

Penerapan *Software Defined Network* Pada MikroTik RouterOS Menggunakan OpenFlow Berbasis ONOS Controller

Application of Software Defined Network on MikroTik RouterOS Using OpenFlow Based on ONOS Controller

Dadang Sumantri Pulungan^{1*}, Ade Rukmana², Bambang Sugiarto³

^{1*2,3} Fakultas Teknik Universitas Garut, Garut, Jawa Barat, 44151, Indonesia

dadangspulungan@gmail.com^{1*}, ade.rukmana@uniga.ac.id², bsugiarto991@gmail.com³

Abstrak – Kesulitan dalam mengkonfigurasi jalur pengiriman data pada jaringan komputer tradisional membuat dibutuhkannya solusi untuk menyelesaikan permasalahan tersebut. Salah satu solusi yang berkembang saat ini adalah SDN. *Software Defined Network* mempermudah konfigurasi perangkat jaringan karena adanya pemisahan pada control plane dan data plane. SDN tidak membutuhkan konfigurasi pada setiap perangkat untuk mengirimkan data. SDN hanya membutuhkan sebuah kontroler untuk mengatur setiap pengiriman data sehingga memudahkan saat mengkonfigurasi jaringan. Akan tetapi SDN masih harus diuji dalam kehandalan mengirimkan data. Pada penelitian ini dilakukan penerapan SDN dan pengujian QoS SDN pada MikroTik RouterOS dengan kontroler ONOS. Ada 4 parameter QoS yang diujikan yaitu jitter, throughput, delay dan packet loss. Pada pengujian QoS SDN di MikroTik didapatkan nilai QoS yang sangat bagus pada setiap parameter yang diujikan dengan nilai indeks 4. Hasil perbandingan QoS SDN terhadap QoS jaringan komputer tradisional menunjukkan QoS SDN lebih baik daripada QoS jaringan komputer tradisional.

Kata Kunci: QoS, SDN, jaringan, MikroTik, ONOS.

Abstract – The difficulty in configuring data transmission paths on traditional computer networks makes a solution to solve these problems needed. One solution that is currently developing is SDN. *Software Defined Network* simplifies configuration of network devices due to the separation of the control plane and data plane. SDN does not require any configuration on each device to transmit data. SDN only requires a controller to manage each data transmission, making it easier to configure the network. However, SDN still has to be tested in the reliability of sending data. In this study, the implementation of SDN and SDN QoS testing on MikroTik RouterOS with ONOS controller was carried out. There are 4 tested QoS parameters, namely jitter, throughput, delay and packet loss. In the SDN QoS test at MikroTik, a very good QoS value was obtained for each parameter tested with an index value of 4. The results of the comparison of SDN QoS against traditional computer network QoS showed SDN QoS was better than traditional computer network QoS.

Keywords: QoS, SDN, network, MikroTik, ONOS.

1. Pendahuluan

Salah satu kompleksitas pada jaringan tradisional disebabkan belum dipisahkannya *data plane* dan *control plane* pada forwarding data [1]. Pada jaringan tradisional, administrator jaringan harus mengkonfigurasi *control plane* dan *data plane* untuk pengiriman paket data pada masing-masing perangkat fisik jaringan seperti router maupun switch router. Ini membuat routing jaringan kurang efektif karena harus dikonfigurasi pada setiap router.

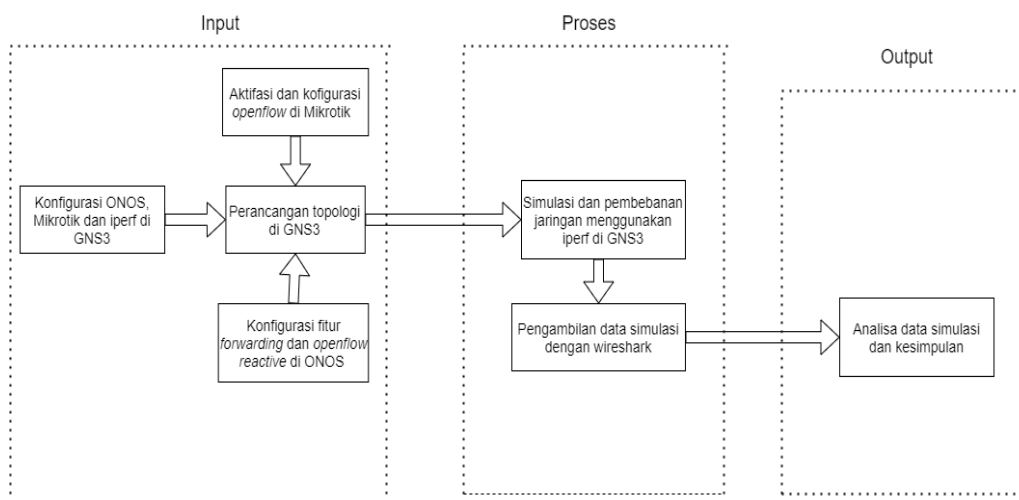
Pemisahan *control plane* dan *data plane* dapat mempermudah routing jaringan karena perangkat fisik pada jaringan seperti router dapat diatur oleh satu alat pengontrol. Pemisahan tersebut membuat administrator jaringan tidak perlu untuk mengatur protokol maupun rute paket pada setiap routernya. Pemisahan *control plane* dan *data plane* pada jaringan komputer terdapat

pada konsep jaringan SDN. *Software defined network* (SDN) merupakan konsep jaringan yang memisahkan antara control plane dan data plane. Pemisahan ini terjadi karena adanya protokol *Openflow* dan kontroler SDN. *Openflow* bertugas sebagai pengganti *data plane* dan kontroler SDN bertanggung jawab atas *control plane*. Sehingga setiap pengaturan paket data yang dikirimkan dapat diatur secara terpusat oleh kontroler dan router hanya berperan sebagai *data plane* yang mengalirkan paket data [1].

Penerapan SDN akan dilakukan pada perangkat jaringan yang banyak digunakan saat ini yaitu router MikroTik. Router Mikrotik masih menggunakan konsep dan protokol jaringan tradisional. Untuk menjadikan MikroTik sebagai switch SDN, diperlukan instalasi paket openflow. Paket openflow digunakan untuk mengaktifkan protocol openflow pada MikroTik sehingga dapat memisahkan *data plane* dan *control plane* pada MikroTik. MikroTik hanya mendukung penggunaan protokol openflow versi 1.0 saja. Protokol openflow pada router MikroTik tidak diaktifkan secara *defult* sehingga harus diaktifkan terlebih dahulu. MikroTik yang sudah menjadi switch SDN kemudian akan dikontrol menggunakan kontroler SDN yaitu ONOS. Penerapan SDN ini kemudian akan diuji melalui parameter *Quality of Service* (QoS) jaringan yaitu *throughput*, *jitter*, *delay* dan *packet loss*. Hasil pengujian QoS SDN kemudian akan dibandingkan dengan hasil dari pengujian QoS jaringan tradisional yang menggunakan router MikroTik dan QoS dari simulator mininet dengan topologi yang sama. Perbandingan ini untuk mengukur bagaimana performansi penerapan SDN pada MikroTik terhadap performansi jaringan tradisional dan performansi SDN menggunakan Simulator mininet.

2. Metode Penelitian

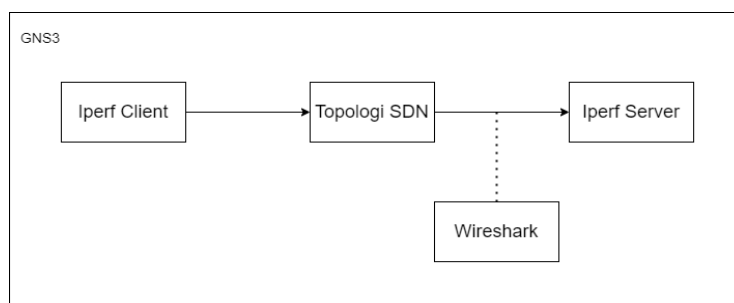
Penerapan SDN pada MikroTik dimulai dengan menginstall simulator jaringan GNS3 di PC untuk menjadi wadah virtualisasi setiap komponen penelitian ini. GNS3 yang sudah diinstall kemudian di konfigurasi supaya terhubung dengan pc hostnya. Setelah instalasi GNS3 kemudian setiap komponen yang dibutuhkan pada simulasi ini diinstallkan di GNS3. Komponen yang sudah diinstall kemudian di setting supaya siap digunakan dalam simulasi.



Gambar 1. Blok Diagram Sistem

Tahapan selanjutnya adalah Menyusun setiap komponen sesuai dengan topologi yang sudah dirancang. Pada penelitian ini, penulis menggunakan topologi tree dengan 7 router dan 12 host. Kemudian router MikroTik akan diubah menjadi switch SDN dengan menerapkan protocol Openflow di MikroTik. MikroTik yang sudah menjadi switch SDN akan dihubungkan dengan *controller* ONOS. Kontroler ONOS dikonfigurasi terlebih dahulu agar dapat mengatur paket pada

switch SDN MikroTik. ONOS akan menjadi pengatur paket di dalam jaringan SDN. Setelah MikroTik sudah terhubung dengan ONOS, di setiap host akan diberi alamat ip untuk dapat terhubung dengan SDN MikroTik. Di setiap host akan menggunakan metode ping untuk pengetesan jaringan.



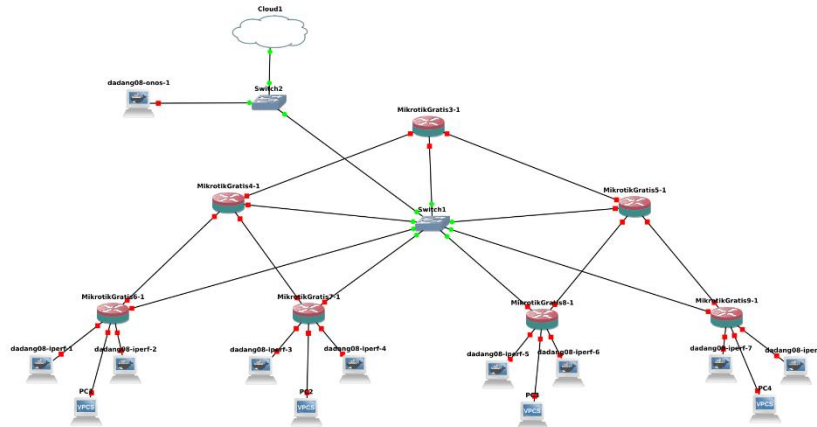
Gambar 2. Blok Diagram Pengujian Jaringan

Setelah semua sudah terhubung, SDN di MikroTik akan diberikan load trigger melalui software iperf[7]. Besaran data yang akan diuji diberi secara default oleh iperf. Kemudian beban yang dikirimkan oleh iperf ke jaringan SDN MikroTik akan dianalisa kualitas jaringannya menggunakan Wireshark menurut parameter QoS. Hasil Analisa pada SDN MikroTik kemudian akan dibandingkan dengan hasil analisa dari jaringan tradisional menggunakan router MikroTik dan SDN menggunakan emulator mininet yang sudah disimulasikan sebelumnya.

2.1. Prosedur Pengerjaan Penelitian

Berikut prosedur pekerjaan dalam penelitian ini: 1) Memasang GNS3. GNS3 digunakan sebagai wadah simulasi SDN di MikroTik menggunakan OpenFlow dan ONOS Controller dan juga simulasi untuk jaringan tradisional. 2) Memasang VirtualBox. VirtualBox digunakan untuk menjalankan MikroTik RouterOS secara virtual agar dapat disimulasikan di GNS3. 3) Memasang MikroTik RouterOs pada VirtualBox. 4) Memasang Docker. Docker digunakan untuk membuat docker image ONOS dan docker image Iperf agar ONOS dan Iperf dapat dijalankan di GNS3. 5) Membuat docker image ONOS dan docker image Iperf. ONOS digunakan sebagai kontroler untuk SDN dan Iperf digunakan untuk membebani jaringan di SDN agar dapat diuji kehandalannya. 6) Memasang Python. Python digunakan untuk menjalankan mininet, membuat topologi pada mininet dan menghitung jitter. 7) Memasang Mininet. Mininet digunakan sebagai simulator SDN. Mininet dapat menggambarkan jaringan SDN secara ideal. 8) Menambahkan dan Mengkonfigurasi MikroTik RouterOS, ONOS dan Iperf ke GNS3. 9) Membuat Topologi yang telah dirancang pada GNS3 dan di Mininet. 10) Memasang OpenFlow pada MikroTik RouterOS. OpenFlow digunakan untuk mengubah MikroTik RouterOS menjadi switch pada SDN. 11) Konfigurasi OpenFlow yang telah dipasang di MikroTik RouterOS. 12) Penambahan paket-paket aplikasi pada ONOS supaya ONOS dapat terhubung dengan OpenFlow. 13) Tes Koneksi dari ONOS ke OpenFlow dan tes koneksi antar host/Iperf. 14) Pengujian pembebanan pada SDN dengan Iperf. 15) Pengambilan data pengujian dengan wireshark. Perhitungan parameter QoS dari data yang telah diambil menggunakan Wireshark.

2.2. Desain Topologi



Gambar 3. Topologi Jaringan

Didalam topologi yang akan dibuat di GNS3, komponen-komponen yang digunakan adalah cloud, kontroler ONOS, switch, MikroTik RouterOS, iperf dan vpcs. Gambar diatas terdapat 1 kontroler ONOS, 2 switch, 7 router MikroTik RouterOS yang akan dijadikan switch SDN dan 12 host. 12 host terdiri dari 10 vpcs dan 2 iperf berbasis docker yang akan dijadikan sebagai server dan client yang digunakan untuk menguji QoS SDN di MikroTik.

2.3. Alat dan Bahan

Tabel 1. Alat dan Bahan

No	Perangkat	Kegunaan
1.	Laptop/ <i>Personal Computer</i>	Spesifikasi dari laptop/personal computer minimal dapat diinstallkan semua perangkat lunak yang dibutuhkan.
2.	MikroTik RouterOS	Desain perancangan database yang dipakai dalam penelitian
3.	ONOS	Kontroler MikroTik RouterOS yang sudah dijadikan sebagai switch SDN. ONOS berfungsi sebagai control plane SDN untuk mengatur paket data plane di switch SDN.
5.	QEMU	Qemu digunakan sebagai mesin virtual untuk router MikroTik RouterOS di GNS3.
7.	VPCS	VPCS digunakan sebagai host virtual untuk menguji jalannya paket pada SDN.
8.	Mininet	Mininet digunakan sebagai pembanding terhadap kualitas dari penerapam SDN di MikroTik
9.	GNS 3	Perangkat lunak untuk mendesain topologi jaringan internet yang dirancang, serta mensimulasikannya
10.	Iperf	Iperf adalah alat untuk mengukur <i>bandwidth</i> dan kualitas link jaringan, tersedia pada sistem Linux, berukuran kecil dan dapat diinstall dengan mudah. Iperf digunakan untuk ujicoba QoS pada SDN di MikroTik.
11.	Wireshark	Wireshark digunakan sebagai alat analisa kualitas SDN di MikroTik menurut parameter QoS

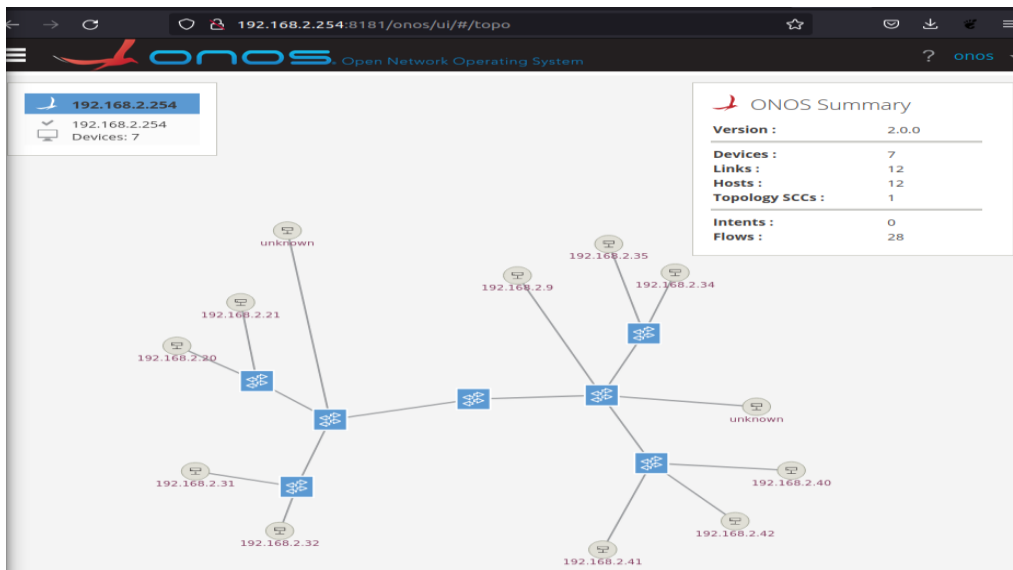
No	Perangkat	Kegunaan
12	Docker	Docker digunakan untuk membuat <i>image</i> ONOS dan iperf supaya dapat digunakan di GNS3 tanpa install dan konfigurasi aplikasi tersebut dari awal

3. Hasil dan Pembahasan

3.1. Hasil Penerapan SDN pada MikroTik

	▲ Interface	Switch	Port Num...	Tx Bytes	Tx Packets	Tx Drops	Rx Bytes	Rx Packets	Rx Drops
- D	ether2	oflow1	1	51.3 KiB	386 B	0 B	202.7 KiB	1179 B	0 B
- D	ether3	oflow1	2	182.0 KiB	860 B	0 B	0 B	0 B	0 B
- D	ether4	oflow1	3	182.9 KiB	870 B	0 B	2030 B	25 B	0 B
- D	ether5	oflow1	4	182.0 KiB	860 B	0 B	0 B	0 B	0 B
- D	ether6	oflow1	5	182.0 KiB	860 B	0 B	0 B	0 B	0 B

Gambar 4. Port OpenFlow di MikroTik



Gambar 5. Tampilan ONOS

Gambar diatas menunjukkan bagaimana SDN sudah dapat diterapkan pada router MikroTik. Gambar 3.1 menunjukkan port OpenFlow yang aktif pada router MikroTik. Gambar 5 menunjukkan ONOS yang sudah dapat mengontrol SDN MikroTik.

3.2. Hasil Pembebanan Menggunakan Iperf

```

root@dadang08-iperf-1:/# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 192.168.2.40, port 50838
[ 5] local 192.168.2.20 port 5201 connected to 192.168.2.40 port 50840
[ ID] Interval      Transfer    Bitrate
[ 5]  0.00-1.00  sec    233 KBytes  1.91 Mbits/sec
    
```

Gambar 6. Tampilan Muka Iperf Server

```

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
dadang08-iperf-2 console is now available... Press RETURN to get started.
root@dadang08-iperf-2:/# iperf3 -c 192.168.2.20 -n 10240K
Connecting to host 192.168.2.20, port 5201
[ 5] local 192.168.2.40 port 50840 connected to 192.168.2.20 port 5201
[ ID] Interval      Transfer    Bitrate      Retr  Cwnd
[ 5]  0.00-1.00  sec    339 KBytes  2.78 Mbits/sec   0   32.5 KBytes
[ 5]  1.00-2.00  sec    206 KBytes  1.69 Mbits/sec   0   30.6 KBytes
    
```

Gambar. 7 Tampilan Muka Iperf Client

3.3. Hasil Pengujian QoS

Pengukuran QoS pada SDN MikroTik dilakukan dengan melakukan pembebanan pada topologi menggunakan Iperf. Iperf adalah pembebanan jaringan dengan konsep server-client. Untuk pembebanan iperf, dibutuhkan 2 host untuk menjadi server dan client. Untuk pengujian pada penelitian ini, jaringan akan dibebani dengan pengiriman data sebesar 200MB dengan bandwidth 50Mbit/s pada protokol tcp/ip dan udp dengan menguji 10 kali pengujian. Pada saat pengujian ada 8 host yang saling terkoneksi. 4 akan menjadi server dan 4 yang lainnya akan menjadi *client*. Setiap *server-client* iperf akan memberi beban sebesar 200MB sehingga secara bersamaan jaringan akan melakukan transfer data sebanyak 800MB. Pada sisi server ketikkan perintah **iperf3 -s** untuk mengaktifkan iperf *server*. Dan pada sisi client ketikkan **perintah iperf3 -c <ip-server> -b 50M -t 200** untuk mengaktifkan iperf client dengan bandwidth 50Mbit/ dan mengirim selama 200 detik. Untuk client yang akan diambil datanya dengan wireshark ketikkan perintah **iperf3 -c <ip-server> -b 50M -n 200M** agar dapat mengirim data sebesar 200MB pada bandwidth 50Mbit/s. Perintah tersebut untuk menguji protocol TCP. Pada protocol UDP perintah yang digunakan adalah **iperf3 -c <ip-server> -b 50M -n 200M -u**. Berikut merupakan rumus dan tabel kategori degradasi yang dipakai untuk mencari ke 4 parameter dari QoS:

1. Delay

$$\text{Rata-rata delay} = \frac{\text{Total Delay}}{\text{Total Paket Yang diterima}} \dots\dots\dots (1)$$

Tabel 2. – Kategori Delay

Kategori Delay*	Delay (ms)	Indeks
Sangat bagus	< 150 ms	4
Bagus	150 – 300 ms	3
Sedang	300 – 450 ms	2
Jelek	> 450 ms	1

2. Packet Loss

$$\text{Packet loss} = \frac{\text{paket dikirim} - \text{paket diterima}}{\text{paket dikirim}} \times 100 \dots\dots\dots (2)$$

Tabel 3. – Kategori Delay

Kategori Degradasi*	Packet Loss	Indeks
Sangat bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

3. Jitter

$$\text{Total variasi delay} = (\text{delay}2 - \text{delay}1) + \dots + (\text{delay} n - \text{delay}(n-1)) \dots\dots\dots (3)$$

$$\text{Jitter} = \frac{\text{Total Variasi Delay}}{\text{Total paket diterima} - 1} \dots\dots\dots (4)$$

Tabel 4. – Kategori Jitter

Kategori Degradasi*	Jitter	Indeks
Sangat bagus	0 ms	4
Bagus	1 – 75 ms	3
Sedang	76 – 125 ms	2
Jelek	126 – 225 ms	1

4. Throughput

$$\text{Throughput} = \frac{\text{paket diterima}}{\text{Waktu antara paket pertama dan terakhir}} \dots\dots\dots (5)$$

$$\text{Throughput}(\%) = \frac{\text{Throughput}(\text{Mbps})}{\text{total bandwidth}(\text{Mbps})} \times 100\% \dots\dots\dots (6)$$

Tabel 5. – Kategori Jitter

Kategori Throughput*	Throughput	Indeks
Sangat bagus	100%	4
Bagus	75 %	3
Sedang	50%	2
Jelek	< 25%	1

3.4. Hasil Perhitungan QoS SDN MikroTik

3.4.1. Delay

TCP

rata – rata delay :

$$= \frac{0,162 + 0,162 + 0,162 + 0,162 + 0,162 + 0,160 + 0,162 + 0,162 + 0,161 + 0,161}{10}$$

$$= 0,161 \text{ ms}$$

UDP

Rata – rata delay :

$$= \frac{0,230 + 0,230 + 0,231 + 0,230 + 0,231 + 0,231 + 0,229 + 0,232 + 0,231 + 0,230}{10}$$

$$= 0,2305 \text{ ms}$$

3.4.2. Throughput

TCP

rata – rata Throughput

$$= \frac{49,162 + 49,590 + 49,655 + 49,211 + 49,162 + 49,151 + 49,181 + 49,459 + 49,253 + 49,423}{10}$$

$$= 49,324 \text{ Mbit/s}$$

$$\frac{49,324 \text{ Mbps}}{50 \text{ Mbps}} \times 100\% = 98,65 \%$$

UDP

rata – rata Throughput =

$$\frac{46,621 + 47,547 + 47,482 + 47,741 + 47,384 + 47,516 + 48,955 + 47,194 + 47,473 + 48,644}{10}$$

$$= 47,656 \text{ Mbit/s}$$

$$\frac{47,656 \text{ Mbps}}{50 \text{ Mbps}} \times 100\% = 95,312 \%$$

3.4.3. Packet Loss

TCP

rata – rata packet loss

$$= \frac{0,024 + 0,025 + 0,028 + 0,003 + 0,005 + 0,025 + 0,012 + 0,006 + 0,017 + 0,010}{10}$$

$$= 0,017 \%$$

UDP

rata – rata packet loss :

$$= \frac{0,004 + 0,005 + 0,001 + 0,007 + 0,006 + 0,010 + 0,009 + 0,005 + 0,004 + 0,011}{10}$$

$$= 0,0055 \%$$

3.4.4. Jitter

TCP

rata – rata jitter

$$= \frac{3,965 + 3,756 + 10,29 + 20,47 + 17,88 + 2,513 + 5,404 + 4,931 + 7,573 + 2,432}{10}$$

$$= 7,921 \times 10^{-6} \text{ ms}$$

UDP

rata – rata jitter

$$= \frac{32,42 + 14,33 + 5,420 + 10,15 + 13,40 + 8,203 + 2,692 + 11,96 + 24,09 + 7,090}{10}$$

$$= 12,975 \times 10^{-6} \text{ ms}$$

3.5. Perbandingan SDN MikroTik, MikroTik Tradisional, dan SDN Mininet

TCP

Tabel 6. Perbandingan TCP

Parameter QoS	SDN MikroTik	MikroTik Tradisional	SDN Mininet
Delay	0,161 ms	0,162 ms	0,163 ms
Packet Loss	0,017 %	0,022 %	0,003 %
Jitter	$7,921 \times 10^{-6} \text{ ms}$	$18,775 \times 10^{-6} \text{ ms}$	$17,013 \times 10^{-6} \text{ ms}$
Throughput	49,324 Mbit/s	48, 816 Mbit/s	48,429 Mbit/s

UDP

Tabel 7. Perbandingan UDP

Parameter QoS	SDN MikroTik	MikroTik Tradisional	SDN Mininet
Delay	0,230 ms	0,231 ms	0,223 ms
Packet Loss	0,0055 %	0,0053 %	0,0002 %
Jitter	$12,975 \times 10^{-6} \text{ ms}$	$8,832 \times 10^{-6} \text{ ms}$	$61,517 \times 10^{-6} \text{ ms}$
Throughput	47,656 Mbit/s	47,341 Mbit/s	49,596 Mbit/s

4. Kesimpulan

Berdasarkan hasil pengujian dan Analisa yang diuraikan pada bab-bab sebelumnya, maka diperoleh kesimpulan sebagai berikut:

1. Pada MikroTik RouterOS dapat diterapkan SDN dengan menggunakan OpenFlow dan kontroler ONOS. Jaringan terkoneksi dengan baik tanpa harus merouting pada setiap router. Hasil pengujian QoS pada SDN MikroTik menunjukkan hasil yang memuaskan.
2. Berdasarkan hasil pengujian didapat nilai delay SDN MikroTik sebesar 0,161 ms untuk TCP dan 0,2305 ms untuk UDP. Untuk nilai *jitter* didapat nilai sebesar $7,921 \times 10^{-6}$ ms untuk TCP dan $12,975 \times 10^{-6}$ ms untuk UDP. Untuk nilai *throughput* didapat nilai sebesar 49,324 Mbit/s (98%) untuk TCP dan 47,656 Mbit/s (95%) untuk UDP. Untuk nilai *packet loss* didapat nilai sebesar 0,0170 % untuk TCP dan 0,0055 % untuk UDP.
3. Berdasarkan hasil nilai pengujian QoS SDN MikroTik, nilai delay TCP dan UDP berada pada indeks 4 menurut parameter PITHON dengan katagori sangat bagus. Untuk nilai *jitter* SDN MikroTik berada pada indeks 4 dengan katagori sangat bagus. Pada nilai *throughput* SDN MikroTik didapat nilai dengan indeks 4 dengan katagori sangat bagus. Demikian dengan nilai *packet loss* SDN MikroTik didapat nilai dengan indeks 4 dengan katagori sangat bagus
4. Hasil QoS SDN MikroTik dibandingkan dengan SDN Mininet dan MikroTik tradisional. Hasil perbandingan didapatkan nilai QoS MikroTik lebih baik daripada MikroTik tradisional dan sama baiknya dengan SDN Mininet.

Ucapan Terima Kasih

Penelitian ini dapat dilaksanakan dengan baik berkat bantuan dari berbagai pihak, untuk itu penulis mengucapkan terimakasih kepada Prodi Teknik Elektro, Fakultas Teknik, Universitas Garut, dan pihak lain yang telah memberikan kerjasama yang baik dalam penelitian ini.

Referensi

- [1] C. B. T. C. Paul Goranson, *Software Defined Networks A Comprehensive Approach*, Cambridge: Todd Green, 2017.
- [2] A. Fadli, "Implemetasi Quality Of Service Pada Campus Network menggunakan Teknologi Software Defined Networking Dan Opendaylight Controller Dengan Metoe Hierarchical Token Bucket," *Skripsi*, p. 5, 2018.
- [3] Tanembum, *Computer Networks*, AS: Prentise Hall, 1996.
- [4] I. Riadi, "Optimalisasi Keamanan Jaringan Menggunakan Pemfilteran Aplikasi berbasis Mikrotik," *JUSI*, vol. 1, p. 74, 2011.
- [5] M. R. Krisnadhi, "Himaster Mipa UNS," UNS, Selasa Maret 2021. [Online]. Available: <https://himaster.mipa.uns.ac.id/blog/index.php/2021/03/16/mengenal-virtualbox-dan-kegunaannya/>. [Accessed Sabtu Oktober 2021].
- [6] M. Fuszner, "Graphical Network Simulator," p. 1, 11 2008.
- [7] R. Rosnelly, "Membandingkan Analisa Trafik Data Pada Jaringan Komputer Antara Wireshark dan Nmap," *Konferensi Nasional Sistem Informasi*, vol. 1, pp. 936-937, 2011.
- [8] R. Wulandari, "Analisis QoS Pada Jaringan Internet (Studi Kasus : UPT Loka Uji Teknik Penambangan Jampang Kulon - LIPI)," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 2, p. 1, 2016.

- [9] T. Pratama, "Perbandingan Metode PCQ, SFQ, RED dan FIFO Pada MikroTik Sebagai Upaya Optimalisasi Layanan Jaringan Pada Fakultas Teknik Universitas Tanjungpura," *Jurnal Teknik Universitas Tanjungpura*, pp. 2-3, 2015.