dan *controller* akan berkomunikasi melalui protokol *OpenFlow* ini, metode tersebut dikenal sebagai *Software Defined Network* (SDN).

Menggunakan arsitektur *Software Defined Network* (SDN) seluruh pengguna tidak harus bergantung pada vendor ataupun produk dengan merk tertentu pada implementasi di dalam jaringan. *Software Defined Network* (SDN) menjadi solusi akan hal tersebut yang menciptakan independensi jaringannya sendiri yaitu sebuah jaringan universal [7].

**1.3. Protokol TCP/IP**

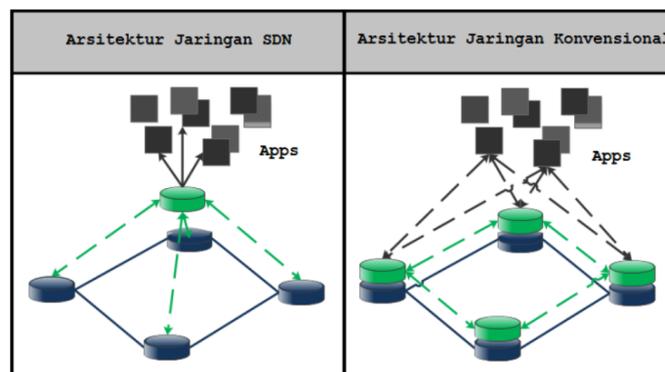
Arsitektur dari protokol *Transmission Control Protocol/Internet Protocol* (TCP/IP) menggunakan model referensi dari DARPA yang terdiri dari 4 lapisan atau *layer*. Lapisan atau *layer* tersebut dapat dipetakan terhadap lapisan atau *layer* pada model referensi OSI. Gambar 2 menunjukkan perbandingan dari model referensi OSI dan TCP/IP.

OSI Layers	TCP/IP Layers	TCP/IP Protocols				
Application Layer	Application Layer	HTTP	FTP	Telnet	SMTP	DNS
Presentation Layer		TCP		UDP		
Session Layer		IP				
Transport Layer	Transport Layer	Ethernet		Token Ring	Other Link-Layer Protocols	
Network Layer	Network Layer					
Data Link Layer	Network Interface Layer					
Physical Layer						

Gambar 2 TCP/IP Layer

**1.4. Perbedaan Arsitektur Jaringan SDN dan Konvensional**

Perbedaan dari arsitektur jaringan pada SDN dan Konvensional dapat terlihat pada gambar 3, pada sisi kiri merupakan arsitektur dari jaringan *Software Defined Network* (SDN) yang memperlihatkan *data plane* dan *control plane* yang terpisah. Pada sisi kanan merupakan arsitektur dari jaringan konvensional yang memperlihatkan *data plane* dan *control plane* berada di tempat yang sama. Hal tersebut menunjukkan jaringan *Software Defined Network* (SDN) menggunakan *control plane* yang tersentralisasi, jika pada jaringan konvensional menggunakan *control plane* yang terdistribusi.[2], [4], [6]



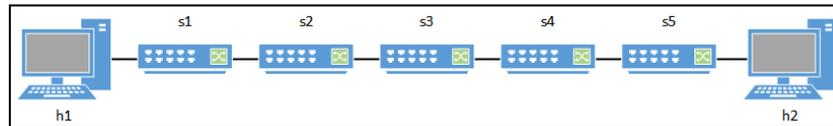
Gambar 3 Perbedaan arsitektur jaringan SDN dan Konvensional

**2. Metoda dan Proses Penyelesaian**

Pada tahap ini, akan dilakukan analisa terhadap kebutuhan sistem yang akan mensimulasikan jaringan baik pada arsitektur SDN maupun konvensional baik secara aplikasi dan menggunakan *router* fisik. Beberapa hal yang dianalisa meliputi kebutuhan perangkat keras dan perangkat lunak agar sistem dapat bekerja dan berfungsi dengan baik.

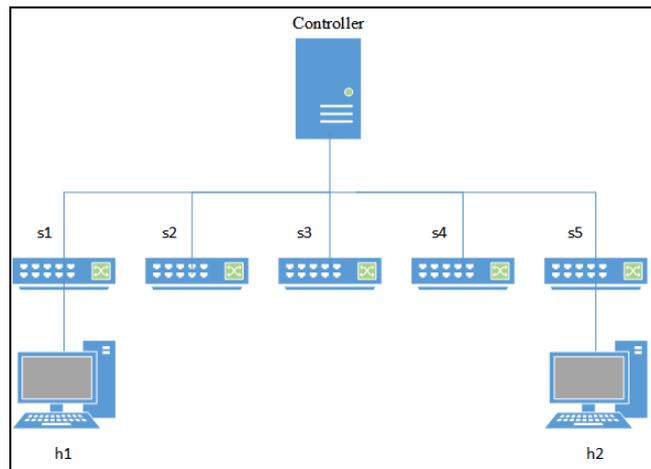
**2.1. Topologi Menggunakan *Emulator Mininet***

Untuk topologi dengan menggunakan *emulator mininet*, dilakukan dengan 5 switch sampai dengan 12 switch. Pada gambar 4 merupakan topologi yang dibangun pada jaringan arsitektur konvensional menggunakan file .py yang dibuat menggunakan aplikasi gedit pada linux.



Gambar 4 Topologi Jaringan Arsitektur Konvensional

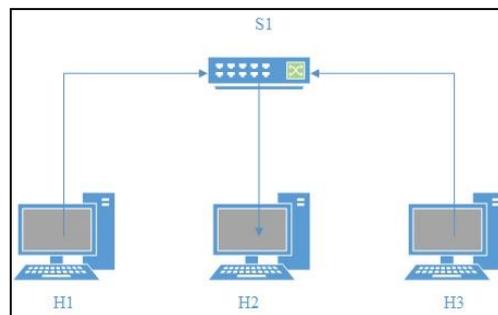
Gambar 5 menunjukkan topologi yang dibangun pada jaringan arsitektur SDN menggunakan file .py yang dibuat menggunakan aplikasi gedit pada linux.



Gambar 5 Topologi Jaringan Arsitektur SDN

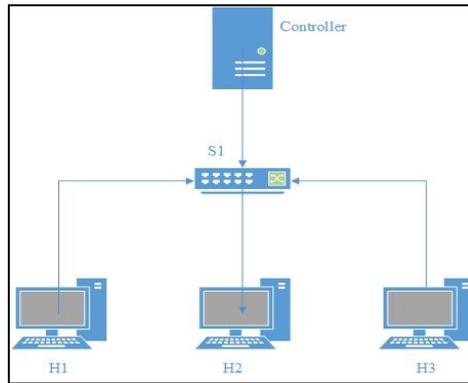
**2.2. Topologi Menggunakan *Router Mikrotik RB-750***

Gambar 6 merupakan topologi jaringan arsitektur konvensional menggunakan Router Mikrotik RB-750 yang dihubungkan ke PC dengan menggunakan kabel LAN.



Gambar 6 Topologi jaringan konvensional menggunakan Router Mikrotik RB-750

Gambar 7 merupakan topologi jaringan arsitektur SDN menggunakan Router Mikrotik RB-750 yang dihubungkan ke PC dengan menggunakan kabel LAN.



Gambar 7 Topologi jaringan SDN menggunakan Router Mikrotik RB-750

3. Hasil dan Analisa

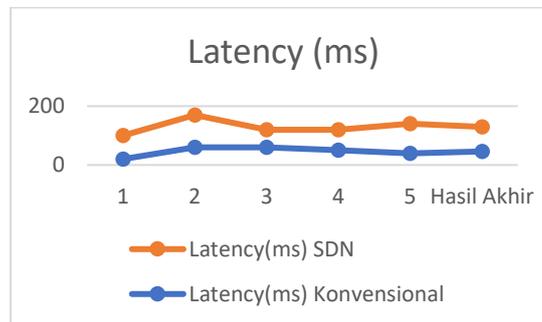
3.1. Pengujian dan Analisa Jaringan Software Defined Network dan Konvensional menggunakan Router Mikrotik RB-750

1. Latency

Tabel 1 Hasil pengukuran Latency menggunakan Router Mikrotik RB-750

No.Percobaan	Latency(ms)	
	Konvensional	SDN
1	20	80
2	60	110
3	60	60
4	50	70
5	40	100
<b>Rata-rata</b>	<b>46</b>	<b>84</b>

Hasil dari pengujian *Latency* tersebut, dapat terlihat bahwa nilai rata-rata *latency* dengan menggunakan jaringan SDN memiliki nilai lebih besar yaitu 84ms. Grafik pengujian *latency* dapat terlihat pada Gambar 8.

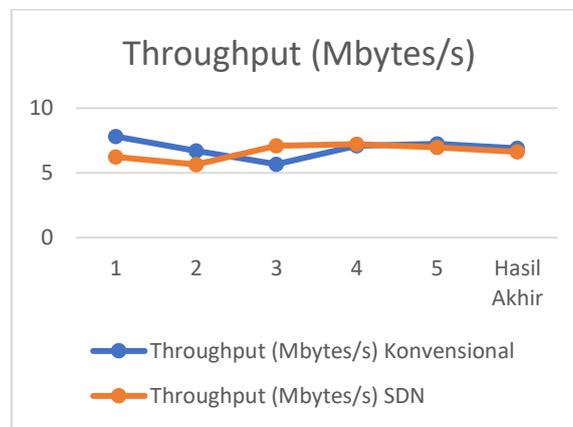


Gambar 8 Grafik hasil pengukuran Latency menggunakan Router Mikrotik RB-750

2. *Throughput*

Tabel 2 Hasil pengukuran Throughput menggunakan Router Mikrotik RB-750

Throughput (Mbytes/s) SDN	Throughput (Mbytes/s) Konvensional
8.16	6.24
6,64	5.65
5.65	7.09
7.22	7.22
6.98	6.98
<b>Rata-rata: 6.93</b>	<b>Rata-rata: 6.64</b>



Gambar 9 Grafik hasil pengukuran Throughput menggunakan Router Mikrotik RB-750

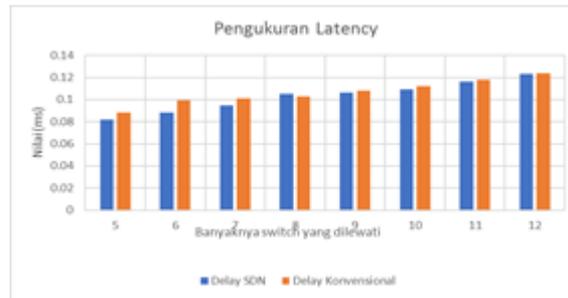
3.2. **Pengujian dan Analisa Jaringan Software Defined Network dan Konvensional menggunakan Emulator Mininet**

1. *Latency*

Tabel 3 Hasil pengukuran Latency menggunakan emulator Mininet

Banyaknya switch yang dilewati	Latency (ms)	
	SDN	Konvensional
5	0.082	0.0885
6	0.0885	0.0995
7	0.0945	0.101
8	0.105	0.103
9	0.1065	0.108
10	0.109	0.112
11	0.116	0.118
12	0.123	0.124
<b>Rata - Rata</b>	<b>0.8245</b>	<b>0.854</b>

Pengujian skenario yaitu melihat unjuk kerja jaringan baik SDN maupun konvensional menggunakan *Emulator Mininet*. Pada Tabel 3 disajikan hasil uji coba Jaringan SDN dan konvensional menggunakan *Emulator Mininet*. Sementara penyajian grafik dari hasil uji Latency menggunakan *Emulator Mininet* dapat ditunjukkan pada Gambar 10.



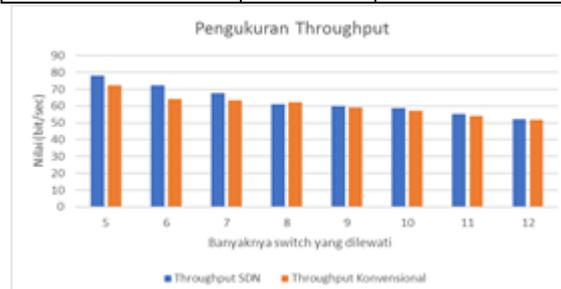
Gambar 10 Grafik hasil pengukuran Latency menggunakan Emulator Mininet

Berdasarkan standar ETSI, nilai Latency yang diperoleh masuk dalam kategori “sangat baik” karena memenuhi standar tersebut. Dapat dilihat pada Tabel 3 dan Gambar 10, nilai *latency* tertinggi dengan arsitektur SDN terjadi pada 12 switch yang dilewati yaitu 0.123 ms dan nilai *latency* tertinggi dengan arsitektur Konvensional terjadi pada 12 switch yang dilewati yaitu 0.124 ms hal ini membuktikan semakin banyaknya switch yang digunakan dapat mempengaruhi performasi dari arsitektur SDN dan Konvensional. Hal ini terlihat dari grafik yang telah disediakan.

2. *Throughput*

Tabel 4 Hasil pengukuran Throughput menggunakan emulator Mininet

Banyaknya switch yang dilewati	Throughput (Mbytes/s)	
	SDN	Konvensional
5	78.048	72.316
6	72.316	64.321
7	67.724	63.366
8	60.952	62.135
9	60.093	59.259
10	58.715	57.142
11	55.172	54.237
12	52.032	51.612
<b>Rata - rata</b>	<b>505.052</b>	<b>484.388</b>



Gambar 11 Grafik hasil pengukuran Throughput menggunakan Emulator Mininet

Gambar 11 Menyajikan grafik hasil uji Throughput menggunakan Emulator Mininet sementara. Tabel 4 dan Gambar 11 menunjukkan bahwa nilai throughput dengan arsitektur SDN memiliki nilai yang lebih tinggi dibandingkan dengan arsitektur Konvensional. Besarnya nilai tersebut dipengaruhi oleh besarnya trafik yang mengalir pada Protokol OpenFlow.

#### 4. Kesimpulan

Penggunaan arsitektur SDN dengan protokol *OpenFlow* tidak menghambat unjuk kerja perangkat. Semakin banyaknya switch yang dilewati, semakin tinggi pula trafik yang dihasilkan. Hal ini berbanding lurus dengan semakin tinggi pemakaian pada jalur jaringan maka semakin besar pula trafik yang mengalir. Pada OpenFlow, *controller* secara default melakukan pemilahan dan membuang paket-paket yang tidak terdefinisi pada Flow Entry. Berbeda dengan Konvensional yang tidak memiliki proses pemilahan paket-paket tersebut. Hal inilah yang menyebabkan arsitektur SDN lebih baik dibandingkan dengan arsitektur Konvensional. Hasil uji coba dan pengukuran jaringan pada arsitektur *Software Defined Network* menunjukkan nilai delay yang lebih rendah dibandingkan dengan jaringan konvensional karena *Software Defined Network* menggunakan sistem tersentral (*controller*) dan sistem tersebut dapat langsung mengkalkulasi jalur dari setiap node. Pengukuran dengan menggunakan Router fisik menunjukkan arsitektur konvensional lebih tinggi karena dengan arsitektur SDN harus melalui kontroler dahulu

#### Daftar Pustaka

- [1] R. Kartadie, E. Utami and E. Pramono, "PROTOTYPE INFRASTRUKTUR SOFTWARE-DEFINED NETWORK DENGAN," *Open Journal Sistem Jurnal Ilmiah Dasi*, 2014. vol. 15, no. 1: 8-14
- [2] Fei Hu ; Qi Hao ; Ke Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation", *IEEE Communications Society*, 2014, Volume: 16 , Issue: 4: 2181 - 2206,
- [3] Carlos J. Bernardos, "An architecture for software defined wireless networking", *IEEE Communications Society*, 2014, Volume: 21 , Issue: 3 : 52-61
- [4] Diego Kreutz, "Software-Defined Networking: A Comprehensive Survey", 2015, Volume: 103 , Issue: 1: 14 - 76
- [5] Ian F.Akyildiz, "SoftAir: A software defined networking architecture for 5G wireless systems", *ELSEVIER*, 2015, volume 85, 5: 1-18
- [6] R. Kartadie and B. Satya, "UJI PERFORMA IMPLEMENTASI SOFTWARE-BASED OPENFLOW SWITCH BERBASIS OPENWRT PADA INFRASTRUKTUR," *Open Journal Sistem Jurnal Ilmiah Dasi*, 2015, vol. 16, no. 3, pp. 8-9.
- [7] Wenfeng Xia, Yonggang Wen, "A Survey on Software-Defined Networking" *IEEE Explore Digital Library*, Volume: 17 , Issue: 1: 27 - 51
- [8] Danda B. Rawat ; Swetha R. Reddy, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey", *IEEE Communications Surveys & Tutorials*, 2017, Volume: 19 , Issue: 1: 325 - 346