

TESTING QOS IN SDN NETWORKS USING POX AND RYU

Eka Stephani Sinambela¹, Angel Gracia Simamora², Petrus Tristan Siregar³, Ronaldo Julius Siregar⁴

^{1,2,3,4}Teknologi Komputer, Fakultas Vokasi, Institut Teknologi Del

Email: ¹eka@del.ac.id, ²angelsimamora05@gmail.com, ³petrustristan.siregar1804@gmail.com,

⁴ronaldosiregar100@gmail.com

*Penulis Korespondensi

(Naskah masuk: dd mmm yyyy, diterima untuk diterbitkan: dd mmm yyyy)

Abstrak

Jaringan tradisional menghadapi berbagai keterbatasan dalam mengelola lalu lintas data yang semakin dinamis dan kompleks. Hal ini disebabkan oleh arsitektur yang bersifat kaku, proses konfigurasi manual yang tidak efisien, serta keterbatasan dalam adaptasi terhadap perubahan kondisi jaringan. Untuk mengatasi permasalahan tersebut, dikembangkanlah pendekatan Software-Defined Networking (SDN) yang memisahkan antara control plane dan data plane melalui arsitektur terpusat. SDN memberikan fleksibilitas, skalabilitas, dan kemudahan pengelolaan jaringan secara dinamis. Namun demikian, pemilihan controller SDN yang tepat, seperti POX dan Ryu, masih menjadi tantangan karena masing-masing memiliki keunggulan dan kelemahan yang berbeda. Penelitian ini bertujuan untuk mengembangkan dokumentasi serta kompilasi berbasis web yang menyajikan panduan implementasi dan fitur simulasi interaktif guna menguji parameter Quality of Service (QoS) pada kedua controller tersebut. Proses simulasi dilakukan menggunakan emulator Mininet pada beberapa skenario dengan tingkat kompleksitas yang berbeda. Parameter yang diuji mencakup throughput, jitter, packet loss, dan packet delivery ratio. Hasil pengujian menunjukkan bahwa controller Ryu memiliki performa lebih baik dalam menangani jaringan berskala besar karena efisiensi manajemen lalu lintas dan kemampuan adaptasi terhadap perubahan topologi. Ryu menghasilkan throughput yang lebih tinggi, jitter yang lebih rendah, dan kestabilan pengiriman paket yang lebih konsisten. Sementara itu, controller POX menawarkan konfigurasi yang lebih sederhana dan ringan, tetapi kurang optimal dalam skenario jaringan yang kompleks. Dengan demikian, Ryu direkomendasikan untuk implementasi jaringan berskala besar dan dinamis, sedangkan POX lebih sesuai untuk penggunaan sederhana dan kebutuhan pembelajaran dasar.

Kata kunci: *Software-Defined Networking, POX Controller, Ryu Controller, Quality of Service, Mininet*

TESTING QOS IN SDN NETWORKS USING POX AND RYU

Abstract

Traditional networks face various limitations in managing increasingly dynamic and complex data traffic. These challenges stem from rigid architectures, inefficient manual configuration processes, and limited adaptability to changing network conditions. To address these issues, the Software-Defined Networking (SDN) paradigm was introduced, separating the control plane from the data plane through a centralized architecture. SDN offers flexibility, scalability, and simplified dynamic network management. However, selecting the appropriate SDN

controller, such as POX or Ryu, remains a challenge due to their distinct strengths and weaknesses. This study aims to develop web-based documentation and a comprehensive compilation that provides implementation guidelines and interactive simulation features to evaluate Quality of Service (QoS) parameters using both controllers. The simulation process is conducted using the Mininet emulator across several scenarios with varying levels of complexity. The evaluated parameters include throughput, jitter, packet loss, and packet delivery ratio. The results indicate that the Ryu controller demonstrates better performance in handling large-scale networks due to its efficient traffic management and adaptability to topology changes. Ryu achieves higher throughput, lower jitter, and more consistent packet delivery stability. In contrast, the POX controller offers simpler and lightweight configuration, but proves less optimal in complex network scenarios. Therefore, Ryu is recommended for the implementation of large-scale and dynamic networks, while POX is more suitable for simpler use cases and basic educational purposes.

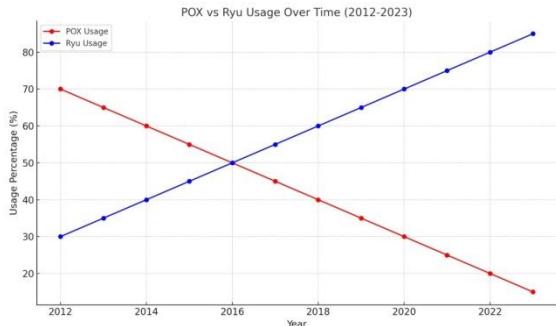
Keywords: Software-Defined Networking, POX Controller, Ryu Controller, Quality of Service, Mininet

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang begitu pesat dalam satu dekade terakhir telah mendorong transformasi signifikan dalam arsitektur jaringan komputer modern. Peningkatan tajam volume lalu lintas data yang dipicu oleh maraknya penggunaan layanan berbasis komputasi awan (cloud computing), Internet of Things (IoT), media sosial, serta aplikasi real-time seperti konferensi video dan layanan streaming telah menuntut keberadaan infrastruktur jaringan yang memiliki fleksibilitas tinggi, efisiensi optimal, serta kemampuan skalabilitas yang adaptif terhadap dinamika kebutuhan. Namun demikian, arsitektur jaringan konvensional yang masih mengintegrasikan fungsi pengendalian (control plane) dan fungsi pengalihan data (data plane) dalam satu perangkat fisik, seperti switch dan router, kerap kali tidak mampu memenuhi ekspektasi tersebut karena keterbatasan pada sisi pengelolaan yang bersifat manual, tidak terpusat, serta tidak dinamis [1].

Sebagai respons terhadap keterbatasan tersebut, muncul paradigma baru yang dikenal dengan Software Defined Networking (SDN). Paradigma ini

memisahkan antara control plane dan data plane dengan memindahkan fungsi kontrol ke sebuah entitas logis terpusat yang disebut controller [2]. Controller berperan sebagai otak jaringan, yang bertugas menetapkan kebijakan dan keputusan perutean, sedangkan perangkat jaringan hanya menjalankan fungsi penerusan paket sesuai instruksi dari controller tersebut. Pemisahan ini memberikan kemudahan dalam pengelolaan jaringan secara menyeluruh, meningkatkan kemampuan otomatisasi, serta mendukung penerapan kebijakan secara terprogram dan fleksibel. Protokol komunikasi yang lazim digunakan antara controller dan perangkat forwarding adalah OpenFlow [3], yang memungkinkan controller mengendalikan jalur data secara langsung melalui tabel forwarding pada switch. Dalam konteks ini, controller memiliki peran strategis dalam pengelolaan lalu lintas, peningkatan kualitas layanan Quality of Service (QoS), keamanan jaringan, serta pemantauan kondisi jaringan secara real-time.



Gambar 1. POX vs Ryu (2012-2023)

(Sumber:

<https://dspace.vut.cz/server/api/core/bitstreams/5a5284b5-2910-4b57-b911-dbc992aea013/content>

Berdasarkan grafik, terlihat bahwa pada 2012-2015 POX mendominasi dengan 70% penggunaan, sementara Ryu hanya 30%. Pada 2016, keduanya seimbang diangka 50%. Namun sejak 2017 hingga 2023, penggunaan POX menurun drastis menjadi 15%, sedangkan Ryu meningkat tajam hingga 85%. Hal ini mencerminkan pergeseran kebutuhan menuju solusi jaringan yang lebih canggih dan skalabel.

Dari sekian banyak controller SDN yang tersedia, dua di antaranya yang sering digunakan dalam penelitian akademik dan eksperimen laboratorium adalah POX dan Ryu. POX merupakan controller berbasis bahasa pemrograman Python yang dirancang untuk kemudahan pengembangan dan pembelajaran [4]. Kesederhanaannya menjadikan POX sangat sesuai digunakan pada lingkungan akademik, khususnya dalam simulasi skenario jaringan skala kecil hingga menengah. Sebaliknya, Ryu hadir sebagai controller yang lebih modular dan mendukung lebih banyak versi protokol OpenFlow serta protokol tambahan seperti NetConf dan OF-Config. Arsitektur Ryu memungkinkan pengembangan aplikasi jaringan secara lebih kompleks dan mendalam, serta menjadikannya lebih sesuai untuk skenario jaringan berskala besar atau lingkungan industri yang menuntut fleksibilitas dan skalabilitas tinggi [5].

Berbagai penelitian terdahulu telah dilakukan guna menganalisis performa controller SDN dalam beragam konfigurasi dan skenario jaringan. Musril Harianto, misalnya, menerapkan algoritma Dijkstra dalam routing OSPF guna menentukan jalur terpendek dengan biaya minimum pada graf berbobot [6]. Ridha Mauldina melakukan pengujian terhadap protokol OSPF menggunakan RouteFlow dan controller POX pada platform Mininet, serta menemukan bahwa peningkatan jumlah switch memberikan pengaruh terhadap waktu konvergensi jaringan [7]. Moh Wahyudi Putra juga melakukan analisis terhadap beberapa controller, dan menyimpulkan bahwa Ryu memiliki ketebalan throughput yang lebih unggul [8]. Berdasarkan tinjauan terhadap studi-studi terdahulu tersebut, dapat disimpulkan bahwa belum terdapat kajian yang secara khusus dan komprehensif membandingkan performa POX dan Ryu berdasarkan parameter QoS dalam berbagai bentuk topologi dan tingkat kompleksitas jaringan.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk melakukan evaluasi dan analisis perbandingan performa antara controller POX dan Ryu dalam pengelolaan lalu lintas jaringan berbasis SDN melalui simulasi menggunakan Mininet [9]. Penelitian ini tidak hanya mencakup pengukuran parameter QoS, seperti throughput, delay, jitter, dan packet loss [10], tetapi juga mengembangkan modul pengujian berbasis web yang dapat digunakan untuk menyederhanakan proses eksperimen tanpa memerlukan konfigurasi manual secara menyeluruh. Topologi jaringan yang digunakan dalam simulasi mencerminkan variasi kompleksitas yang umum ditemukan dalam praktik implementasi jaringan nyata. Dengan demikian, hasil dari penelitian ini diharapkan dapat memberikan kontribusi nyata bagi para peneliti dan praktisi dalam memilih controller SDN yang paling sesuai berdasarkan kebutuhan spesifik jaringan, baik untuk kepentingan akademik,

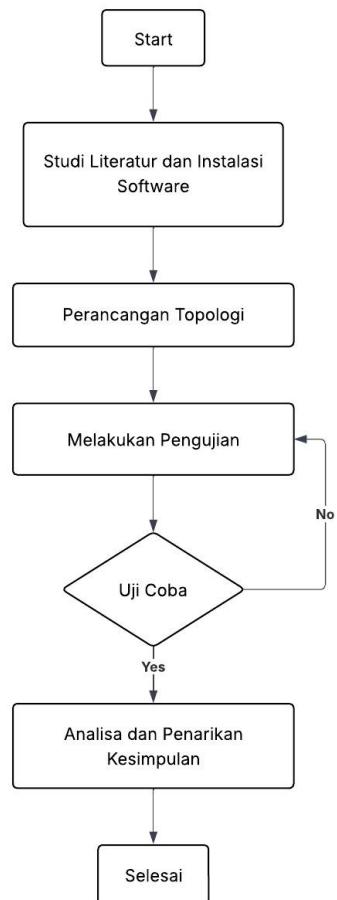
penelitian lanjutan, maupun implementasi di lingkungan operasional profesional.

Lebih jauh lagi, penelitian ini juga bertujuan untuk menyediakan dokumentasi dan kompilasi berbasis web yang komprehensif dan mudah diakses terkait penggunaan controller POX dan Ryu dalam konteks Software Defined Networking (SDN). Dokumentasi ini mencakup penjelasan terperinci mengenai konsep dasar, arsitektur, serta perbedaan sintaksis dan cara kerja antara POX dan Ryu, sehingga pengguna dapat memahami karakteristik masing-masing controller secara lebih sistematis. Selain itu, dokumentasi ini dilengkapi dengan panduan penggunaan yang bersifat praktis melalui tutorial langkah demi langkah serta penerapan pada berbagai skenario pengujian parameter kualitas layanan jaringan, termasuk delay, jitter, packet loss, dan throughput. Modul ini juga disertai fitur interaktif berbasis web yang memungkinkan pengguna untuk menjalankan simulasi dan pengujian secara langsung, sehingga dapat meningkatkan efisiensi, akurasi, dan kemudahan dalam proses pembelajaran maupun pengembangan sistem berbasis SDN secara keseluruhan.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental yang bertujuan untuk membandingkan performa dua controller Software Defined Networking (SDN), yaitu POX dan Ryu, dalam mengelola lalu lintas jaringan berdasarkan parameter Quality of Service (QoS). Pengujian dilakukan menggunakan emulator jaringan Mininet yang dijalankan pada sistem operasi Ubuntu 20.04.1, dengan masing-masing controller dikonfigurasi pada virtual machine (VM) terpisah. Modul yang digunakan untuk masing-masing controller adalah ./pox.py openflow.of_01 untuk POX dan

ryu-manager simple_switch_13.py untuk Ryu.



Gambar 2. Diagram Alur Penelitian

Diagram pada gambar menggambarkan tahapan yang dilaksanakan dalam proyek penelitian. Berikut deskripsi diagram alur yang mencerminkan alur penelitian :

1. Studi Literatur dan Instalasi Software

Pada tahap ini, peneliti akan melakukan pencarian dan penelaahan berbagai sumber referensi seperti jurnal ilmiah, buku, artikel, atau publikasi lain yang relevan dengan topik penelitian. Tujuannya adalah untuk memahami dasar teori, konsep, metode yang sudah ada, hasil penelitian sebelumnya, serta

mengidentifikasi celah penelitian atau masalah yang belum terpecahkan.

Bersamaan atau setelah studi literatur, peneliti akan mempersiapkan alat bantu yang diperlukan, yaitu menginstal perangkat lunak yang akan digunakan dalam penelitian. Ini bisa berupa software simulasi jaringan, software analisis data, atau software lain yang menunjang perancangan dan pengujian.

2. Perancangan Topologi

Setelah memahami teori dan menyiapkan perangkat lunak, tahapan selanjutnya adalah merancang atau mendesain arsitektur jaringan (topologi) yang akan diuji atau diimplementasikan. Perancangan ini meliputi penentuan jenis perangkat (router, switch, server, dll.), tata letak fisik dan logis jaringan, konfigurasi alamat IP, protokol routing, dan parameter-parameter jaringan lainnya sesuai dengan tujuan penelitian.

3. Melakukan Pengujian

Pada tahap ini, rancangan topologi yang telah dibuat akan diimplementasikan dalam lingkungan simulasi (jika menggunakan software simulasi) atau diatur pada perangkat fisik. Kemudian, berbagai skenario pengujian akan dijalankan untuk mengumpulkan data performa jaringan, seperti throughput, delay, jitter, packet loss, dan lain-lain. Pengujian ini dilakukan untuk melihat bagaimana topologi yang dirancang berperilaku di bawah kondisi tertentu.

4. Uji Coba (Decision / Keputusan)

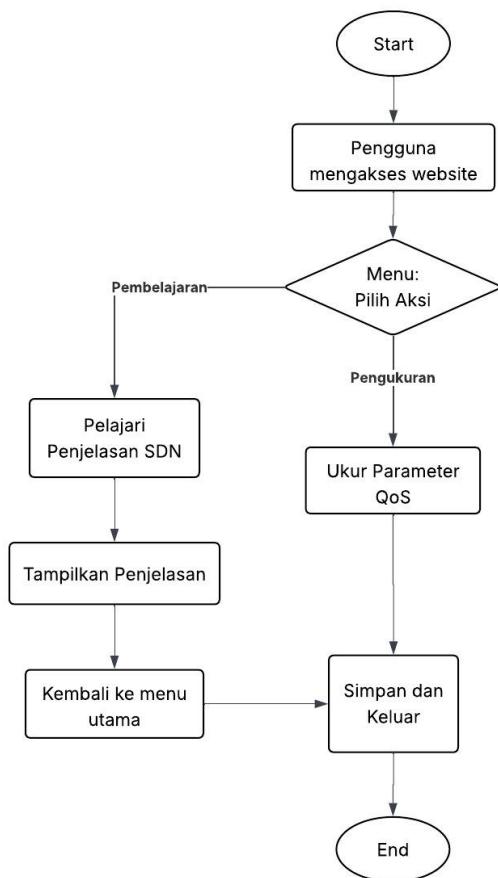
Ini adalah titik keputusan. Setelah melakukan pengujian, data yang terkumpul akan dievaluasi. Jika hasil pengujian sudah dianggap memuaskan, sesuai dengan kriteria yang ditetapkan, atau data yang terkumpul sudah cukup untuk analisis, maka proses akan berlanjut ke tahapan berikutnya.

5. Analisa dan Penarikan Kesimpulan

Setelah semua data pengujian terkumpul dan dianggap valid, pada tahapan ini data tersebut akan dianalisis secara mendalam. Analisis bisa menggunakan metode statistik, perbandingan dengan standar, atau interpretasi kualitatif.

2.1. Flowchart Cara Kerja Sistem

Dalam Flowchart ini akan digambarkan cara kerja *POX and Ryu*.



Gambar 3. Flowchart Kerja Sistem

Untuk memahami bagaimana sistem *Testing Qos In SDN Networks Using Pox and Ryu* bekerja, berikut merupakan penjelasan langkah-langkahnya :

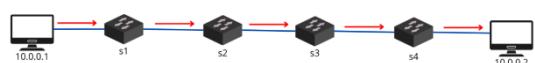
1. Pada tahap ini, interaksi pertama dimulai. Pengguna membuka peramban web dan memasukkan alamat *website* yang dituju untuk memulai penggunaan sistem.
2. Setelah *website* diakses, Jika pengguna ingin melakukan pembelajaran terlebih dahulu, maka dapat dilihat penjelasan-penjelasan pada halaman yang sudah tersedia berupa teks, gambar ataupun perintah.
3. Setelah materi disiapkan, sistem akan menampilkannya kepada pengguna di antarmuka *website*.

4. Setelah pengguna selesai mempelajari penjelasan, sistem akan memberikan opsi untuk mengembalikan pengguna ke menu utama.
5. Jika pengguna ingin melakukan pengukuran, dapat memilih fitur pengukuran performa.
6. Pada tahap ini, sistem akan melakukan proses pengukuran kualitas layanan (QoS) jaringan pada topologi yang tersedia. Parameter yang diukur bisa meliputi *throughput*, *packet loss*, *delay*, *jitter*, atau parameter lain yang relevan. Proses ini biasanya melibatkan pengiriman dan penerimaan paket data untuk mengumpulkan statistik performa.
7. Setelah pengukuran selesai dan data parameter QoS terkumpul, sistem akan menyimpan hasil pengukuran tersebut. Setelah penyimpanan berhasil, pengguna dapat keluar dari sesi pengukuran.

2.2. Penjelasan Topologi

Penelitian ini menggunakan 2 topologi yang akan diujikan, dan topologi memiliki beban pada setiap jalurnya yang telah di konfigurasi pada kontroler POX dan Ryu menggunakan algoritma djikstra untuk pemilihan rutenya. Berikut rancangan topologi jaringan yang digunakan :

1) Topologi 1

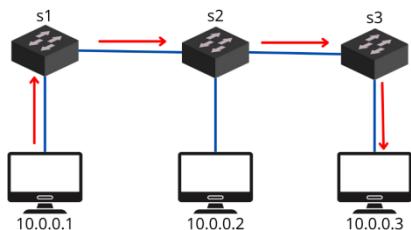


Gambar 4. Topologi 1

Topologi pada gambar merupakan topologi linear yang terdiri dari dua host, masing-masing dengan alamat IP 10.0.0.1 dan 10.0.0.2, yang dihubungkan melalui empat switch berturut-turut, yaitu s1, s2, s3, dan s4. Data dari host pengirim akan melewati setiap switch secara berurutan sebelum mencapai host tujuan. Struktur ini membentuk jalur komunikasi satu arah yang sederhana namun efektif

untuk kebutuhan pengujian dan simulasi dalam jaringan Software-Defined Networking (SDN). Dalam konteks SDN, topologi ini sering digunakan untuk menguji kinerja controller seperti POX dan Ryu, karena aliran data yang terkontrol dengan baik memungkinkan evaluasi parameter Quality of Service (QoS) seperti throughput, jitter, packet loss, dan packet delivery ratio. Selain itu, topologi linear juga mempermudah pemantauan dan analisis lalu lintas data antar node, sehingga cocok untuk penelitian yang berfokus pada efisiensi routing dan pengelolaan trafik jaringan oleh controller SDN.

2) Topologi 2



Gambar 5. Topologi 2

Topologi pada gambar menunjukkan struktur jaringan linear bertingkat yang terdiri dari tiga host, masing-masing dengan alamat IP 10.0.0.1, 10.0.0.2, dan 10.0.0.3, yang terhubung ke tiga switch berbeda, yaitu s1, s2, dan s3. Setiap host terhubung langsung ke switch terdekat, sementara switch-switch tersebut saling terhubung secara berurutan dari s1 ke s2 lalu ke s3. Aliran data dalam jaringan ini mengikuti jalur dari satu switch ke switch lainnya sebelum mencapai host tujuan. Topologi ini umum digunakan dalam simulasi Software-Defined Networking (SDN) karena strukturnya yang sederhana namun mendukung skenario komunikasi multi-host dan multi-hop. Dalam konteks ini, controller SDN seperti POX dan Ryu bertugas mengatur jalur komunikasi antar host dengan menanamkan aturan (flow entries) pada masing-masing switch. Topologi ini juga memungkinkan pengujian parameter Quality of Service (QoS) seperti throughput, jitter, packet

loss, dan packet delivery ratio dalam berbagai kombinasi komunikasi antar host. Selain itu, struktur ini mendukung analisis efisiensi routing dan performa controller dalam merespons perubahan lalu lintas atau topologi, menjadikannya sangat cocok untuk penelitian SDN yang berfokus pada pengelolaan trafik secara terpusat.

2.3. Scenario Pengujian

Pengujian performansi jaringan dilakukan pada dua topologi dengan dua metode, yaitu dengan menghitung waktu konvergensi jaringan dan melalui pengujian end-to-end menggunakan perangkat lunak Distributed Internet Traffic Generator (D-ITG). Pengujian dilakukan sebanyak 3 kali dengan mengirim 25 paket UDP, masing-masing berukuran 1200 byte, selama 10 detik.

Kemudian diambil rerata dari hasil yang dapat pada setiap pengujian dan selanjutnya dibandingkan hasil dari pengujian untuk menentukan controller mana yang memiliki performansi terbaik. Pada pengujinya berdasarkan parameter throughput, packet loss, delay, dan jitter. Berikut penjelasan dari parameter yang akan digunakan.

1) Throughput

Throughput adalah nilai kecepatan pengiriman data berdasarkan satuan bit per second (bps) yang dapat dihitung berdasarkan total paket yang sukses diterima pada waktu interval tertentu dan dibagi dengan waktu interval tersebut. Semakin besar nilai Throughput maka semakin bagus maka semakin baik performansi jaringan tersebut. Perhitungan rumus Throughput ditampilkan pada persamaan.

$$\text{Throughput} = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman data}}$$

Tabel 1. Standarisasi Nilai Throughput Versi TIPHON

Kategori <i>Throughput</i>	Besar <i>Throughput</i>
Sangat Bagus	>1200 Kbps
Bagus	700 s/d 1200 Kbps
Sedang	338 s/d 700 Kbps
Buruk	0 s/d 338 Kbps

2) Packet Loss

Packet loss didefinisikan sebagai kegagalan transmisi paket data mencapai tujuannya. Perhitungan rumus Packet Loss ditampilkan pada persamaan. Untuk Standarisasi nilai packet loss dapat dilihat pada Tabel.

$$\text{Packet loss} = \frac{\text{paket dikirim} - \text{paket diterima}}{\text{paket dikirim}}$$

Tabel 2. Standarisasi Nilai Packet Loss Versi TIPHON

Kategori <i>Packet Loss</i>	Besar <i>Packet Loss</i>
Sangat Bagus	0%
Bagus	3%
Sedang	15%
Buruk	25%

3) Delay

Delay atau latency adalah waktu yang dibutuhkan paket untuk mencapai tujuan, karena adanya antrian, atau jeda yang dibutuhkan node asal ke node tujuan. Perhitungan rumus Delay ditampilkan pada persamaan dibawah ini. Berikut standarisasi nilai delay.

$$\text{Delay} = \frac{\text{total delay}}{\text{total paket diterima}}$$

Tabel 3. Standarisasi Nilai Delay Versi TIPHON

Kategori <i>Packet Delay</i>	Besar <i>Packet Delay</i>
Sangat Bagus	< 150 ms
Bagus	150 s/d 300 ms
Sedang	300 s/d 450 ms
Buruk	> 450 ms

4) Jitter

Jitter adalah variasi delay (penundaan) pada transmisi paket data di jaringan. Jitter disebabkan oleh perubahan panjang antrian paket, waktu pengolahan data, dan waktu penghimpunan ulang paket di sisi penerima. Jitter berhubungan erat dengan latency, yaitu total waktu yang dibutuhkan untuk mengirim paket dari satu titik ke titik lain di jaringan.

$$\text{Jitter} = \frac{\text{total variasi delay}}{\text{total paket yang diterima}}$$

Tabel 4. Standarisasi Nilai Jitter Versi TIPHON

Kategori <i>Jitter</i>	Besar <i>Jitter</i>
Sangat Bagus	0 ms
Bagus	0 s/d 75 ms
Sedang	75 s/d 125 ms
Buruk	125 s/d 225 ms

3. HASIL DAN PEMBAHASAN

3.1. Persiapan Perangkat Simulasi Jaringan

Saat seluruh persiapan konfigurasi routing OSPF simulasi telah dilakukan, maka scenario pengujian konfigurasi topologi bisa dijalankan. Skenario pengujian performasi routing OSPPF pada controller POX dan Ryu menggunakan spesifikasi :

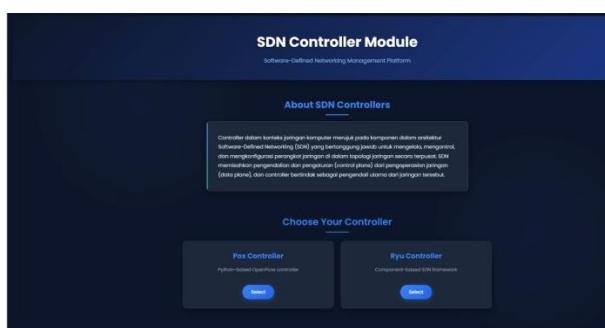
Tabel 5. Spesifikasi Hardware dan Software

Komponen	Spesifikasi / Versi
Processor	Intel i5 10300H
RAM	16 Gb
Storage	512 Gb
Mininet	2.3.1b4
Virtualbox	7.0.18
Ryu	4.34
POX	0.7.0
D-ITG	2.8.1

3.2. Desain Software

3.2.1 Implementasi Desain Page Home

Bagian ini menyajikan implementasi desain pada halaman utama (Home) yang dirancang secara sederhana untuk memperkenalkan konsep dasar SDN kepada pengguna. Halaman ini menampilkan judul dan subjudul yang mencerminkan fungsi platform sebagai sarana pengelolaan SDN. Terdapat penjelasan singkat mengenai peran controller dalam arsitektur SDN, serta dua pilihan utama, yaitu POX dan Ryu, yang disajikan dalam bentuk kartu informasi dengan tombol pemilih. Desain halaman mengutamakan kemudahan akses, tampilan modern, serta kenyamanan visual bagi pengguna.

**Gambar 6. Implementasi Design Page Home**

3.2.2 Implementasi Desain Page Modules

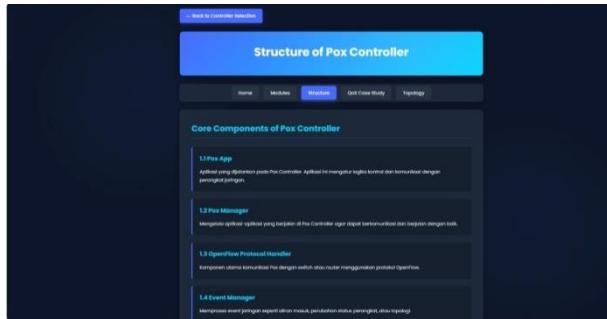
Bagian ini menyajikan implementasi desain pada halaman modules mengenai berbagai modul forwarding pada POX Controller, yang berperan dalam pengelolaan lalu lintas jaringan SDN. Pengguna dapat mengakses navigasi seperti Home, Structure, QoS Case Study, dan Topology untuk eksplorasi lebih lanjut.

Di bagian utama halaman, ditampilkan modul-modul penting seperti layer 2 learning dan hub, lengkap dengan penjelasan singkat dan perintah terminal yang dapat langsung dijalankan. Desain halaman mengutamakan struktur yang rapi dan kontras visual, guna menunjang pembelajaran yang praktis dan mandiri.

**Gambar 7. Implementasi Design Page Modules**

3.2.3 Implementasi Desain Page Structure

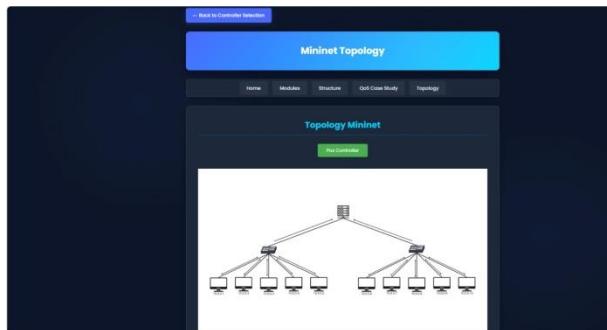
Bagian ini menyajikan implementasi desain pada halaman structure yang menampilkan struktur utama POX Controller, yang terdiri dari Pox App, Pox Manager, dan OpenFlow Protocol Handler. Setiap komponen memiliki peran penting, seperti pengelolaan logika kontrol, integrasi aplikasi, dan komunikasi dengan perangkat jaringan melalui OpenFlow. Desain halaman disusun secara terpisah dan jelas untuk memudahkan pemahaman pengguna.



Gambar 8. Implementasi Design Page Structure

3.2.4 Implementasi Desain Page Topology

Bagian ini menyajikan implementasi desain pada halaman topology yang menampilkan visualisasi topologi dasar yang digunakan dalam simulasi jaringan Mininet sebagai dasar pengujian konfigurasi dan pengukuran QoS dengan POX Controller. Topologi sederhana dipilih agar pengguna dapat mengamati lalu lintas antar host secara jelas. Halaman ini juga menjadi tempat pelaksanaan pengujian QoS, seperti delay, jitter, packet loss, dan throughput, serta membantu pengguna memahami dampak konfigurasi terhadap performa jaringan.



Gambar 9. Implementasi Design Page Topology

4 PENGUJIAN

Pengujian sistem *Testing Qos In SDN Networks Using Pox and Ryu* bertujuan untuk memastikan bahwa sistem dapat berjalan dengan baik, responsive, dan sesuai dengan fungsi dan kebutuhan pengguna.

4.1. POX Controller

Bagian ini menampilkan hasil pengujian Quality of Service (QoS) POX Controller pada Topologi 1 dan Topologi 2, meliputi parameter Throughput, Packet Loss, Jitter, dan Delay pada 3 kali pengujian.

4.1.1 Topologi 1

Pada Topologi 1 didapatkan hasil pengujian seperti gambar dibawah ini:

Pengujian 1

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.980695 s
Total packets	= 1321
Minimum delay	= 0.000096 s
Maximum delay	= 0.004083 s
Average delay	= 0.000351 s
Average jitter	= 0.000138 s
Delay standard deviation	= 0.000364 s
Bytes received	= 84544000
Average bitrate	= 226910.837909 Kbit/s
Average packet rate	= 443.185230 pkts
Packets dropped	= 10 (0.75 %)
Average lossburst size	= 10.000000 pkt
Error lines	= 0

Gambar 10. Hasil Pengujian 1 Topologi 1 POX Controller di Web

Berdasarkan hasil pengujian, performa jaringan menunjukkan nilai-nilai sebagai berikut: Throughput rata-rata tercatat sebesar 226910.837909 Kbit/s. Tingkat packet loss atau paket yang hilang cukup rendah, yaitu 0.75% (sejumlah 10 paket). Untuk jitter atau variasi keterlambatan paket, nilai rataratanya adalah 0.000138 detik. Sementara itu, delay atau waktu tunda rata-rata yang dialami paket adalah 0.000351 detik, dengan nilai minimum 0.000096 detik dan maksimum 0.004083 detik.

Pengujian 2

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.985148 s
Total packets	= 1373
Minimum delay	= 0.000054 s
Maximum delay	= 0.002935 s
Average delay	= 0.000307 s
Average jitter	= 0.000098 s
Delay standard deviation	= 0.000259 s
Bytes received	= 87872000
Average bitrate	= 235491.171627 Kbit/s
Average packet rate	= 459.943695 pkt/s
Packets dropped	= 10 (0.72 %)
Average loss-burst size	= 10.000000 pkt
Error lines	= 0

Gambar 11. Hasil Pengujian 2 Topologi 1 POX Controller di Web

Berdasarkan hasil pengujian yang disajikan, performa jaringan menunjukkan bahwa throughput rata-ratanya adalah 235491.171627 Kbit/s. Tingkat packet loss atau persentase paket yang hilang tercatat sebesar 0.72%, dengan jumlah 10 paket yang terbuang. Untuk jitter atau variasi waktu tunda, nilai rata-ratanya adalah 0.000098 detik. Sementara itu, delay atau waktu tunda rata-rata yang dialami paket adalah 0.000307 detik, dengan nilai minimum 0.000054 detik dan maksimum 0.002935 detik.

Pengujian 3

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.965884 s
Total packets	= 1214
Minimum delay	= 0.000099 s
Maximum delay	= 0.007862 s
Average delay	= 0.000381 s
Average jitter	= 0.000158 s
Delay standard deviation	= 0.000394 s
Bytes received	= 77696000
Average bitrate	= 209572.592859 Kbits
Average packet rate	= 409.321470 pkts
Packets dropped	= 11 (0.90 %)
Average lossburst size	= 11.000000 pkt
Error lines	= 0

Gambar 12. Hasil Pengujian 3 Topologi 1 POX Controller di Web

Berdasarkan hasil pengujian, performa jaringan menunjukkan bahwa throughput rata-ratanya adalah 209572.592859 Kbit/s. Tingkat packet loss tercatat sebesar 0.90%, dengan 11 paket yang terbuang. Untuk jitter atau variasi waktu tunda, nilai rata-ratanya adalah 0.000158 detik. Sementara itu, delay atau waktu tunda rata-rata yang dialami paket adalah 0.000381 detik, dengan nilai minimum 0.000099 detik dan maksimum 0.007862 detik.

4.1.2 Topologi 2

Pada Topologi 2 didapatkan hasil pengujian seperti gambar dibawah ini:

Pengujian 1

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.974713 s
Total packets	= 1295
Minimum delay	= 0.000014 s
Maximum delay	= 0.002175 s
Average delay	= 0.000307 s
Average jitter	= 0.000115 s
Delay standard deviation	= 0.000251 s
Bytes received	= 82880000
Average bitrate	= 222892.090766 Kbit/s
Average packet rate	= 435.336115 pkt/s
Packets dropped	= 8 (0.61 %)
Average loss-burst size	= 8.000000 pkt
Error lines	= 0

Gambar 13. Hasil Pengujian 1 Topologi 2 POX Controller di Web

Berdasarkan hasil pengujian yang disajikan, performa jaringan menunjukkan bahwa throughput rata-ratanya adalah 222892.090766 Kbit/s. Tingkat packet loss atau persentase paket yang hilang tercatat sebesar 0.61%, dengan 8 paket yang terbuang. Untuk jitter atau variasi waktu tunda, nilai rata-ratanya adalah 0.000115 detik. Sementara itu, delay atau waktu tunda rata-rata yang dialami paket adalah 0.000307 detik, dengan nilai minimum 0.000014 detik dan maksimum 0.002175 detik.

Pengujian 2

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.987903 s
Total packets	= 1117
Minimum delay	= 0.000073 s
Maximum delay	= 0.050991 s
Average delay	= 0.000309 s
Average jitter	= 0.000196 s
Delay standard deviation	= 0.001532 s
Bytes received	= 71488000
Average bitrate	= 191406.481402 Kbit/s
Average packet rate	= 373.840784 pkt/s
Packets dropped	= 5 (0.45 %)
Average lossburst size	= inf pkt
Error lines	= 0

Gambar 14. Hasil Pengujian 2 Topologi 2 POX Controller di Web

Berdasarkan hasil pengujian yang disajikan, performa jaringan menunjukkan bahwa throughput rata-ratanya adalah 191406.481402 Kbit/s. Tingkat packet loss atau persentase paket yang hilang tercatat sebesar 0.45%, dengan 5 paket yang terbuang. Untuk jitter atau variasi waktu tunda, nilai

rata-ratanya adalah 0.000196 detik. Sementara itu, delay atau waktu tunda rata-rata yang dialami paket adalah 0.000309 detik, dengan nilai minimum 0.000073 detik dan maksimum 0.050991 detik.

Pengujian 3

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.987659 s
Total packets	= 1239
Minimum delay	= 0.000073 s
Maximum delay	= 0.008249 s
Average delay	= 0.000269 s
Average jitter	= 0.000111 s
Delay standard deviation	= 0.000312 s
Bytes received	= 79296000
Average bitrate	= 212329.452591 Kbit/s
Average packet rate	= 414.705962 pkt/s
packets dropped	= 6 (0.48 %)
Average loss-burst size	= 6.000000 pkt
Error lines	= 0

Gambar 15. Hasil Pengujian 3 Topologi 2 POX Controller di Web

Berdasarkan hasil pengujian yang disajikan, performa jaringan menunjukkan bahwa throughput rata-ratanya adalah 212329.452591 Kbit/s. Tingkat packet loss atau persentase paket yang hilang tercatat sebesar 0.48%, dengan 6 paket yang terbuang. Untuk jitter atau variasi waktu tunda, nilai rata-ratanya adalah 0.000111 detik. Sementara itu, delay atau waktu tunda rata-rata yang dialami paket adalah 0.000269 detik, dengan nilai minimum 0.000073 detik dan maksimum 0.008249 detik.

4.2. Ryu Controller

Bagian ini menampilkan hasil pengujian Quality of Service (QoS) Ryu Controller pada Topologi 1 dan Topologi 2, meliputi parameter Throughput, Packet Loss, Jitter, dan Delay pada 3 kali pengujian.

4.2.1 Topologi 1

Pada Topologi 1 didapatkan hasil pengujian seperti gambar dibawah ini:

Pengujian 1

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.998396 s
Total packets	= 1429
Minimum delay	= 0.000100 s
Maximum delay	= 0.008807 s
Average delay	= 0.000329 s
Average jitter	= 0.000121 s
Delay standard deviation	= 0.000415 s
Bytes received	= 91456000
Average bitrate	= 244013.132355 Kbits
Average packet rate	= 476.588149 pkts
packets dropped	= 0 (0.00 %)
Average loss-burst size	= 0 pkt
Error lines	= 0

Gambar 16. Hasil Pengujian 1 Topologi 1 Ryu Controller di Web

Berdasarkan hasil pengujian yang disajikan, performa jaringan menunjukkan bahwa throughput rata-ratanya adalah 244013.132355 Kbit/s. Tingkat packet loss tercatat sebesar 0.00%, yang berarti tidak ada paket yang hilang. Untuk jitter atau variasi waktu tunda, nilai rata-ratanya adalah 0.000121 detik. Sementara itu, delay atau waktu tunda rata-rata yang dialami paket adalah 0.000329 detik, dengan nilai minimum 0.000100 detik dan maksimum 0.008807 detik.

Pengujian 2

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.997307 s
Total packets	= 1184
Minimum delay	= 0.000097 s
Maximum delay	= 0.010983 s
Average delay	= 0.000336 s
Average jitter	= 0.000135 s
Delay standard deviation	= 0.000438 s
Bytes received	= 75776000
Average bitrate	= 202250.887213 Kbits
Average packet rate	= 395.021264 pkts
packets dropped	= 0 (0.00 %)
Average loss-burst size	= 0 pkt
Error lines	= 0

Gambar 17. Hasil Pengujian 2 Topologi 1 Ryu Controller di Web

Berdasarkan hasil pengujian yang disajikan, performa jaringan menunjukkan bahwa throughput rata-ratanya adalah 202250.887213 Kbit/s. Tingkat packet loss tercatat sebesar 0.00%, yang berarti tidak ada paket yang hilang. Untuk jitter atau variasi waktu tunda, nilai rata-ratanya adalah 0.000135 detik. Sementara itu, delay atau waktu tunda rata-rata yang dialami paket adalah 0.000336 detik,

dengan nilai minimum 0.000097 detik dan maksimum 0.010983 detik.

Pengujian 3

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.997138 s
Total packets	= 1419
Minimum delay	= 0.000097 s
Maximum delay	= 0.002984 s
Average delay	= 0.000333 s
Average jitter	= 0.000105 s
Delay standard deviation	= 0.000288 s
Bytes received	= 90816000
Average bitrate	= 242407.256523 Kbits
Average packet rate	= 473.451673 pkts
Packets dropped	= 0 (0.00 %)
Average lossburst size	= 0 pkt
Error lines	= 0

Gambar 18. Hasil Pengujian 3 Topologi 1 Ryu Controller di Web

Berdasarkan hasil pengujian yang disajikan, performa jaringan menunjukkan bahwa throughput rata-ratanya adalah 242407.256523 Kbit/s. Tingkat packet loss tercatat sebesar 0.00%, yang berarti tidak ada paket yang hilang. Untuk jitter atau variasi waktu tunda, nilai rata-ratanya adalah 0.000105 detik. Sementara itu, delay atau waktu tunda rata-rata yang dialami paket adalah 0.000333 detik, dengan nilai minimum 0.000097 detik dan maksimum 0.002984 detik.

4.2.2 Topologi 2

Pada Topologi 2 didapatkan hasil pengujian seperti gambar dibawah ini:

Pengujian 1

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.999606 s
Total packets	= 1231
Minimum delay	= 0.000052 s
Maximum delay	= 0.003256 s
Average delay	= 0.000310 s
Average jitter	= 0.000094 s
Delay standard deviation	= 0.000222 s
Bytes received	= 78784000
Average bitrate	= 210118.262199 Kbits
Average packet rate	= 410.387231 pkts
Packets dropped	= 0 (0.00 %)
Average lossburst size	= 0 pkt
Error lines	= 0

Gambar 19. Hasil Pengujian 1 Topologi 2 Ryu Controller di Web

Dari hasil pengujian yang ditampilkan, dapat disimpulkan bahwa performa jaringan menunjukkan

hasil yang sangat baik. Throughput mencapai 210118.262199 Kbits, menandakan kapasitas transfer data yang tinggi. Aspek packet loss tercatat 0 (0.00 %), yang berarti tidak ada paket data yang hilang selama transmisi, menunjukkan keandalan jaringan yang sangat optimal. Sementara itu, jitter sebesar 0.000094 s dan delay rata-rata 0.000310 s menunjukkan variasi waktu kedatangan paket yang sangat rendah dan waktu tunda yang minimal, berkontribusi pada pengalaman pengguna yang lancar dan responsif. Secara keseluruhan, data ini mengindikasikan koneksi jaringan yang stabil, cepat, dan efisien.

Pengujian 2

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.996613 s
Total packets	= 1197
Minimum delay	= 0.000051 s
Maximum delay	= 0.008911 s
Average delay	= 0.000257 s
Average jitter	= 0.000102 s
Delay standard deviation	= 0.000330 s
Bytes received	= 76608000
Average bitrate	= 204518.901840 Kbits
Average packet rate	= 399.450980 pkts
Packets dropped	= 0 (0.00 %)
Average lossburst size	= 0 pkt
Error lines	= 0

Gambar 20. Hasil Pengujian 2 Topologi 2 Ryu Controller di Web

Berdasarkan hasil pengujian jaringan yang ditampilkan, performa koneksi menunjukkan karakteristik yang sangat baik dan stabil. Throughput tercatat sebesar 204518.901840 Kbits, menandakan kapasitas transfer data yang tinggi. Jaringan juga menunjukkan keandalan optimal dengan packet loss sebesar 0 (0.00 %), yang berarti tidak ada paket data yang hilang selama proses transmisi. Selain itu, jitter (variasi keterlambatan paket) sangat rendah pada 0.000102 s dan delay rata-rata juga minimal, yaitu 0.000257 s. Secara keseluruhan, data ini mengindikasikan koneksi jaringan yang efisien, cepat, dan responsif.

Pengujian 3

***** TOTAL RESULTS *****	
Number of flows	= 1
Total time	= 2.996855 s
Total packets	= 1362
Minimum delay	= 0.000050 s
Maximum delay	= 0.004633 s
Average delay	= 0.000323 s
Average jitter	= 0.000103 s
Delay standard deviation	= 0.000281 s
Bytes received	= 87168000
Average bitrate	= 232691.938716 Kbits
Average packet rate	= 454.476443 pkts
packets dropped	= 0 (0.00 %)
Average lossburst size	= 0 pkt
Error lines	= 0

Gambar 21. Hasil Pengujian 3 Topologi 2 Ryu Controller di Web

Berdasarkan hasil pengujian jaringan yang disajikan, koneksi menunjukkan performa yang sangat baik dan andal. Throughput tercatat sangat tinggi, mencapai 232691.938716 Kbits, mengindikasikan kemampuan transfer data yang besar dan efisien. Jaringan juga menunjukkan keandalan optimal dengan packet loss sebesar 0 (0.00 %), memastikan tidak ada data yang hilang selama transmisi. Selain itu, jitter sangat rendah pada 0.000103 s dan delay rata-rata minimal 0.000323 s, menunjukkan variasi waktu kedatangan paket yang sangat kecil dan waktu tunda yang minimal, sehingga menjamin pengalaman pengguna yang responsif dan lancar.

4.3. Hasil Pengujian

A. Hasil Pengujian Throughput

Pada tabel dibawah ini hasil dari pengujian Throughput pada tiap topologi dengan pengujian sebanyak 3 kali dan dihasilkan rerata untuk controller POX dan Ryu.

Tabel 6. Perbandingan Rerata Throughput Controller POX dan Ryu

No	POX (Kb/s)	Ryu (Kb/s)
Topologi 1		
1	226910.837909	244013.132355
2	235491.171627	202250.887213
3	209572.592859	242407.256523
Topologi 2		

1	222892.090766	210118.262199
2	191406.481402	204518.901840
3	212329.452591	232691.938716
Rerata	216433.77	222666.73

Berdasarkan hasil pengujian *throughput* yang telah dilakukan, *controller* Ryu menunjukkan performa rata-rata yang sedikit lebih unggul dengan **222666.73 Kb/s**, dibandingkan dengan *controller* POX yang mendapatkan rerata **216433.77 Kb/s**. Selisih *throughput* antara kedua *controller* ini adalah **6232.96 Kb/s**, di mana Ryu lebih tinggi. Perbedaan ini kemungkinan besar disebabkan oleh berbagai faktor seperti perbedaan dalam arsitektur internal, algoritma pengelolaan aliran data, *overhead* pemrosesan, serta optimasi implementasi kode pada masing-masing *controller*, yang secara kolektif memengaruhi efisiensi mereka dalam meneruskan data dalam jaringan.

B. Hasil Pengujian Packet Loss

Pada tabel dibawah ini hasil dari pengujian Packet Loss pada tiap topologi dengan pengujian sebanyak 3 kali dan dihasilkan rerata untuk controller POX dan Ryu.

Tabel 7. Perbandingan Rerata Packet Loss Controller POX dan Ryu

No	POX (%)	Ryu (%)
Topologi 1		
1	0.75	0.00
2	0.72	0.00
3	0.90	0.00
Topologi 2		
1	0.61	0.00
2	0.45	0.00
3	0.48	0.00
Rerata	0.65	0.00

Berdasarkan data *packet loss* yang disajikan, terlihat perbedaan performa yang signifikan antara kedua *controller*. Controller Ryu secara konsisten menunjukkan hasil yang sempurna dengan **0.00%** *packet loss* di semua pengujian, baik pada Topologi 1 maupun Topologi 2. Hal ini mengindikasikan keandalan yang sangat tinggi dalam pengiriman paket data oleh Ryu. Sebaliknya, Controller POX mengalami *packet loss* pada setiap pengujian, dengan rerata mencapai sekitar **0.65%**. Perbedaan ini menegaskan bahwa dalam skenario pengujian ini, Ryu lebih unggul dalam menjaga integritas pengiriman data tanpa kehilangan paket, yang mungkin disebabkan oleh manajemen *flow*, *buffering*, atau respons *controller* yang lebih efisien dibandingkan POX.

C. Hasil Pengujian Delay

Pada tabel dibawah ini hasil dari pengujian Delay pada tiap topologi dengan pengujian sebanyak 3 kali dan dihasilkan rerata untuk controller POX dan Ryu.

Tabel 8. Perbandingan Rerata Delay Controller POX dan Ryu

No	POX (ms)	Ryu (ms)
A		
1	0.000351	0.000329
2	0.000307	0.000336
3	0.000381	0.000333
B		
1	0.000307	0.000094
2	0.000309	0.000102
3	0.000269	0.000103
Rerata	0.000321	0.000216

Berdasarkan data *delay* yang disajikan, Controller Ryu secara rata-rata menunjukkan performa yang lebih baik dengan nilai *delay* yang lebih rendah, yaitu sekitar **0.000216 ms**, dibandingkan dengan Controller POX yang memiliki rerata *delay* sekitar

0.000321 ms. Selisih rata-rata *delay* antara POX dan Ryu adalah sekitar **0.000105 ms**, di mana Ryu lebih unggul. Perbedaan ini terutama sangat jelas terlihat pada pengujian di Topologi 2, di mana *delay* Ryu menurun drastis dibandingkan dengan POX. Hal ini mengindikasikan bahwa Ryu memiliki kemampuan yang lebih efisien dalam memproses dan meneruskan paket data, menghasilkan latensi yang lebih rendah dan respons jaringan yang lebih cepat dibandingkan dengan POX dalam kondisi pengujian ini.

D. Hasil Pengujian Jitter

Pada tabel dibawah ini hasil dari pengujian Jitter pada tiap topologi dengan pengujian sebanyak 3 kali dan dihasilkan rerata untuk controller POX dan Ryu.

Tabel 9. Perbandingan Rerata Jitter Controller POX dan Ryu

No	POX (ms)	Ryu (ms)
A		
1	0.000138	0.000121
2	0.00098	0.000135
3	0.000158	0.000105
B		
1	0.000115	0.000310
2	0.000196	0.000257
3	0.000111	0.000323
Rerata	0.000136	0.0002085

Berdasarkan data *jitter* yang ditampilkan, terlihat bahwa Controller POX secara rata-rata menunjukkan nilai *jitter* yang lebih rendah, yaitu sekitar **0.000136 ms**. Sementara itu, Controller Ryu memiliki rata-rata *jitter* yang lebih tinggi, sekitar **0.0002085 ms**. Dengan demikian, selisih rata-rata *jitter* antara Ryu dan POX adalah sekitar **0.0000725 ms**, di mana POX menunjukkan performa yang lebih baik dalam stabilitas waktu kedatangan paket. Pola ini sangat kentara pada Topologi 2, di mana *jitter* Ryu meningkat secara signifikan dibandingkan dengan

POX yang cenderung lebih stabil. Hal ini mengindikasikan bahwa POX mungkin lebih efektif dalam mengurangi variasi keterlambatan paket, terutama dalam konfigurasi jaringan tertentu, dibandingkan dengan Ryu.

Setelah dilakukan pengujian QoS pada ke 2 topologi pada pengujian parameter Throughput, Packet Loss, Delay, dan Jitter didapatkan hasil untuk perbandingan kedua controller dengan dilakukan sistem point, jika controller yang lebih baik diberi point 2 dan lainnya diberi point 1. Pada hasil performansi di dapatkan point pada tabel di bawah ini.

Tabel 10. Tabel Point Controller POX dan Ryu

Parameter	POX	Ryu
Throughput	216433.77 Kb/s	222666.73 Kb/s
Packet Loss	0.65 %	0.00 %
Delay	0.000321 ms	0.000216 ms
Jitter	0.000136 ms	0.000208 ms
Total Point	5	7

Pada semua pengujian rerata dibandingkan untuk performansi dari controller POX dan controller Ryu pada 4 parameter.

Pada pengujian Throughput semakin tinggi nilai Throughput semakin baik performansi jaringan tersebut. Pada pengujian Delay semakin rendah waktu yang di dapat semakin baik performansi jaringan tersebut. Pada pengujian Jitter semakin rendah waktu yang di dapat semakin baik performansi jaringan tersebut. Pada paket loss semakin rendah persentase packet loss yang di dapat semakin baik performansi jaringan tersebut.

5 KESIMPULAN

Penelitian ini telah mencapai tujuan utamanya dengan menghasilkan sebuah modul inovatif yang dirancang untuk mempermudah pengujian kualitas layanan (QoS) pada jaringan SDN untuk pengguna. Modul ini berhasil menghilangkan kompleksitas konfigurasi manual, memberikan solusi yang efisien dan akurat untuk para pengguna, terutama administrator jaringan. Dengan desain yang intuitif dan modern, modul ini tidak hanya meningkatkan produktivitas tetapi juga mempermudah aksesibilitas teknologi SDN ke berbagai lapisan pengguna. Implementasi modul ini menunjukkan bahwa pendekatan otomatisasi dan pengelolaan terpusat mampu memberikan dampak positif yang signifikan pada efisiensi operasional jaringan.

Dan berdasarkan hasil dari pengujian yang telah dilakukan, Ryu terbukti lebih responsif terhadap perubahan topologi jaringan dan lebih efisien dalam manajemen lalu lintas, menjadikannya pilihan yang lebih sesuai untuk implementasi jaringan berskala besar yang membutuhkan performa tinggi dan fleksibilitas. Sebaliknya, POX, meskipun menawarkan konfigurasi yang lebih sederhana dan beban pemrosesan yang rendah, menunjukkan keterbatasan dalam skenario jaringan yang lebih kompleks dan cenderung mengalami *packet loss*. Oleh karena itu, untuk kebutuhan jaringan dengan performa optimal dan keandalan tinggi, Ryu merupakan pilihan yang lebih unggul.

DAFTAR PUSTAKA

- [1] F. F. Shodiq and A. Prihanto, "Analisis Perbandingan Performansi Kontroler RYU Dan POX Berbasis Software Defined Network (SDN) Pada Routing OSPF," *J. Informatics Comput. Sci.*, vol. 3, no. 03, pp. 216–223, 2021, doi: 10.26740/jinacs.v3n03.p216-223.

- [2] M. H. Albowarab, N. A. Zakaria, and Z. Z. Abidin, “Software Defined Network: Architecture and Programming Language Survey,” *Int. J. Pure Appl. Math.*, vol. 119, no. 18, pp. 561–572, 2018.
- [3] C. Fu, W. John, and C. Meirosu, “EPLE: An Efficient Passive Lightweight Estimator for SDN packet loss measurement,” *2016 IEEE Conf. Netw. Funct. Virtualization Softw. Defin. Networks, NFV-SDN 2016*, pp. 192–198, 2017, doi: 10.1109/NFV-SDN.2016.7919497.
- [4] G. M. S. Sadam Fauzi, Sifa Larasati, Adhwia Alifia Putri, Nissa Restyasari, “Simulasi Multi-topologi Jaringan Berbasis SDN dengan Controller POX,” *J. homepage http://ejournal.upi.edu/index.php/TELNECT/*, vol. 1, no. 2, pp. 77–84, 2021, [Online]. Available: <https://ejournal.upi.edu/index.php/TELNECT/article/download/40820/17835>
- [5] N. Naim, M. Imad, M. B. A. , Muhammad Abul Hassan, Shabir, Khan, and and Amir Ullah Khan, “POX and RYU Controller Performance Analysis on Software Defined Network,” vol. 9, no. 3, 2023.
- [6] M. M. Mutoffar, A. A. N. Gunawan, A. A. N. F. C. Negara, N. Gunantara, and H. A. Musril, *Jaringan Komputer: Konsep dan Aplikasi Modern*. 2024.
- [7] I. N. Khoerotunisa, S. N. Hertiana, and R. M. Negara, “Analysis Of User Mobility Performance On Software Defined Wireless Network Using Dijkstra Algorithm,” *J. Tek. Inform.*, vol. 2, pp. 127–133, 2021.
- [8] M. W. Putra, E. S. Pramukantoro, and W. Yahya, “Analisis Perbandingan Performansi Kontroller Floodlight , Maestro , RYU , POX Dan ONOS Dalam Arsitektur Software Defined Network (SDN),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 10, pp. 3779–3787, 2018.
- [9] Y. S. Hariyani, I. D. Irawati, S. Danu Dwi, and M. Nuruzzamanirridha, “Routing implementation based-on software defined network using Ryu controller and openvswitch,” *J. Teknol.*, vol. 78, no. 5, pp. 295–298, 2016, doi: 10.11113/jt.v78.8315.
- [10] M. Karakus and A. Durresi, “Quality of Service (QoS) in Software Defined Networking (SDN): A survey,” *J. Netw. Comput. Appl.*, vol. 80, pp. 200–218, 2017, doi: 10.1016/j.jnca.2016.12.019.