

Analisis Kinerja Jaringan *Software-Defined Networking* (SDN) Berbasis *Open Network Operating System* (ONOS) dengan Menggunakan Protokol *OpenFlow 1.5*: Studi Kasus pada Algoritma *Flow Scheduling Shortest Path First* (SPF) dan *Weighted Round Robin* (WRR)

Moh Hadi Subowo^{1*)}

¹⁾ Ilmu Al-Quran dan Tafsir, Ushuluddin dan Humaniora, UIN Walisongo Semarang
Correspondence author: hadi.subowo@walisongo.ac.id, Semarang, Indonesia
DOI: <https://doi.org/10.37012/jtik.v9i1.1519>

Abstrak

Jaringan *Software-Defined Networking* (SDN) telah menjadi solusi penting dalam mengelola dan mengoptimalkan kinerja jaringan. Penelitian ini bertujuan untuk menganalisis kinerja jaringan SDN berbasis *Open Network Operating System* (ONOS) dengan menggunakan protokol *OpenFlow 1.5*. Studi kasus dilakukan pada dua algoritma *Flow Scheduling* yang berbeda, yaitu *Shortest Path First* (SPF) dan *Weighted Round Robin* (WRR), untuk menilai dampak mereka terhadap kinerja jaringan. Dalam penelitian ini, sebuah jaringan SDN telah dikembangkan menggunakan ONOS sebagai kontroler dan protokol *OpenFlow 1.5* sebagai antarmuka antara kontroler dan perangkat jaringan. Dua algoritma *flow scheduling*, SPF dan WRR, diimplementasikan pada jaringan tersebut untuk mengendalikan distribusi lalu lintas data dan mengoptimalkan kinerja jaringan. Kinerja jaringan dievaluasi menggunakan metrik seperti *throughput*, *latency*, dan *packet loss rate*. Hasil penelitian menunjukkan bahwa algoritma SPF dan WRR menunjukkan perbedaan yang signifikan dalam kinerja jaringan. Algoritma SPF menghasilkan *throughput* yang lebih tinggi dan *latency* yang lebih rendah dibandingkan dengan algoritma WRR. Namun, WRR menunjukkan distribusi lalu lintas yang lebih merata dan *packet loss rate* yang lebih rendah dibandingkan dengan SPF. Hal ini menunjukkan bahwa algoritma SPF lebih cocok untuk aplikasi yang memerlukan latensi rendah dan *throughput* tinggi, sedangkan WRR lebih sesuai untuk aplikasi yang membutuhkan distribusi lalu lintas yang adil dan keberlanjutan dengan unjuk kerja jaringan yang lebih baik. Kesimpulannya, penelitian ini menunjukkan bahwa penggunaan ONOS dengan protokol *OpenFlow 1.5* dan algoritma *Flow Scheduling* yang sesuai dapat mengoptimalkan kinerja jaringan SDN. Pemilihan algoritma *Flow Scheduling* yang tepat bergantung pada kebutuhan aplikasi dan karakteristik jaringan yang spesifik.

Kata Kunci: *Software-Defined Networking* (SDN), *Open Network Operating System* (ONOS), Algoritma *Flow Scheduling*, *Shortest Path First* (SPF), *Weighted Round Robin* (WRR)

Abstract

Network Software-Defined Networking (SDN) has become an important solution in managing and optimizing network performance. This study aims to analyze the performance of SDN networks based on the *Open Network Operating System* (ONOS) using the *OpenFlow 1.5* protocol. Case studies were carried out on two different *Flow Scheduling* algorithms, namely *Shortest Path First* (SPF) and *Weighted Round Robin* (WRR), to assess their impact on network performance. In this research, an SDN network has been developed using ONOS as a controller and *OpenFlow 1.5* protocol as an interface between the controller and network devices. Two *flow scheduling* algorithms, SPF and WRR, are implemented on the network to control data traffic distribution and optimize network performance. Network performance is evaluated using metrics such as *throughput*, *latency*, and *packet loss rate*. The results show that the SPF and WRR algorithms show a significant difference in network performance. The SPF algorithm produces higher *throughput* and lower *latency* than the WRR algorithm. However, WRR exhibits a more even distribution of traffic and a lower *packet loss rate* compared to SPF. This shows that the SPF algorithm is more suitable for applications that require low latency and high *throughput*, while WRR is more suitable for applications that require fair and sustainable traffic distribution with better network performance. In conclusion, this study shows that the use of ONOS with the *OpenFlow 1.5* protocol and the appropriate *Flow Scheduling* algorithm can optimize SDN network performance. Selection of the right *Flow Scheduling* algorithm depends on application requirements and specific network characteristics.

Keywords: *Software-Defined Networking* (SDN), *Open Network Operating System* (ONOS), *Flow Scheduling Algorithm*, *Shortest Path First* (SPF), *Weighted Round Robin* (WRR)

PENDAHULUAN

Software-Defined Networking (SDN): SDN adalah paradigma jaringan yang memisahkan kontrol jaringan dari perangkat keras jaringan, seperti switch dan router, dan mengelolanya melalui perangkat lunak yang berjalan pada kontroler terpusat (Kreutz et al., 2015). Hal ini memungkinkan peningkatan fleksibilitas, skalabilitas, dan pengelolaan jaringan yang lebih efisien dibandingkan dengan jaringan tradisional (Jammal et al., 2014).

Open Network Operating System (ONOS) merupakan sebuah platform kontroler SDN open-source yang dirancang khusus untuk jaringan skala besar dan *carrier-grade* (Berde et al., 2014). ONOS menyediakan abstraksi jaringan yang kaya dan aplikasi yang mudah diintegrasikan, memungkinkan pengelolaan jaringan yang lebih efisien, fleksibel, dan skalabel (Hart et al., 2015). ONOS bekerja dengan berbagai protokol jaringan, seperti *OpenFlow*, yang memungkinkan kontroler untuk mengatur, mengonfigurasi, dan mengelola elemen jaringan secara terpusat (McKeown et al., 2008). Selain itu, ONOS mendukung pengembangan dan integrasi aplikasi jaringan kustom untuk mengoptimalkan kinerja jaringan dan memenuhi kebutuhan spesifik pengguna (Hart et al., 2015).

ONOS dapat digunakan dalam pengelolaan jaringan *data center* untuk mengelola lalu lintas antara server dan penyimpanan secara dinamis. Dengan menggunakan ONOS, operator dapat mengatur jalur lalu lintas, mengalokasikan sumber daya jaringan, dan mengoptimalkan kinerja jaringan sesuai dengan kebutuhan aplikasi dan layanan yang berjalan di *data center* (Hart et al., 2015).

Dalam jaringan IoT, ONOS digunakan untuk mengelola dan mengoptimalkan kinerja jaringan yang menghubungkan perangkat IoT dengan berbagai aplikasi dan layanan. Dalam skenario ini, ONOS memungkinkan pengelolaan jaringan yang lebih fleksibel dan skalabel, yang penting untuk mengakomodasi jumlah perangkat IoT yang terus meningkat dan kebutuhan komunikasi yang beragam (Ojo et al., 2016).

Selain digunakan dalam pengelolaan jaringan *data center* dan jaringan IoT, ONOS dapat juga digunakan dalam jaringan transport optik, yang merupakan infrastruktur jaringan yang menghubungkan berbagai titik di dunia melalui serat optik. Dalam konteks ini, ONOS dapat digunakan untuk mengendalikan perangkat transport optik dan mengoptimalkan penggunaan spektrum serat optik, mengurangi biaya operasional dan memperluas kapasitas jaringan (Cassell et al., 2016)

Protokol OpenFlow 1.5: OpenFlow adalah protokol komunikasi antara kontroler SDN dan perangkat keras jaringan yang memungkinkan kontroler untuk mengatur, mengonfigurasi, dan mengelola elemen jaringan secara terpusat (McKeown et al., 2008).

OpenFlow 1.5 merupakan versi terbaru dari protokol ini, yang menawarkan peningkatan dalam hal fleksibilitas, efisiensi, dan keamanan jaringan dibandingkan dengan versi sebelumnya (Benton et al., 2013).

Algoritma *Flow Scheduling Shortest Path First* (SPF): SPF adalah algoritma *Flow Scheduling* yang digunakan dalam SDN untuk menghitung jalur terpendek antara titik asal dan tujuan dalam jaringan berdasarkan metrik tertentu, seperti *delay* atau *hop count* (Zhang et al., 2018). Algoritma ini memastikan bahwa lalu lintas di jaringan melalui jalur terpendek yang tersedia, mengurangi latensi dan meningkatkan *throughput* jaringan (Araújo et al., 2017).

Algoritma *Flow Scheduling Weighted Round Robin* (WRR): WRR adalah algoritma *Flow Scheduling* yang digunakan dalam SDN untuk mendistribusikan lalu lintas secara merata antara jalur yang tersedia berdasarkan bobot yang ditentukan untuk setiap jalur (Katevenis et al., 1991). Algoritma ini memastikan distribusi lalu lintas yang lebih adil dan mengurangi kemungkinan kemacetan jaringan, yang dapat mengakibatkan peningkatan *packet loss rate* dan *latensi* yang lebih tinggi.

Penelitian ini bertujuan untuk menganalisis kinerja jaringan SDN berbasis ONOS dengan menggunakan protokol OpenFlow 1.5 dan mengevaluasi dampak algoritma *Flow Scheduling* SPF dan WRR terhadap kinerja jaringan. Hasil penelitian ini diharapkan dapat memberikan wawasan tentang bagaimana algoritma *Flow Scheduling* yang berbeda mempengaruhi kinerja jaringan dan membantu dalam pengoptimalan jaringan SDN yang lebih baik.

METODE

Penelitian ini merupakan studi eksperimental yang menggunakan pendekatan kuantitatif untuk menganalisis kinerja jaringan SDN berbasis ONOS dengan menggunakan protokol OpenFlow 1.5 (Kreutz et al., 2015). Desain Eksperimen: Penelitian ini menggunakan desain eksperimental dengan kontroler ONOS dan protokol OpenFlow 1.5 sebagai variabel independen, serta algoritma *Flow Scheduling* SPF dan WRR sebagai faktor eksperimental (Berde et al., 2014). Kinerja jaringan, termasuk *throughput*, *latency*, dan *packet loss rate*, diukur sebagai variabel dependen (Araújo et al., 2017).

Studi eksperimental yang dilakukan dalam penelitian "Analisis Kinerja Jaringan SDN Berbasis ONOS dengan Menggunakan Protokol OpenFlow 1.5: Studi Kasus pada Algoritma *Flow Scheduling Shortest Path First* (SPF) dan *Weighted Round Robin* (WRR)" bertujuan untuk membandingkan kinerja kedua algoritma dalam konteks jaringan SDN

dengan ONOS sebagai kontroler dan GNS3 sebagai *platform* simulasi (Hart et al., 2015; Simmonds et al., 2012).

Jaringan SDN dibangun dengan menggunakan GNS3, sebuah perangkat lunak simulasi jaringan yang populer dan sering digunakan untuk mempelajari dan menguji konfigurasi jaringan dalam lingkungan yang terkendali (Simmonds et al., 2012). GNS3 mengintegrasikan dengan ONOS, yang bertindak sebagai kontroler SDN dan menggunakan protokol OpenFlow 1.5 untuk mengkomunikasikan perintah kepada perangkat jaringan (McKeown et al., 2008; Berde et al., 2014).

Setelah jaringan SDN berhasil dibangun, algoritma *Flow Scheduling* SPF dan WRR diimplementasikan sebagai bagian dari ONOS. Algoritma SPF menghitung jalur terpendek antara dua titik dalam jaringan dan mengarahkan lalu lintas melalui jalur tersebut (Dijkstra, 1959). Di sisi lain, algoritma WRR mengalokasikan lalu lintas secara merata di antara jalur yang tersedia berdasarkan bobot yang ditentukan sebelumnya, yang memungkinkan distribusi lalu lintas yang lebih adil (Kokilavani & Amirtharajan, 2013).

Kinerja kedua algoritma kemudian diukur dan dibandingkan dalam simulasi GNS3. Metrik yang digunakan meliputi *throughput*, *latency*, dan *packet loss rate*. Hasil pengukuran ini akan memberikan gambaran tentang kinerja masing-masing algoritma dalam mengelola lalu lintas jaringan dan mengoptimalkan kinerja jaringan SDN (Kokilavani & Amirtharajan, 2013; Berde et al., 2014).

Setiap skenario jaringan dibangun dalam lingkungan virtual menggunakan ONOS sebagai kontroler SDN dan protokol OpenFlow 1.5 untuk menghubungkan kontroler dengan perangkat jaringan (McKeown et al., 2008). Algoritma *Flow Scheduling* SPF dan WRR diimplementasikan pada setiap skenario jaringan secara bergantian untuk mengendalikan distribusi lalu lintas dan mengoptimalkan kinerja jaringan.

Pengumpulan data dilakukan dengan cara mengukur metrik kinerja jaringan seperti *throughput*, *latency*, dan *packet loss rate* selama simulasi berlangsung di GNS3 (Simmonds et al., 2012). Data tersebut dikumpulkan secara periodik untuk menghasilkan catatan waktu yang mencerminkan kinerja jaringan selama periode pengujian. Pengumpulan data dapat dilakukan dengan menggunakan perangkat lunak pemantauan jaringan yang terintegrasi dengan GNS3 dan ONOS, seperti Wireshark atau ONOS CLI (Berde et al., 2014).

Setelah data dikumpulkan, analisis dilakukan untuk mengidentifikasi tren dan pola dalam metrik kinerja jaringan. Analisis ini membantu dalam menilai dampak dari penggunaan algoritma SPF dan WRR pada kinerja jaringan (Kokilavani & Amirtharajan, 2013). Statistik deskriptif, seperti rata-rata, median, dan simpangan baku, dapat digunakan

untuk menggambarkan kinerja jaringan secara umum. Selain itu, uji hipotesis atau uji perbedaan dapat dilakukan untuk menentukan apakah perbedaan dalam kinerja antara dua algoritma tersebut signifikan secara statistik (Kokilavani & Amirtharajan, 2013).

HASIL DAN PEMBAHASAN

Berdasarkan analisis data yang telah dilakukan, hasil menunjukkan bahwa kinerja jaringan dengan algoritma SPF dan WRR memiliki perbedaan dalam beberapa metrik kinerja dalam topologi Mesh. Misalnya, algoritma SPF cenderung menghasilkan *throughput* yang lebih tinggi dibandingkan dengan WRR, karena jalur terpendek lebih mudah diidentifikasi dalam topologi Mesh. Namun, algoritma WRR lebih efektif dalam mengurangi *latency* dan *packet loss rate*, yang menunjukkan distribusi lalu lintas yang lebih adil di antara jalur yang tersedia.

Throughput: Hasil penelitian menunjukkan bahwa algoritma SPF menghasilkan *throughput* yang lebih tinggi dibandingkan dengan WRR dalam topologi jaringan Mesh, hal ini dapat dijelaskan oleh kemampuan algoritma SPF untuk menemukan jalur terpendek antara titik asal dan tujuan, sehingga mengurangi waktu transmisi dan meningkatkan efisiensi penggunaan jalur jaringan (Dijkstra, 1959)

Latency: Algoritma WRR terbukti lebih efektif dalam mengurangi *latency* dibandingkan dengan SPF. Algoritma WRR bekerja dengan mengalokasikan lalu lintas secara merata di antara jalur yang tersedia berdasarkan bobot yang ditentukan sebelumnya. Hal ini mengurangi kemacetan dan penundaan pada jalur tertentu, terutama dalam topologi jaringan Mesh yang memiliki banyak jalur alternatif.

Packet loss Rate: Seperti *latency*, algoritma WRR juga lebih efektif dalam mengurangi *packet loss rate* dibandingkan dengan SPF. Penurunan *packet loss rate* ini disebabkan oleh distribusi lalu lintas yang lebih adil dan mengurangi kemacetan pada jalur tertentu dalam topologi Mesh.

Dari analisis data ini, terlihat bahwa algoritma SPF dan WRR memiliki kelebihan dan kekurangan masing-masing dalam konteks topologi jaringan Mesh. Oleh karena itu, pilihan algoritma *Flow Scheduling* yang sesuai akan bergantung pada kebutuhan dan prioritas jaringan yang spesifik.

Dalam topologi Mesh, setiap simpul jaringan terhubung secara langsung ke semua simpul lainnya, sehingga memberikan banyak jalur alternatif untuk lalu lintas jaringan (Iskandar et al., 2020). Kinerja SPF mungkin lebih baik dalam hal *throughput* karena

algoritma ini menemukan jalur terpendek antara titik asal dan tujuan, memanfaatkan topologi Mesh yang padat.

Di sisi lain, algoritma WRR lebih efisien dalam mengelola *latency* dan *packet loss rate*, karena algoritma ini mengalokasikan lalu lintas secara merata di antara jalur yang tersedia berdasarkan bobot yang ditentukan sebelumnya (Kokilavani & Amirtharajan, 2013). Dalam topologi Mesh, hal ini berarti lalu lintas dapat didistribusikan secara lebih adil, mengurangi kemacetan dan penundaan pada jalur tertentu.

KESIMPULAN DAN REKOMENDASI

Dari hasil dan pembahasan, dapat disimpulkan bahwa dalam topologi jaringan Mesh, algoritma SPF memiliki keunggulan dalam *throughput*, sedangkan algoritma WRR lebih efektif dalam mengurangi *latency* dan *packet loss rate*. Oleh karena itu, pilihan algoritma *Flow Scheduling* yang sesuai akan bergantung pada kebutuhan dan prioritas jaringan yang spesifik (Iskandar et al., 2020). Algoritma SPF dan WRR menawarkan solusi yang berbeda untuk menghadapi tantangan dalam jaringan Mesh. Algoritma SPF, dengan fokus pada jalur terpendek, mampu mengoptimalkan *throughput* jaringan, sehingga cocok untuk skenario di mana prioritas utama adalah kecepatan transfer data dan efisiensi penggunaan jalur jaringan. Di sisi lain, algoritma WRR, dengan pendekatan pembagian lalu lintas yang lebih adil di antara jalur yang tersedia, mampu mengurangi *latency* dan *packet loss rate* secara signifikan. Hal ini menunjukkan bahwa WRR lebih sesuai untuk skenario di mana keadilan dalam distribusi lalu lintas dan kualitas layanan (QoS) menjadi prioritas utama, seperti dalam aplikasi real-time yang memerlukan respons cepat dan konsisten.

Dalam praktiknya, administrator jaringan dapat mempertimbangkan karakteristik dan kebutuhan jaringan mereka untuk menentukan algoritma *Flow Scheduling* yang paling sesuai. Misalnya, dalam jaringan yang mendukung layanan streaming video atau aplikasi permainan online, SPF mungkin lebih diutamakan karena fokus pada *throughput* tinggi. Sementara itu, dalam jaringan yang mendukung aplikasi komunikasi real-time seperti VoIP atau video konferensi, WRR lebih sesuai karena kemampuannya untuk mengurangi *latency* dan *packet loss rate*. Selain itu, ada kemungkinan untuk menggabungkan kedua algoritma atau mengembangkan algoritma *Flow Scheduling* hibrida yang mempertimbangkan keunggulan masing-masing algoritma dalam mencapai kinerja jaringan yang optimal. Dengan demikian, penelitian lebih lanjut dapat diarahkan pada pengembangan algoritma *Flow Scheduling* yang dapat menyesuaikan diri dengan kebutuhan jaringan yang beragam dan dinamis (Iskandar et al., 2020).

REFERENSI

- Araújo, F. R. C., Oliveira, R. L., Brito, S. M., & Sampaio, L. N. (2017). A Performance Analysis of SDN Network with Flow Scheduling Algorithms. *Journal of Computer Networks and Communications*, 2017, 1-10.
- Bari, M. F., Chowdhury, S. R., Ahmed, R., & Boutaba, R. (2013). On orchestrating virtual network functions in NFV. *Proceedings of the 11th International Conference on Network and Service Management (CNSM)*, 1-9.
- Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., ... & Radoslavov, P. (2014). ONOS: towards an open, distributed SDN OS. In *Proceedings of the third workshop on Hot topics in software defined networking* (pp. 1-6).
- Chowdhury, N. M., & Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, 54(5), 862-876.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269-271.
- Hart, J., Berde, P., Kobayashi, M., Koide, T., & Parulkar, G. (2015). Network innovation using OpenFlow: A case study. In *Proceedings of the 2015 IEEE 23rd International Conference on Network Protocols (ICNP)* (pp. 384-389).
- Jammal, M., Singh, T., Shami, A., Asal, R., & Li, Y. (2014). Software defined networking: State of the art and research challenges. *Computer Networks*, 72, 74-98.
- Katevenis, M., Sidiropoulos, S., & Courcoubetis, C. (1991). Weighted round-robin cell multiplexing in a general-purpose ATM switch chip. *IEEE Journal on Selected Areas in Communications*, 9(8), 1265-1279.
- Kokilavani, T., & Amirtharajan, R. (2013). Load balanced routing mechanisms for mobile ad hoc networks. *International Journal of Computer Science & Engineering Technology*, 4(5), 536-544.
- Kompella, V. P., Pasquale, J. C., & Polyzos, G. C. (1993). Multicast routing for multimedia communication. *IEEE/ACM Transactions on Networking*, 1(3), 286-292.
- Kreutz, D., Ramos, F. M., Verissimo, P., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14-76.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2), 69-74.

ONOS. (2021). ONOS: Open Network Operating System. Retrieved from <https://onosproject.org/>

Open Networking Foundation. (2014). OpenFlow Switch Specification Version 1.5.0 (Protocol version 0x06). Retrieved from <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.5.0.noipr.pdf>