



Analisis perbandingan algoritma Dijkstra dan *Breadth First Search* pada *packet forwarding* arsitektur SDN

Usman B. Hanafi¹, M. Anantio Dwika Arie², Muhammad Rausyi Fikri^{3*}, Taufik Irfan⁴

^{1,2,3,4}Jurusan Teknik Elektro, Politeknik Negeri Bandung

Jl. Gegerkalong Hilir, Ciwaruga, Kecamatang Parongpong, Kabupaten Bandung Barat, Indonesia

¹usmanb@polban.ac.id, ²m.anantio.tcom417@polban.ac.id, ^{3*}muhammad.rausyi.tcom417@polban.ac.id,

⁴taufik.irfan@polban.ac.id

ABSTRAK

Pengelolaan perangkat jaringan seperti *switch*, *router*, dan *link* menjadi salah satu tantangan dalam memperbaiki *Quality of Service* (QoS) suatu jaringan yang berskala besar. Sebagai solusi dari permasalahan tersebut, dapat digunakan konsep arsitektur *Software Defined Networking* (SDN). SDN merupakan sebuah metode dimana perangkat jaringan dibuat secara virtual dengan cara memisahkan antara *control plane* dengan *data plane*. Proses *packet forwarding* pada arsitektur SDN bertujuan untuk mengirimkan data yang merujuk pada tabel *forwarding* dalam sistem untuk mengukur performansi QoS. Pada penelitian ini dilakukan analisis perbandingan algoritma Dijkstra dan *Breadth First Search* (BFS) pada *packet forwarding* dalam arsitektur SDN. Parameter yang diukur adalah *average jitter*, *average delay*, dan *packet loss*. Hasil penelitian dapat disimpulkan bahwa pada *packet forwarding* di topologi jaringan *ring* menggunakan IPF-Dijkstra lebih baik performansi QoS-nya daripada IPF-BFS. Sedangkan di topologi jaringan *hybrid* menunjukkan bahwa nilai performansi QoS saat *packet forwarding* menggunakan IPF-BFS lebih baik daripada IPF-Dijkstra. Pengukuran ini juga menunjukkan bahwa topologi jaringan dan algoritma *routing* yang digunakan sangat berpengaruh terhadap performansi QoS. Beberapa parameter yang mempengaruhi performansi QoS yaitu ukuran paket, kuantitas paket, dan protokol yang digunakan pada *transport layer* saat mengirimkan paket.

Kata kunci: Dijkstra, BFS, *routing*, QoS, SDN

ABSTRACT

Management of network devices such as switches, routers, and links is one of the challenges in improving the Quality of Service (QoS) of a large-scale network. As a solution to these problems, the concept of Software Defined Networking (SDN) architecture can be used. SDN is a method where network devices are created virtually by separating the control plane from the data plane. The packet forwarding process on the SDN architecture aims to send data that refers to the forwarding table in the system to measure QoS performance. In this study, a comparative analysis of the Dijkstra and Breadth First Search (BFS) algorithms was carried out on packet forwarding in the SDN architecture. Parameters measured are average jitter, average delay, and packet loss. The results of the study can be concluded that the packet forwarding in the ring network topology using IPF-Dijkstra has better QoS performance than IPF-BFS. While the hybrid network topology shows that the QoS performance value when packet forwarding using IPF-BFS is better than IPF-Dijkstra. This measurement also shows that the network topology and routing algorithm used greatly affect the QoS performance. Several parameters that affect QoS performance are packet size, packet quantity, and the protocol used at the transport layer when sending packets.

Keywords : Dijkstra, BFS, QoS, *routing*, SDN

1. PENDAHULUAN

Dewasa ini, perkembangan teknologi komunikasi menyebabkan terjadinya peningkatan penggunaan perangkat teknologi komunikasi jaringan. Pada teknologi jaringan tersebut masih tergolong konvensional dimana teknologi tersebut memiliki tingkat fleksibilitas rendah dan kompleksitas yang tinggi, sehingga dalam melakukan konfigurasi protokol-protokol masih bersifat *on-site* dan *embedded* di tiap perangkat [1]. Agar tidak terjadi kesalahan yang diakibatkan oleh kompleksitas dan fleksibilitas dalam melakukan konfigurasi suatu jaringan, terdapat konsep jaringan

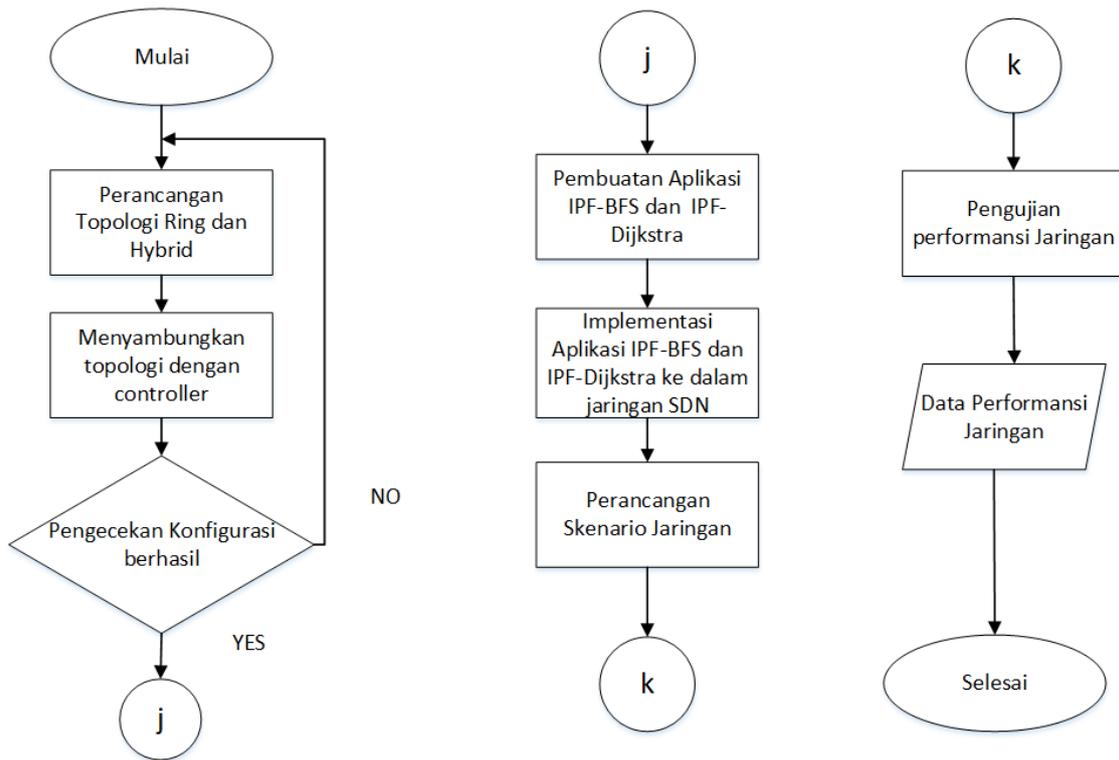
Software Defined Networking (SDN). Konsep jaringan SDN merupakan konsep jaringan modern yang memisahkan antara *data plane* dan *control plane*, dimana *data plane* yang berfungsi sebagai infrastruktur jaringan dan *control plane* berfungsi sebagai pengelola protokol-protokol jaringan yang akan diimplementasikan di dalam *data plane* [2]. Pada SDN pengelolaan protokol jaringan menjadi terpusat atau *centralized* sehingga manajemen jaringan menjadi efisien.

Penerapan SDN dan implementasi *routing* saat *packet forwarding* pada arsitektur SDN sudah pernah dikembangkan sebelumnya. Penelitian [2] melakukan pengukuran parameter QoS pada algoritma Floyd-Warshall. Pada pengujiannya, digunakan *controller* Ryu dengan menguji beberapa pengujian paket berjenis data, *voice*, dan juga video dengan menggunakan aplikasi D-ITG. Parameter yang diukur yaitu *delay* dan *packet loss*. Pada jurnal tersebut penelitian mengadopsi pembuatan mekanisme simulasi pengiriman suatu paket. Kemudian penelitian [3] menjelaskan bagaimana mengimplementasikan algoritma Dijkstra untuk mengirimkan paket dari sumber ke penerima pada arsitektur jaringan SDN menggunakan aplikasi ONOS. Berikutnya, peneliti [4] membahas pengukuran *latency* dalam *packet forwarding* menggunakan algoritma *routing* Dijkstra. Selanjutnya, penelitian [5] membahas bagaimana mengimplementasikan *routing Open Shortest Path First* (OSPF) dalam *packet forwarding* dan juga mengukur nilai QoS yaitu *throughput*, *jitter*, *delay*, dan *packet loss*. Penelitian [6] juga menjelaskan analisis terhadap parameter QoS suatu arsitektur jaringan. Pengukuran ini secara spesifik menganalisis performansi OSPF sebagai *routing* dalam arsitektur SDN. Dalam jurnal ini, penulis mempertimbangkan *protocol routing* OSPF untuk digunakan dalam riset ini. Berikutnya, penelitian [7] membuat visualisasi terhadap parameter *packet sent*, *packet received*, *bytes sent*, dan *bytes received*. Informasi parameter tersebut didapatkan dari ONOS REST API Core. Selain itu, terdapat penelitian juga yang membahas tentang penggunaan algoritma BFS dan DFS. Pada penelitian tersebut *controller* yang digunakan yaitu Ryu, dimana pada penelitian ini bertujuan untuk membandingkan waktu konvergensi antara algoritma *Breadth First Search* (BFS) dan *Depth First Search* (DFS) [8]. Berikutnya, penelitian [9] membahas tentang penggunaan algoritma *Routing Information Protocol* (RIP) pada jaringan SDN, setelah itu membahas tentang QoS pada hasil pengirimannya. Selain itu terdapat perbandingan pemakaian *controller* antara Ryu dan ONOS, dimana perbandingan QoS pada ONOS lebih baik dibandingkan Ryu [10].

Apabila dilihat dari penelitian sebelumnya, secara tidak langsung penelitian ini berusaha mengembangkan hasil dari beberapa penelitian di atas. Namun, penelitian ini memiliki perbedaan yaitu membandingkan dua algoritma yaitu algoritma Dijkstra dan BFS dalam *routing* saat *packet forwarding*. Beralih ke tujuan dari penelitian ini, dimana tujuannya yang pertama ini yaitu menganalisis performansi QoS pada *packet forwarding* menggunakan IPF-Dijkstra dan IPF-BFS di suatu arsitektur SDN. Kedua, penelitian ini memiliki tujuan yaitu menganalisis parameter lain yang berpengaruh dalam performansi QoS pada pengiriman paket selain dari algoritma *routing* yang digunakan.

2. METODE PENELITIAN

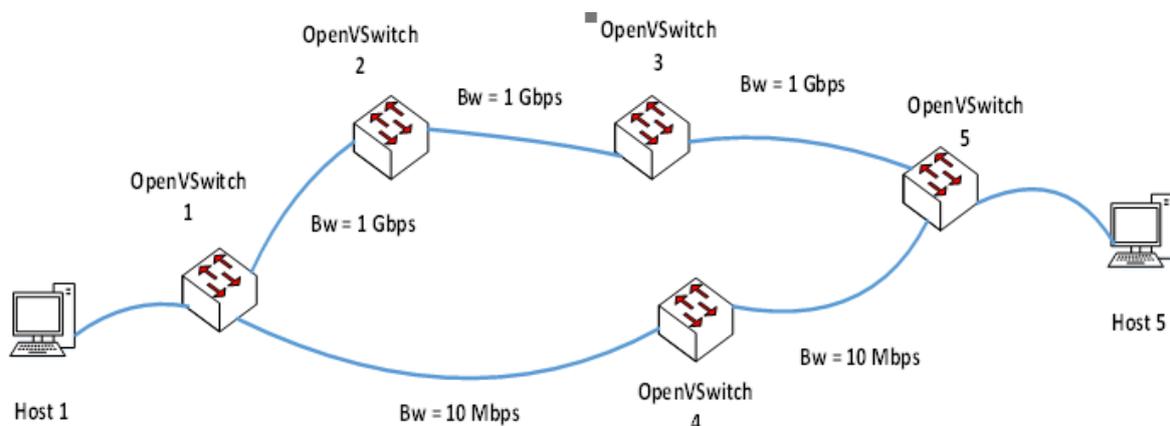
Dalam melakukan penelitian ini, berikut kami tampilkan alurnya dalam bentuk *flowchart* yang ada pada Gambar 1. Pada *flowchart* tersebut menunjukkan metode penelitian yang dilakukan dalam mengerjakan penelitian ini. Proses pertama ialah merancang topologi jaringan yang digunakan dalam penelitian pada arsitektur SDN. Setelah itu membuat aplikasi penerapan algoritma *routing* Dijkstra dan BFS yang dinamakan IPF-Dijkstra dan IPF-BFS. Aplikasi tersebut digunakan dalam menerapkan algoritma *routing* Dijkstra dan BFS saat *packet forwarding*. Kemudian perlu dilakukan perancangan skenario pengujian saat menguji QoS pada *packet forwarding*. Proses yang terakhir yaitu menentukan data-data performansi yang diujikan pada hasil pengiriman paket yang dilakukan. Kemudian tahapan-tahapan yang dilakukan pada penelitian tersebut dapat dijelaskan pada sub-bab berikutnya



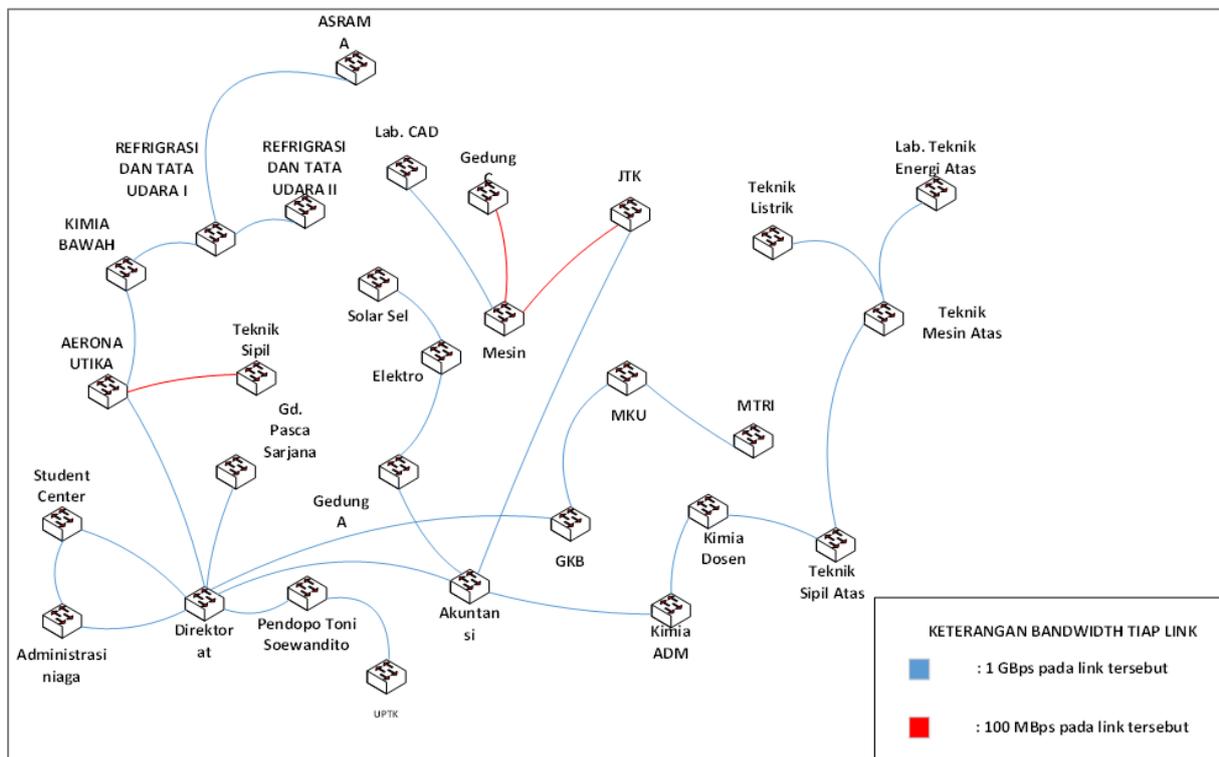
Gambar 1. Flowchart tahapan penelitian

2.1 Perancangan Topologi Jaringan

Dalam penelitian ini, topologi jaringan yang akan digunakan adalah topologi jaringan *ring* dan topologi jaringan *hybrid*. Topologi jaringan *ring* digunakan untuk menguji *routing* pada dua jalur yang berbeda antara algoritma Dijkstra dan BFS saat *packet forwarding*. Selanjutnya topologi jaringan *hybrid* digunakan sebagai referensi implementasi konsep jaringan SDN pada infrastruktur jaringan yang ada di Politeknik Negeri Bandung. Topologi jaringan *hybrid* ini mengadopsi dari topologi jaringan yang digunakan oleh Politeknik Negeri Bandung, dimana topologi jaringan ini merupakan gabungan antara topologi jaringan *ring* dengan topologi jaringan *tree*. Berikut ini perancangan topologi jaringan *ring* yang dapat dilihat pada Gambar 2. Lalu untuk perancangan topologi jaringan *hybrid* dapat dilihat pada Gambar 3.



Gambar 2. Perancangan topologi jaringan *ring*



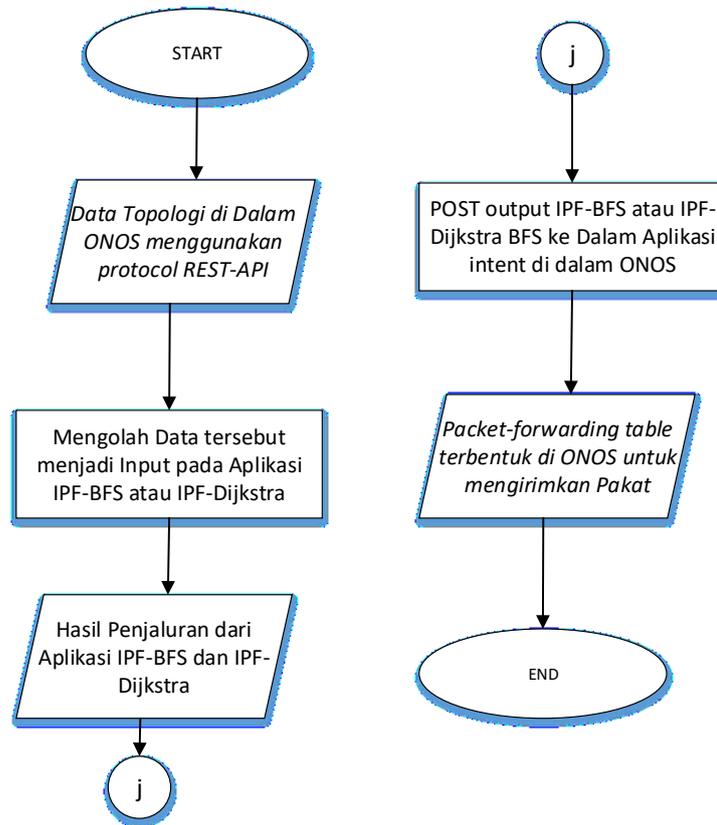
Gambar 3. Rancangan topologi jaringan *hybrid*

2.2 Pembuatan Aplikasi IPF-Dijkstra dan IPF-BFS

Algoritma Dijkstra merupakan algoritma *routing* yang mempertimbangkan nilai *cost* yang rendah saat *packet forwarding*. Dalam implementasinya pada *routing* saat *packet forwarding*, *cost* yang diperhitungkan berasal dari nilai *reference bandwidth* pada kabel *ethernet* yang bernilai 10^{10} bit per detik. Nilai *reference bandwidth* tersebut dibagi dengan nilai *bandwidth* dari kabel. Dengan demikian, algoritma ini akan selalu memilih jalur dengan nilai *bandwidth* yang tinggi agar memiliki *cost* yang rendah. Tujuan dari algoritma ini adalah memilih jalur yang tercepat, dimana pengiriman paket akan lebih cepat sampai ke tujuan apabila menggunakan jalur dengan nilai *bandwidth* yang tinggi [4].

BFS merupakan algoritma *routing* terpendek yang bersifat *unweighted graph* pada graf atau *tree* [8]. Metode pencarian jalurnya yaitu secara melebar di setiap *level* ke *level* selanjutnya jika node pada *level* telah dikunjungi sebelumnya. Ekspansi akan dilakukan ke setiap *node* tetangga di *level* yang sama. Algoritma ini menggunakan struktur data *queue* dengan tingkat kompleksitas $G(V+E)$. Dimana pencarian jalur pada node E akan menelusuri V pada G untuk mencari jalur dengan jumlah E paling rendah.

Pada penelitian ini, kedua algoritma tersebut akan dijadikan aplikasi yang akan digunakan pada jaringan SDN sebagai *routing* pada *packet forwarding* di ONOS. Gambar 4 menampilkan *flowchart* proses implementasi aplikasi IPF-Dijkstra dan IPF-BFS. Aplikasi tersebut berbasis Python dan akan menghasilkan jalur dari masing-masing algoritma. Kemudian hasil tersebut diimplementasikan pada *controller* ONOS melalui aplikasi Intent-Path Forwarding (IPF), lalu dihasilkan jalur pengiriman sesuai aplikasi IPF yang telah diimplementasikan dari hasil jalur aplikasi IPF-Dijkstra dan IPF-BFS.



Gambar 4. Proses implementasi aplikasi IPF-Dijkstra dan IPF-BFS di ONOS

2.3 Skenario Pengujian

Pengujian yang dilakukan bersifat simulasi, dimana *environment* tersebut berjalan pada suatu *virtual machine*. *Virtual machine* yang digunakan yaitu VMWare dengan sistem operasi Ubuntu 18.04. Seperti yang disinggung pada penjelasan sebelumnya, bahwa untuk membangun arsitektur SDN dibutuhkan *data plane* dan *control plane*. Untuk membuat *data plane* digunakan aplikasi Mininet yang berfungsi untuk membangun dan menjalankan topologi jaringan *ring* dan topologi jaringan *hybrid*. Lalu sebagai *control plane* digunakan aplikasi ONOS yang berfungsi untuk menjalankan aplikasi BFS dan Dijkstra sebagai *routing protocol* pada topologi jaringan *ring* dan topologi jaringan *hybrid* yang ada pada Mininet. Aplikasi tersebut secara spesifiknya berjalan pada *application layer* yang dimiliki ONOS. Setelah dijalankan, kedua aplikasi tersebut akan dibandingkan efektifitasnya dalam simulasi pengiriman paket melalui D-ITG.

Simulasi pengiriman paket data tersebut memiliki enam skenario, dimana skenario tersebut ditampilkan pada Tabel 1. Simulasi pengiriman paket data tersebut dijalankan baik di topologi jaringan *ring* maupun topologi jaringan *hybrid*. Pada skenario 1 dan 4, pengujian pengiriman merupakan pengiriman paket data pada keadaan trafik di *link* jaringan dalam keadaan normal atau dapat dibilang tidak penuh. Sedangkan skenario lainnya, merupakan skenario pengujian pada pengiriman dalam kondisi trafik penuh. Simulasi pengiriman paket data ini nantinya akan diukur nilai parameter QoS-nya, dimana parameter tersebut yaitu *average delay*, *average jitter*, dan *packet loss* sesuai dengan persamaan (1) sampai (3).

$$Packet\ loss\ (\%) = \frac{(Paket\ data\ dikirim - Paket\ data\ diterima)}{Paket\ data\ yang\ dikirim} \quad (1)$$

$$Average\ jitter\ (ms) = \frac{\sum Variasi\ Delay}{\sum Paket\ yang\ Diterima} \quad (2)$$

$$Average\ delay\ (ms) = \frac{Total\ Delay}{Total\ paket\ yang\ Diterima} \quad (3)$$

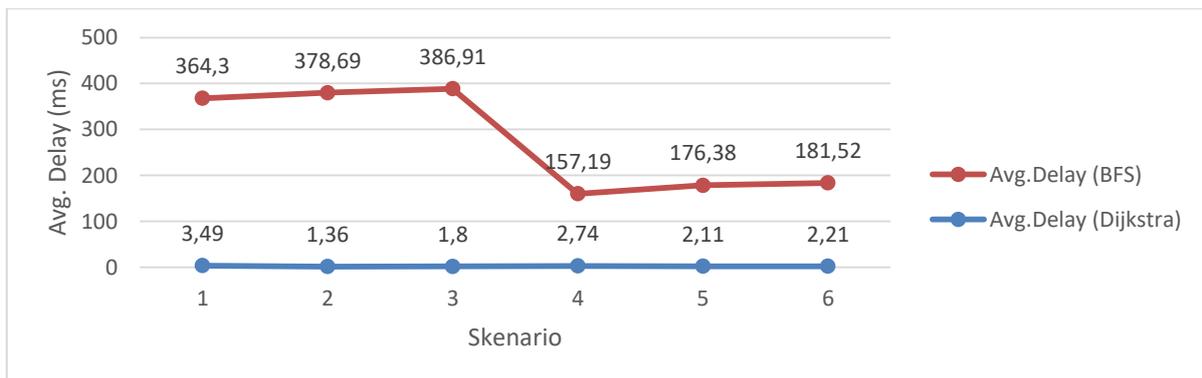
dengan variasi $delay = delay(n) - delay(n-1)$ dan total delay = $delay\ 0 + \dots + delay(n-1) + delay(n)$.

Tabel 1. Tabel skenario simulasi pengiriman paket data

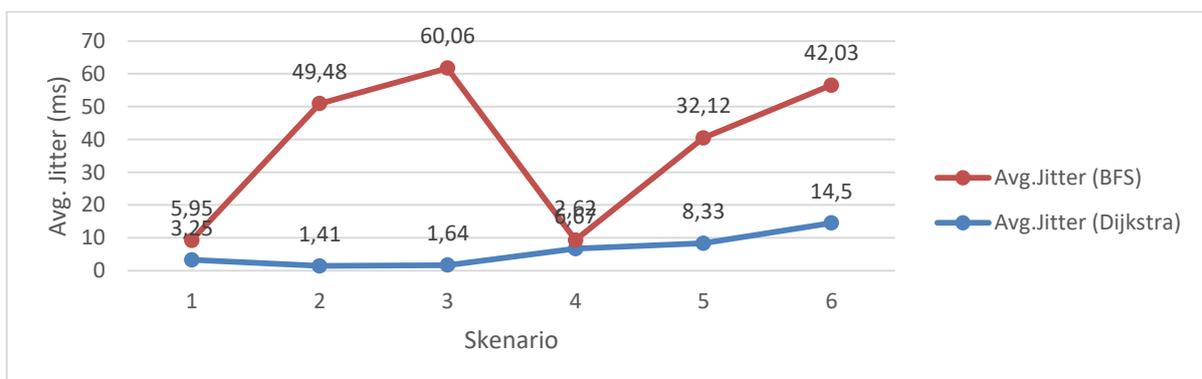
Topologi	Kode skenario	Node awal	Node tujuan	Protokol paket	Ukuran paket (KB)	Kuantitas paket (pps)	Waktu interval (s)
<i>Ring</i>	1	H1	H5	TCP	100	20	30
	2			TCP	100	50	30
	3			TCP	100	100	30
	4			UDP	100	20	30
	5			UDP	100	50	30
	6			UDP	100	100	30
<i>Hybrid</i>	1	Asrama	Adminis-trasi Niaga	TCP	100	100	30
	2			TCP	100	1000	30
	3			TCP	100	10000	30
	4			UDP	100	100	30
	5			UDP	100	1000	30
	6			UDP	100	10000	30

3. HASIL DAN PEMBAHASAN

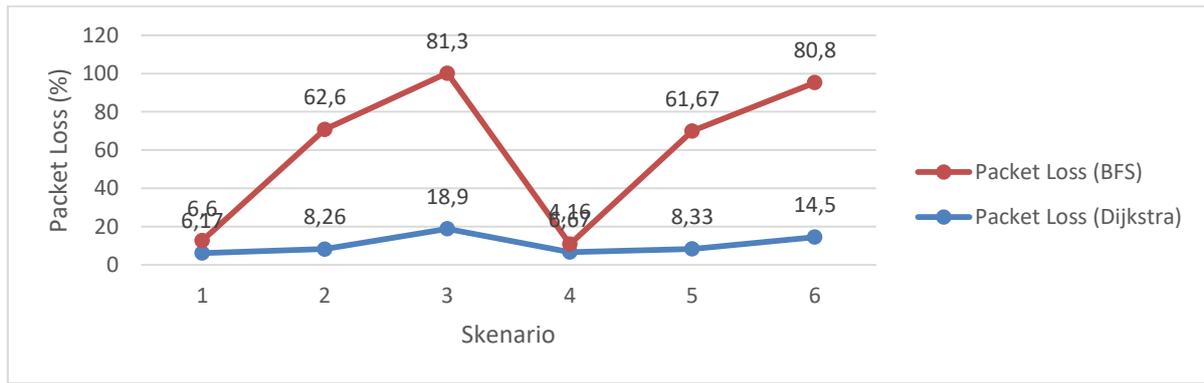
Pada bagian ini didapatkan nilai parameter QoS dari pengukuran pada simulasi penggunaan aplikasi Dijkstra dan BFS pada topologi jaringan *ring* dan topologi jaringan *hybrid* sesuai dengan perancangan skenario yang ada. Berikut grafik nilai ketiga parameter tersebut dalam pengujian pada topologi jaringan *ring* menggunakan *routing protocol* Dijkstra dan BFS dapat dilihat pada Gambar 5, Gambar 6, dan Gambar 7.



Gambar 5. Grafik perbandingan *average delay* pada topologi jaringan *ring*



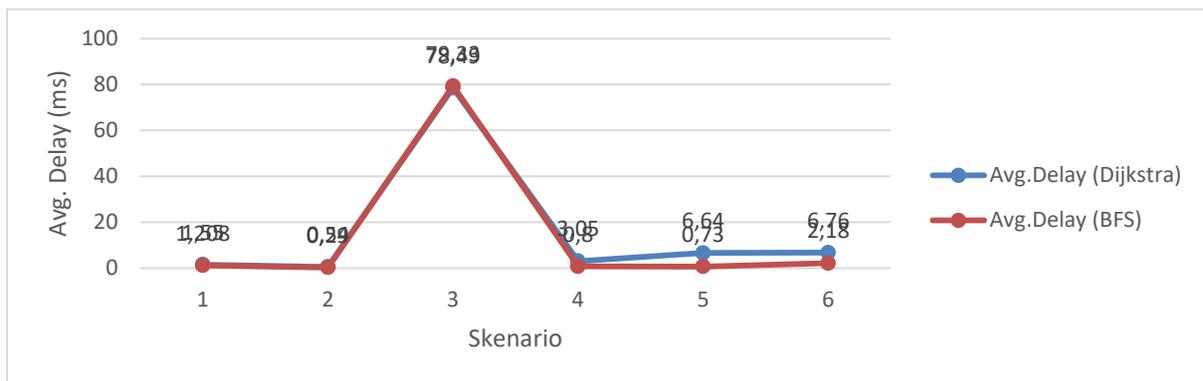
Gambar 6. Grafik perbandingan *average jitter* pada topologi jaringan *ring*



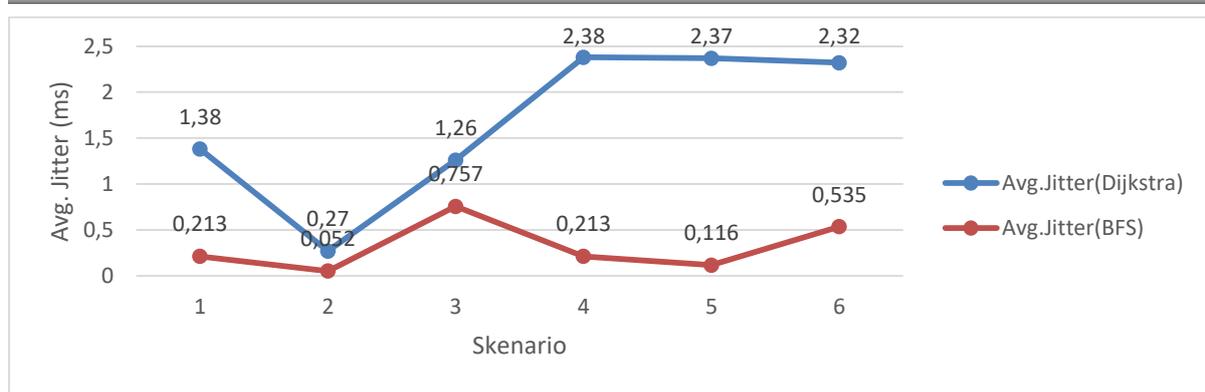
Gambar 7. Grafik perbandingan *packet loss* pada topologi jaringan *ring*

Pada hasil perbandingan dapat dilihat bahwa nilai performansi QoS pada *packet forwarding* menggunakan IPF-Dijkstra cenderung lebih baik daripada menggunakan IPF-BFS. Selain itu dapat dilihat juga untuk nilai pengiriman paket TCP memiliki performansi QoS yang lebih besar dibandingkan pengiriman paket UDP. Pada pengiriman menggunakan aplikasi Dijkstra didapatkan rute yang dipilih memiliki *bandwidth* sebesar 1 Gbps. Namun, efektif *bandwidth* yang dapat digunakan dalam satu pengiriman besarnya 324 Mbps. Sedangkan *routing* dalam *packet forwarding* menggunakan aplikasi BFS memiliki nilai *bandwidth* yang lebih rendah, yaitu 10 Mbps. Pada jalur tersebut *bandwidth* efektif yang dapat digunakan memiliki besar 9,42 Mbps. Hal ini membuktikan bahwa algoritma Dijkstra cenderung memilih jalur yang memiliki *bandwidth* lebih besar saat *routing*. Apabila dihitung dalam setiap detiknya pada setiap skenario, data yang perlu dikirim memiliki *size* sebesar 2 MBps (skenario 1 dan 4), 5 MBps (skenario 2 dan 5), dan 10 MBps (Skenario 3 dan 6). Apabila besaran tersebut dikonversikan secara berurut memiliki ukuran 16 Mbps, 40 Mbps, dan 80 Mbps. Ukuran ini tentu memiliki besar kapasitas yang melebihi *bandwidth* efektif pada link yang digunakan pada pengiriman menggunakan aplikasi BFS. Sehingga didapatkan kondisi trafik yang penuh pada rute yang dipilih oleh *routing protocol* BFS. Sebaliknya pada rute yang dipilih *routing protocol* dijkstra, kondisi trafik cenderung normal. Hal ini dikarenakan *bandwidth* pada link yang dipilih pada rute pengiriman aplikasi dijkstra cenderung memiliki kapasitas yang besar.

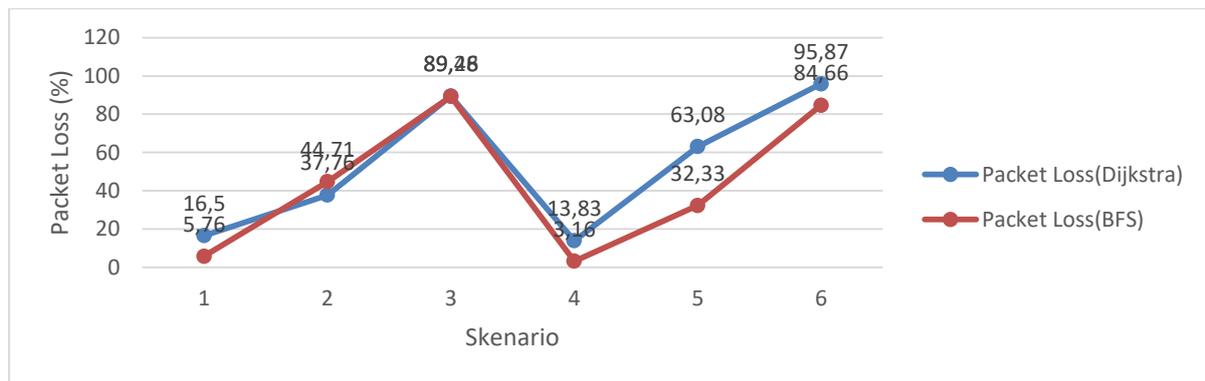
Selanjutnya, ketiga parameter tersebut juga diukur pada simulasi yang dilakukan di topologi jaringan *hybrid*. Adapun hasil perbandingannya dapat dilihat pada Gambar 8, Gambar 9, dan Gambar 10.



Gambar 8. Grafik perbandingan *average delay* pada topologi jaringan *hybrid*



Gambar 9. Grafik perbandingan *average jitter* pada topologi jaringan *hybrid*



Gambar 10. Grafik perbandingan *packet loss* pada topologi jaringan *hybrid*

Berdasarkan hasil perbandingan, dapat dilihat *packet forwarding* menggunakan IPF-BFS cenderung lebih baik performansi QoS-nya daripada menggunakan IPF-Dijkstra. Pada topologi ini terdapat kesamaan *routing* pada kedua algoritma. Hal tersebut dikarenakan topologi jaringan *hybrid* merupakan topologi jaringan gabungan topologi jaringan ring dengan topologi jaringan *tree*, dimana hanya ada percabangan pada jalur menuju *node* Direktorat. Pada jalur selain itu hanya ada satu jalur dalam *packet forwarding*, sehingga proses *routing* menggunakan algoritma BFS akan lebih efektif. Hal ini dikarenakan proses *routing* pada algoritma BFS memiliki mekanisme pencarian secara melebar dan per-level sehingga untuk memutuskan jalur yang akan digunakan lebih efisien lebih cepat. Pada metode pencarian jalur tersebut, tentu akan lebih efektif dibandingkan metode pencarian jalur yang digunakan pada algoritma Dijkstra.

4. KESIMPULAN

Pada pengukuran performansi QoS saat *packet forwarding* menggunakan algoritma Dijkstra dan BFS dapat disimpulkan bahwa pada topologi jaringan *ring*, algoritma *routing* yang tepat dalam *packet forwarding* yaitu algoritma Dijkstra. Hal ini dikarenakan algoritma Dijkstra cenderung memilih jalur dengan *bandwidth* yang lebih besar dengan tujuan pengiriman paket memiliki hambatan yang sangat kecil. Sedangkan pada topologi jaringan *hybrid*, algoritma *routing* yang tepat dalam *packet forwarding* yaitu algoritma Dijkstra. Hal ini dikarenakan topologi jaringan *hybrid* pada proses *packet forwarding* hanya memiliki satu jalur saja. Sehingga berapapun *bandwidth* yang ada pada suatu *link* tidak berpengaruh. Selain algoritma *routing* dan topologi jaringan yang digunakan, adapun parameter yang mempengaruhi nilai QoS yang diukur. Parameter tersebut yaitu protokol yang digunakan pada *transport layer* dan juga besar serta kuantitas dari suatu paket yang dikirimkan. Ketiga parameter tersebut sangat berpengaruh terhadap nilai QoS apabila dilihat dari skenario pengiriman yang disimulasikan. Untuk penelitian selanjutnya, diperlukan simulasi secara *multipoint to multipoint* ataupun *point to multipoint* dalam mengimplementasikan algoritma *routing* Dijkstra maupun BFS dalam *packet forwarding* pada arsitektur SDN.

REFERENSI

- [1] R. M. Negara and R. Tulloh, "Analisis Simulasi Penerapan Algoritma OSPF Menggunakan RouteFlow pada Jaringan Software Defined," *Jurnal Infotel (Informasi-Telekomunikasi-Elektronika)*, vol. 9, no. 1, pp. 75-83, 2017.
- [2] I. A. Saputra and S. N. Hertiana, "Uji Performansi Algoritma floyd-Warshall pada Jaringan Software Defined Network (SDN)," *Jurnal Elektronika dan Telekomunikasi*, vol. 16, no. 2, pp. 52-58, 2016.
- [3] T. Irfan, R. Hakimi, A. C. Risdianto, and E. Mulyana, "ONOS Intent Path Forwarding using Dijkstra Algorithm," in *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, 2019, pp. 549-554.
- [4] F. Ramadhan, R. Primananda, and W. Yahya, "Implementasi Routing Berbasis Algoritme Dijkstra Pada Software Defined," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 7, pp. 2531-2541, 2018.
- [5] A. Irmawati, I. D. Irawati, and Y. S. Hariyani, "Implementation of OSPF Routing Protocol for Software Defined Network Base on Routeflow," *e-Proceeding of Applied Science*, vol. 3, no. 2, pp. 1067-1074, 2017.
- [6] A. Z. Pramudita and I. M. Suartana, "Perbandingan Performa Controller OpenDayLight dan Ryu pada Arsitektur Software Defined Network," *Journal of Informatics and Computer Science*, vol. 1, no. 4, pp. 2531-2541, 2018.
- [7] F. Djohar, *Visualisasi Statistik Southbound Interface*, ITB, Bandung, 2020.
- [8] A. K. Sriastunti, R. Primananda, and W. Yahya, "Implementasi Routing pada Openflow Software-Defined Network dengan Algoritme Depth-first dan Breadth-First Search," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, No. 8, pp. 8112-8120, 2019.
- [9] I. Hidayah, I. D. Irawati, and Y. S. Hariyani, "Implementasi RIP pada Jaringan Berbasis Software Defined Network," *e-Proceeding of applied science*, vol. 3, no. 2, pp. 1039-1046, 2017.
- [10] R. M. Negara and R. Tulloh, "Single Board Computer Performance Comparison Analysis in Software Defined," *IJSST*, vol. 2, pp. 1-6, 2017.

