

Analisis Komparatif Kinerja Kontroler OpenDaylight dan POX pada Jaringan SDN Berdasarkan Parameter QoS

Lalu Delsi Samsumar^{*1}, Zaenudin², Supardianto³

^{1,2,3}Universitas Teknologi Mataram, Jl. Pelor Mas III, Kekalik, Sekarbela, Mataram, NTB

^{1,2,3}Fakultas Teknologi Informasi dan Komunikasi, Universitas Teknologi Mataram, Mataram

e-mail: ^{*1}samsumarld@utmmataram@ac.id, ²zen3d.itb@gmail.com,

³supardianto88mkom@gmail.com

Abstrak

Dalam arsitektur Software-Defined Networking (SDN), kontroler berfungsi sebagai pusat kendali jaringan yang performanya sangat krusial. Kinerja kontroler yang buruk dapat menjadi titik hambat (bottleneck) dan menurunkan kualitas layanan jaringan secara keseluruhan, sehingga pemilihan kontroler yang tepat menjadi fundamental dalam implementasi SDN. Penelitian ini menjadi penting karena melakukan analisis komparatif secara langsung antara dua kontroler populer dengan filosofi desain yang berbeda: OpenDaylight (ODL) yang berbasis industri dan POX yang ringan untuk riset. Tujuan utama penelitian ini adalah mengevaluasi dan membandingkan kinerja kedua kontroler tersebut dalam menangani beban trafik data yang tinggi dan bervariasi. Metode penelitian yang digunakan adalah eksperimen kuantitatif dalam lingkungan simulasi menggunakan Mininet. Kinerja kedua kontroler diuji pada topologi yang sama dengan tiga skenario beban trafik data yang berbeda (1,02 Gb, 5 Gb, dan 10 Gb). Parameter yang diukur adalah Quality of Service (QoS), yang meliputi throughput, delay, packet loss, dan jitter. Hasil penelitian menunjukkan bahwa OpenDaylight secara umum lebih unggul, dengan performa throughput (rata-rata 199 Mbps), delay (rata-rata 0,950 ms), dan jitter (rata-rata 0,950 ms) yang lebih baik. Sebaliknya, kontroler POX menunjukkan keunggulan pada parameter packet loss, dengan nilai rata-rata yang lebih rendah (2,67%) dibandingkan OpenDaylight (3,40%). Kesimpulannya, hasil ini mengindikasikan adanya pertukaran kinerja (trade-off), di mana OpenDaylight lebih cocok untuk jaringan yang menuntut kecepatan tinggi, sementara POX lebih andal untuk aplikasi yang memprioritaskan integritas pengiriman data.

Kata kunci: Software-Defined Networking (SDN), OpenDaylight, POX, Mininet, Quality of Service (QoS)

1. PENDAHULUAN

Latar belakang penelitian ini berfokus pada analisis kinerja kontroler OpenDaylight dan POX dalam konteks jaringan Software Defined Networking (SDN) berdasarkan parameter Quality of Service (QoS). SDN telah diakui sebagai sebuah paradigma inovatif yang merevolusi cara pengelolaan dan penyediaan infrastruktur jaringan, mengatasi kompleksitas yang dihadapi oleh sistem jaringan tradisional melalui pemisahan antara control plane dan data plane. Dengan demikian, pendekatan ini memungkinkan kontrol yang lebih terpusat dan fleksibel [1], [2].

Dalam konteks ini, pemilihan kontroler yang tepat menjadi kritis, karena kontroler adalah entitas utama yang mengatur lalu lintas data dan mempengaruhi faktor-faktor QoS seperti

latensi, jitter, throughput, dan kehilangan paket [3], [4]. Sebagaimana diungkapkan oleh Jayawardena et al., POX adalah salah satu kontroler awal yang dirancang untuk kemudahan dan pengembangan prototipe cepat, sementara OpenDaylight menawarkan fitur yang lebih komprehensif untuk pengelolaan jaringan yang lebih besar dan kompleks [5]. Melalui perbandingan performa kedua kontroler ini, diharapkan dapat diperoleh wawasan mengenai kemampuan dan keterbatasan masing-masing dalam memenuhi kebutuhan QoS dalam aplikasi jaringan modern.

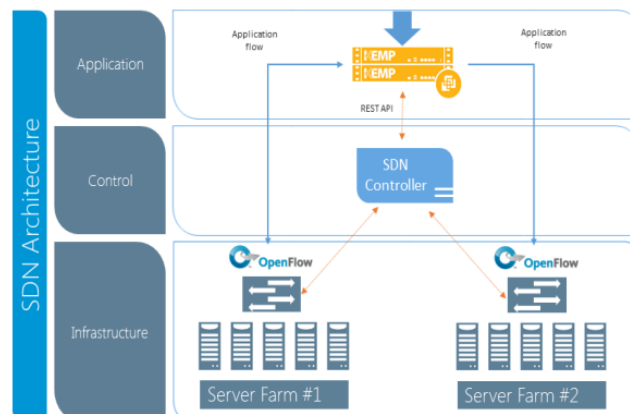
Berdasarkan penelitian sebelumnya, dikemukakan bahwa kontroler POX dan RYU dapat menunjukkan performa yang berbeda dalam berbagai aspek. Dalam analisis performa routing menggunakan OSPF, Iryani et al. menemukan bahwa POX dan RYU menunjukkan perbedaan signifikan dalam hal efisiensi routing, di mana variabel yang digunakan mencakup protokol TCP dan UDP [6]. Penelitian serupa oleh Nuraeni et al. juga menunjukkan bahwa POX membuktikan keunggulan dalam hal latensi dibandingkan dengan kontroler lain seperti FloodLight [7]. Temuan ini menyoroti pentingnya melakukan pemilihan kontroler berdasarkan parameter yang sesuai dengan aplikasi yang diinginkan.

Lebih lanjut, kualitas layanan (QoS) dalam aplikasi SDN sangat dipengaruhi oleh arsitektur dan implementasi kontroler. Kodar et al. menunjukkan bahwa pada pengujian streaming video, POX dan FloodLight memiliki performa yang berbeda berdasarkan pengukuran jitter dan throughput, dua parameter penting dalam konteks QoS [8]. Penelitian ini memperkuat pemahaman bahwa keputusan dalam pemilihan kontroler harus diambil dengan mempertimbangkan konteks spesifik penggunaannya, karena parameter QoS dapat bervariasi tergantung pada karakteristik aplikasi yang dijalankan [9], [10].

Menyusul analisis kinerja kontroler, perhatian juga perlu diberikan kepada mekanisme failover dalam SDN. Dalam penelitian Saputra et al., efek dari mekanisme failover diuji dalam konteks kontroler yang berbeda, mendemonstrasikan pentingnya menjaga kontinuitas layanan meskipun terjadi kegagalan pada kontroler utama [11]. Ini merupakan aspek penting bagi QoS, karena kegagalan dalam sistem pengontrol dapat menyebabkan gangguan layanan yang signifikan dalam jaringan [12]. Dengan demikian, penelitian ini akan mengeksplorasi dan mengevaluasi kemampuan failover dari kontroler OpenDaylight dan POX sebagai bagian dari analisis kinerja keseluruhan.

Dengan memahami interaksi dan performa kontroler dalam konteks QoS, penelitian ini diharapkan dapat memberikan kontribusi signifikan terhadap pengembangan jaringan SDN yang lebih efisien dan handal. Melalui pengukuran sistematis terhadap variabel-variabel yang mempengaruhi QoS, seperti latensi, throughput, dan kehilangan paket, penelitian ini tidak hanya akan memperdalam pemahaman tentang kinerja kontroler yang ada, tetapi juga menetapkan landasan bagi pengembangan solusi teknologi yang lebih baik di masa depan dalam arena jaringan [3], [10]. Dengan adanya pembobotan yang tepat pada setiap parameter QoS, para praktisi dan peneliti dapat membuat keputusan yang lebih terinformasi mengenai arsitektur dan kontroler mana yang paling sesuai untuk aplikasi tertentu dalam lingkungan SDN yang terus berkembang.

Selain itu, penting bagi penelitian ini untuk mengenali dan menjawab tantangan yang dihadapi dalam penerapan SDN, terutama di bidang keamanan dan kepatuhan terhadap QoS yang diharapkan [13], [14].



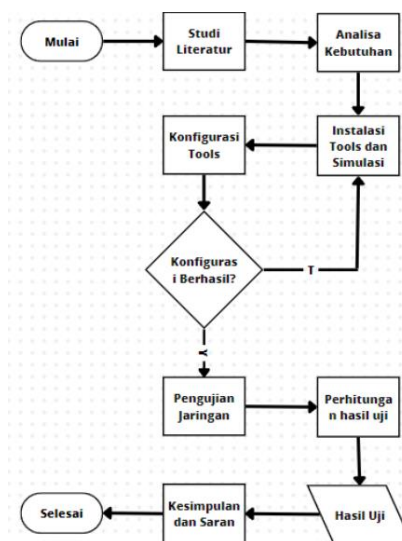
Gambar 1. Arsitektur SDN

Studi ini berupaya untuk mengeksplorasi dan memberi gambaran yang jelas tentang bagaimana kontroler yang berbeda dapat mempengaruhi kualitas layanan, dan menjadikan hasil analisis ini sebagai dasar untuk rekomendasi kebijakan dan praktik terbaik di bidang pengelolaan jaringan [13], [15]. Oleh karena itu, penelitian ini diharapkan tidak hanya berfungsi sebagai analisis teknis semata, tetapi juga sebagai panduan praktis bagi mereka yang bertanggung jawab atas pengelolaan dan perkembangan jaringan yang berbasis pada teknologi SDN.

Melalui pendekatan komprehensif ini, latar belakang penelitian ini berusaha untuk menjelaskan relevansi dan kebutuhan mendesak untuk menggali lebih dalam dan menganalisis kinerja kontroler dalam konteks QoS. Dengan demikian, hasil yang diperoleh diharapkan dapat memperkaya literatur di bidang SDN dan mendukung inisiatif pengembangan teknologi jaringan yang lebih responsif, andal, dan berfokus pada kebutuhan pengguna.

2. METODE PENELITIAN

Penelitian ini menggunakan metode eksperimental kuantitatif yang dilakukan dalam lingkungan simulasi untuk mendapatkan data kinerja dari setiap kontroler. Proses penelitian dirancang secara sistematis untuk memastikan bahwa data yang diperoleh valid dan dapat direplikasi. Tahapan penelitian yang dilakukan diilustrasikan pada Gambar 2 dan diuraikan secara rinci pada sub-bab berikut.



Gambar 2. Tahapan Penelitian

2.1. Kebutuhan Sistem dan Perangkat

Tahap awal penelitian adalah studi literatur dan analisis kebutuhan sistem. Studi literatur dilakukan dengan mengkaji referensi dari jurnal-jurnal penelitian, buku, dan sumber daring untuk menyusun landasan teori yang kokoh terkait SDN, kontroler OpenDaylight, dan POX. Selanjutnya, dilakukan analisis kebutuhan sistem untuk menentukan perangkat keras dan perangkat lunak yang diperlukan untuk mendukung lingkungan simulasi. Perangkat lunak utama yang digunakan dalam penelitian ini adalah:

- a. VirtualBox: Digunakan sebagai platform virtualisasi untuk menjalankan sistem operasi.
- b. Ubuntu Server: Sistem operasi yang menjadi basis untuk instalasi Mininet dan kontroler SDN.
- c. Mininet: Aplikasi emulator jaringan yang digunakan untuk membangun topologi jaringan virtual yang terdiri dari *host*, *switch*, dan kontroler.
- d. Kontroler OpenDaylight dan POX: Dua jenis kontroler SDN yang akan diuji kinerjanya.
- e. Iperf: Alat bantu (*tool*) yang digunakan untuk menghasilkan lalu lintas data dan mengukur parameter QoS.

2.2. Perancangan Topologi dan Skenario Pengujian

Penelitian ini menggunakan topologi jaringan virtual yang dibangun di atas Mininet. Topologi yang dirancang terdiri dari 8 *host* yang terhubung ke 5 *switch* OpenFlow, yang kemudian semuanya diatur oleh satu kontroler terpusat (OpenDaylight atau POX).

Pengujian kinerja dilakukan dengan menjalankan serangkaian skenario yang telah ditentukan. Setiap kontroler diuji sebanyak 3 kali untuk setiap variasi beban trafik data guna memastikan konsistensi hasil. Skenario pengujian dirancang sebagai berikut:

1. Kontroler yang Diuji: OpenDaylight dan POX.
2. Variasi Beban Trafik: Pengiriman paket data antar *host* dilakukan dengan tiga ukuran beban yang berbeda, yaitu 1,02 Gbits, 5 Gbits, dan 10 Gbits.
3. Parameter yang Diukur: Kinerja setiap kontroler diukur berdasarkan empat parameter *Quality of Service* (QoS), yaitu:
 - a. Throughput: Kecepatan transfer data efektif yang berhasil dikirim.
 - b. Delay (Latency): Waktu tunda yang dibutuhkan paket data untuk sampai ke tujuan.
 - c. Packet Loss: Persentase paket data yang hilang selama proses transmisi.
 - d. Jitter: Variasi waktu tunda (delay) antar paket data.

2.3. Prosedur Pengumpulan dan Analisis Data

Proses pengumpulan data dilakukan setelah semua perangkat lunak berhasil diinstalasi dan dikonfigurasi. Pengujian dimulai dengan menjalankan topologi jaringan pada Mininet yang terhubung dengan salah satu kontroler (OpenDaylight atau POX). Selanjutnya, *tool* Iperf digunakan untuk menghasilkan lalu lintas data antar *host* sesuai dengan skenario beban yang telah ditentukan (1,02 Gb, 5 Gb, dan 10 Gb). Selama proses pengiriman data, Iperf secara otomatis mengukur dan mencatat nilai *throughput*, *delay*, *packet loss*, dan *jitter*. Prosedur ini diulangi untuk kedua kontroler pada semua skenario beban.

Setelah data dari semua sesi pengujian terkumpul, tahap selanjutnya adalah perhitungan dan analisis. Data mentah yang diperoleh kemudian dihitung nilai rata-ratanya untuk setiap parameter QoS. Hasil perhitungan tersebut kemudian disajikan dalam bentuk grafik untuk memudahkan proses analisis dan perbandingan visual antara kinerja kontroler OpenDaylight dan POX.

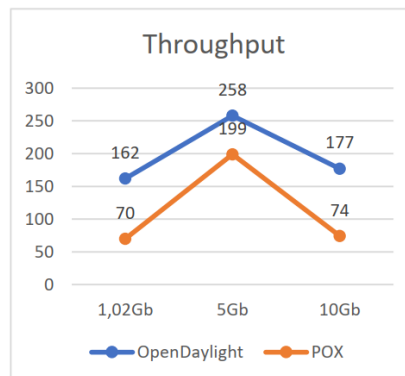
3. HASIL DAN PEMBAHASAN

3.1. Hasil

Setelah melakukan serangkaian pengujian sesuai dengan metode yang telah diuraikan, diperoleh data kinerja untuk kontroler OpenDaylight dan POX. Hasil pengukuran parameter *Quality of Service* (QoS) disajikan secara rinci di bawah ini.

3.1.1. Throughput

Throughput atau laju transfer data efektif dari kedua kontroler pada setiap skenario beban disajikan pada Gambar 3.

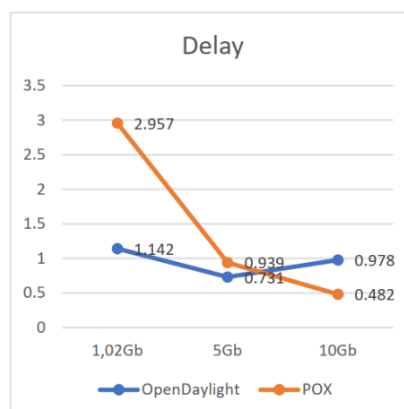


Gambar 3. Grafik Pengujian Throughput

Berdasarkan grafik, terlihat bahwa *throughput* kontroler OpenDaylight secara konsisten lebih tinggi dibandingkan POX di semua skenario beban. Kinerja puncak untuk OpenDaylight tercatat pada beban 5Gb dengan nilai 258 Mbps, sedangkan POX mencapai puncaknya pada beban yang sama dengan nilai 199 Mbps. Menariknya, kedua kontroler mengalami penurunan *throughput* saat diberikan beban tertinggi sebesar 10Gb. Secara keseluruhan, OpenDaylight memiliki rata-rata *throughput* sebesar 199 Mbps, sementara POX memiliki rata-rata 114,33 Mbps.

3.1.2. Delay

Hasil pengukuran *delay* atau waktu tunda pengiriman paket untuk kedua kontroler ditunjukkan pada Gambar 4. Nilai *delay* yang lebih rendah mengindikasikan kinerja yang lebih baik.



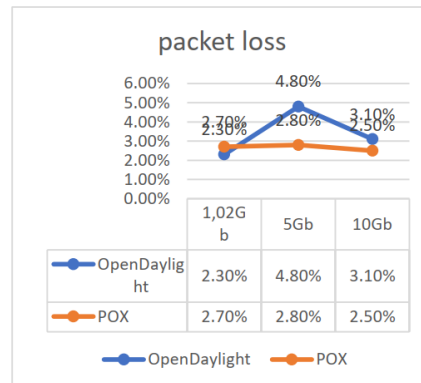
Gambar 4. Grafik Pengujian Delay

Gambar 4 menunjukkan bahwa OpenDaylight memiliki nilai *delay* yang lebih rendah dan lebih stabil dibandingkan POX. Nilai *delay* terendah untuk OpenDaylight adalah 0,731 ms pada

beban 5Gb. Di sisi lain, POX mencatatkan nilai *delay* tertinggi pada beban 1,02Gb sebesar 2,957 ms, yang kemudian menurun drastis seiring penambahan beban. Rata-rata nilai *delay* untuk OpenDaylight adalah 0,950 ms, sedangkan rata-rata untuk POX adalah 1,459 ms.

3.1.3. Packet Loss

Persentase paket yang hilang selama transmisi (*packet loss*) untuk kedua kontroler disajikan pada Gambar 5. Nilai *packet loss* yang lebih rendah menandakan kinerja yang lebih superior.

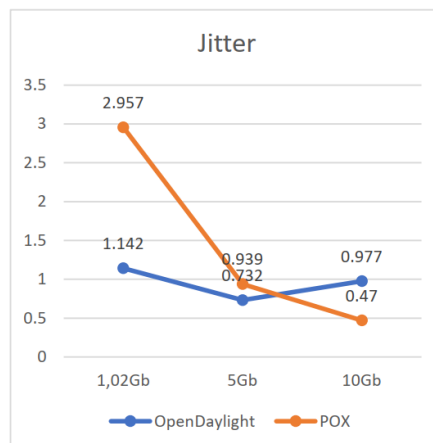


Gambar 5. Grafik Pengujian Packet Loss

Dari hasil pengujian, kontroler POX menunjukkan kinerja *packet loss* yang lebih baik dan stabil. Nilai *packet loss* POX konsisten berada di bawah 3% di semua skenario beban. Sebaliknya, OpenDaylight mengalami lonjakan *packet loss* yang signifikan pada beban 5Gb, mencapai 4,8%. Secara rata-rata, POX memiliki nilai *packet loss* sebesar 2,67%, lebih unggul dibandingkan OpenDaylight dengan rata-rata 3,40%.

3.1.4. Jitter

Jitter atau variasi waktu tunda antar paket disajikan pada Gambar 6. Sama seperti *delay*, nilai *jitter* yang lebih rendah menunjukkan performa yang lebih baik.



Gambar 6. Grafik Pengujian Jitter

Hasil pengujian *jitter* menunjukkan tren yang mirip dengan *delay*. OpenDaylight secara umum memiliki nilai *jitter* yang lebih rendah. Nilai rata-rata *jitter* untuk OpenDaylight adalah 0,950 ms. Sementara itu, POX mencatatkan nilai rata-rata *jitter* yang lebih tinggi, yaitu sebesar 1,455 ms.

3.2. Pembahasan

Pada bagian ini, hasil penelitian yang telah disajikan akan dianalisis lebih dalam untuk memahami faktor-faktor yang memengaruhi perbedaan kinerja antara kontroler OpenDaylight dan POX. Secara keseluruhan, hasil eksperimen menunjukkan bahwa OpenDaylight unggul pada tiga dari empat parameter QoS yang diuji, yaitu *throughput*, *delay*, dan *jitter*. Keunggulan ini dapat diatribusikan pada arsitektur OpenDaylight yang dibangun menggunakan Java dan dirancang sebagai platform yang modular dan *multi-threaded*. Arsitektur ini memungkinkannya untuk menangani banyak tugas secara paralel dengan lebih efisien, sehingga mampu memproses alur (*flow*) data dalam volume besar dengan lebih cepat. Hal ini menjelaskan mengapa nilai *throughput*-nya lebih tinggi serta nilai *delay* dan *jitter*-nya lebih rendah.

Namun, fenomena menarik terjadi pada pengujian *throughput* di beban 10Gb, di mana kedua kontroler mengalami penurunan performa. Ini mengindikasikan adanya titik saturasi, di mana beban trafik yang diberikan telah melebihi kapasitas pemrosesan kontroler. Pada titik ini, kontroler menjadi *bottleneck* dan tidak lagi mampu menangani permintaan alur data baru secara optimal, yang berakibat pada menurunnya laju transfer data efektif.

Di sisi lain, POX menunjukkan keunggulan yang signifikan pada parameter *packet loss*. Kinerjanya yang stabil dengan persentase paket hilang yang lebih rendah kemungkinan disebabkan oleh sifatnya yang lebih ringan dan sederhana. POX, yang berbasis Python, memiliki *overhead* pemrosesan yang lebih kecil dibandingkan OpenDaylight. Arsitekturnya yang tidak sekompleks OpenDaylight mungkin membuatnya lebih efisien dalam mengelola antrian paket pada skenario tertentu, sehingga mengurangi kemungkinan terjadinya tabrakan atau penumpukan paket yang berujung pada *packet loss*. Lonjakan *packet loss* pada OpenDaylight di beban 5Gb mengindikasikan bahwa pada tingkat beban tersebut, kompleksitas fitur internalnya justru menimbulkan tantangan dalam manajemen sumber daya.

Berdasarkan standar yang dirujuk oleh TIPHON, nilai *packet loss* yang baik adalah mendekati 0%. Dalam hal ini, meskipun POX lebih baik, kedua kontroler masih menunjukkan adanya ruang untuk perbaikan. Temuan ini memberikan implikasi praktis bagi para arsitek jaringan: untuk lingkungan yang menuntut *throughput* tinggi dan latensi rendah seperti pusat data, OpenDaylight merupakan pilihan yang lebih superior. Namun, untuk aplikasi di mana integritas data dan kestabilan pengiriman paket menjadi prioritas utama, POX bisa menjadi alternatif yang lebih andal.

4. KESIMPULAN

Berdasarkan analisis hasil pengujian yang telah dilakukan, penelitian ini berhasil menunjukkan adanya perbedaan kinerja yang signifikan antara kontroler OpenDaylight dan POX pada arsitektur jaringan SDN saat dihadapkan pada beban trafik yang bervariasi. Dari perbandingan tersebut, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Kontroler OpenDaylight secara umum menunjukkan kinerja yang lebih unggul pada tiga parameter *Quality of Service* (QoS). Dengan nilai rata-rata *throughput* yang lebih tinggi (199 Mbps), serta nilai rata-rata *delay* (0,950 ms) dan *jitter* (0,950 ms) yang lebih rendah, OpenDaylight terbukti lebih andal untuk lingkungan jaringan yang memprioritaskan kecepatan transfer data tinggi dan latensi minimal.
2. Kontroler POX memiliki keunggulan spesifik pada parameter *packet loss*. Dengan nilai rata-rata *packet loss* yang lebih rendah dan lebih stabil (2,67%) dibandingkan OpenDaylight (3,40%), POX menunjukkan bahwa arsitekturnya yang lebih ringan mampu menjaga integritas pengiriman data dengan lebih baik di bawah tekanan beban trafik yang tinggi.

Dengan demikian, pemilihan antara OpenDaylight dan POX sangat bergantung pada kebutuhan spesifik dari jaringan yang akan diimplementasikan.

5. SARAN

Penelitian ini telah memberikan gambaran perbandingan kinerja antara kedua kontroler. Namun, terdapat beberapa aspek yang dapat dikembangkan lebih lanjut pada penelitian di masa mendatang. Saran untuk penelitian selanjutnya adalah:

1. Pengujian pada Perangkat Fisik: Melakukan pengujian serupa pada lingkungan *testbed* dengan perangkat keras fisik untuk memvalidasi apakah hasil yang diperoleh dalam simulasi konsisten dengan kinerja di dunia nyata.
2. Variasi Topologi Jaringan: Menggunakan topologi jaringan yang lebih kompleks dan beragam, seperti topologi *mesh* atau *fat-tree*, untuk mengetahui bagaimana kedua kontroler menangani skenario perutean yang lebih rumit.
3. Penambahan Kontroler Lain: Menyertakan kontroler SDN populer lainnya dalam analisis perbandingan, seperti Ryu, ONOS, atau Floodlight, untuk memberikan peta perbandingan kinerja yang lebih komprehensif bagi komunitas riset SDN.

DAFTAR PUSTAKA

- [1] K. Nisar *et al.*, "A survey on the architecture, application, and security of software defined networking: Challenges and open issues," *Internet of Things*, vol. 12, p. 100289, 2020.
- [2] N. N. M. Zaki, M. A. Alsaedi, and M. N. Abdullah, "Simulation of 5g mobile core network using sdn & mpls," *Al-Iraqia Journal for Scientific Engineering Research*, vol. 1, no. 1, pp. 88–102, 2022.
- [3] N. Naim, M. Imad, M. A. Hassan, M. B. Afzal, S. Khan, and A. U. Khan, "POX and RYU Controller Performance Analysis on Software Defined Network.," *EAI Endorsed Transactions on Internet of Things*, vol. 9, no. 3, 2023.
- [4] O. Polat and H. Polat, "An intelligent software defined networking controller component to detect and mitigate denial of service attacks," *Journal of Information and Communication Technology*, vol. 20, no. 1, pp. 57–81, 2021.
- [5] C. Jayawardena, J. Chen, A. Bhalla, and L. Bu, "Comparative Analysis of POX and RYU SDN Controllers in Scalable Networks," *arXiv preprint arXiv:2504.12770*, 2025.
- [6] N. Iryani, A. D. Ramadhani, and M. K. Sari, "Analisis Performansi Routing OSPF menggunakan RYU Controller dan POX Controller pada Software Defined Networking," *Incomtech: Jurnal Telekomunikasi Dan Komputer*, vol. 11, no. 1, pp. 73–84, 2021.
- [7] M. safira Nuraeni, E. Wahyudi, and N. Iryani, "Perbandingan Kontroler Pox, Ryu Dan Onos Pada Arsitektur Software Defined Network (Sdn) Menggunakan Topologi Linear,"

- Journal Of Telecommunication Electronics And Control Engineering (JTECE)*, vol. 4, no. 2, pp. 66–79, 2022.
- [8] D. M. Kodar, R. Gunawan, and A. Rahmatulloh, “Performansi Software Defined Network Controller Pada Streaming Video Menggunakan Real-time Transport Protocol,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 2, pp. 381–389, 2021.
 - [9] L. D. Samsumar, B. A. Hidayatulloh, Z. Zaenudin, and P. N. D. Pitaloca, “Analysis of the quality of cloud storage services on Nextcloud and Pydio,” *Journal of Information Technology and Its Utilization*, vol. 6, no. 1, pp. 1–8, 2023.
 - [10] S. J. Rashid, A. M. Alkababji, and A. S. M. Khidhir, “Performance evaluation of software-defined networking controllers in wired and wireless networks,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 21, no. 1, pp. 49–59, 2023.
 - [11] J. W. A. Saputra, M. Faiqurahman, and F. D. S. Sumadi, “Analisis Mekanisme Failover Controller Pada Software Defined Network,” *Jurnal Repositor*, vol. 3, no. 5, pp. 513–522, 2021.
 - [12] A. P. Juledi, M. K. Huda, M. T. D. Putra, L. D. Samsumar, P. D. P. Adi, and Y. Nurdiansyah, “Performance Evaluation of LoRa 923 MHz for the Internet of Things,” in *2024 IEEE 10th Information Technology International Seminar (ITIS)*, IEEE, 2024, pp. 30–34.
 - [13] M. B. Jimenez, D. Fernandez, J. E. Rivadeneira, L. Bellido, and A. Cardenas, “A survey of the main security issues and solutions for the SDN architecture,” *Ieee Access*, vol. 9, pp. 122016–122038, 2021.
 - [14] L. D. Samsumar, B. Imran, M. M. Efendi, R. Muslim, Z. Muahidin, and Z. Mutaqin, “Optimalisasi Keamanan Web Server Ubuntu dengan Teknologi IPS Berbasis Iptables,” 2024.
 - [15] A. Shirmarz and A. Ghaffari, “An autonomic software defined network (SDN) architecture with performance improvement considering,” *Journal of Information Systems and Telecommunication (JIST)*, vol. 8, no. 30, pp. 121–129, 2020.