

ANALISIS PERFORMANSI JARINGAN SOFTWARE DEFINE NETWORK (SDN) MENGGUNAKAN ARUBA VAN CONTROLLER

Rifqi Fauzan^{1*)}, Syariful Ikhwan², Jafaruddin Gusti Amri Ginting³

^{1,2,3}Jurusan Teknik Telekomunikasi, Fakultas Teknik Elektro dan Telekomunikasi, IT Telkom Purwokerto

^{1,2,3}Jl. D.I Panjaitan no.128, Purwokerto, 53147, Indonesia

email: ¹18101227@ittelkom-pwt.ac.id, ²syariful@ittelkom-pwt.ac.id, ³jafaruddin@ittelkom-pwt.ac.id

Special Issue on Seminar Nasional - Inovasi Dalam Teknologi Informasi & Teknologi Pembelajaran 2019

Abstract – SDN technology is a new architecture in the network that separates the control plan from the data plan. Aruba VAN Controller provides centralized control in a network architecture that runs on the OpenFlow protocol. This study conducted an SDN network simulation using the Aruba VAN Controller by applying the Narmox Spear application to simplify network configuration. IP addressing with DHCP server, for testing scenarios using UDP protocol packet delivery with packets sent at 1000 Bytes, the number of packets sent is 100 packets/s, then given background traffic variations of 50 Mbps, 100 Mbps, 150 Mbps and 200 Mbps to networks that have two different link bandwidth values as a comparison for analyzing the results of testing the QoS parameters, which are 100 Mbps and 200 Mbps. The test results are obtained when background traffic exceeds the link bandwidth capacity, the resulting throughput will be smaller on the destination side by 485.64 Kb/s, delay increases by 240.21 ms, then for jitter by 2.48 ms, resulting in overload resulting in packet loss of 37.41%. However, when background traffic increases and does not exceed the specified link bandwidth capacity, the overall QoS parameter value is generated in the good category, namely throughput of 774.42 Kb/s, delay of 0.41 ms, jitter of 0.37 ms and packet loss of 0.00%.

Keyword – SDN, Aruba VAN Controller, OpenFlow, DHCP, UDP, QoS

Abstrak – Teknologi SDN merupakan arsitektur baru dalam jaringan yang memisahkan antara *control plan* dan *data plan*. Aruba VAN Controller menyediakan pemusatan *control* dalam suatu arsitektur jaringan yang berjalan pada protokol OpenFlow. Penelitian ini melakukan simulasi jaringan SDN menggunakan Aruba VAN Controller dengan menerapkan aplikasi Narmox Spear untuk mempermudah konfigurasi jaringan. Pengalamatan IP dengan DHCP server, untuk skenario pengujian menggunakan pengiriman paket protokol UDP dengan besar paket yang dikirim sebesar 1000 Bytes, jumlah paket yang dikirim sebanyak 100 paket/s, kemudian diberikan variasi *background traffic* sebesar 50 Mbps, 100 Mbps, 150 Mbps dan 200 Mbps terhadap jaringan yang memiliki dua nilai *bandwidth link* yang berbeda sebagai pembandingan untuk analisa hasil pengujian parameter QoS, yaitu sebesar 100 Mbps dan 200 Mbps. Hasil pengujian didapatkan ketika *background traffic* melebihi kapasitas *bandwidth link*, maka *throughput* yang dihasilkan akan semakin kecil disisi *destination* sebesar 485,64 Kb/s, *delay* semakin meningkat sebesar 240,21 ms, kemudian untuk *jitter* sebesar 2,48 ms, sehingga terjadi *overload* yang

mengakibatkan adanya *packet loss* sebesar 37,41%. Namun, ketika *background traffic* meningkat dan tidak melebihi kapasitas *bandwidth link* yang sudah ditentukan, maka secara keseluruhan nilai parameter QoS dihasilkan dengan kategori baik, yaitu *throughput* sebesar 774,42 Kb/s, *delay* sebesar 0,41 ms, *jitter* sebesar 0,37 ms dan *packet loss* sebesar 0,00%.

Kata Kunci – SDN, Aruba VAN Controller, OpenFlow, DHCP, UDP, QoS

I. PENDAHULUAN

Seiring perkembangan zaman dari tahun ke tahun, teknologi jaringan telah berkembang lebih lambat dibandingkan dengan teknologi komunikasi lainnya. *Switch* dan *router* merupakan salah satu peralatan jaringan yang telah banyak dikembangkan oleh vendor, setiap vendor membuat *firmware* dan perangkat lunak untuk dapat mengoperasikan perangkat jaringan dari setiap vendor itu sendiri secara eksklusif dan tertutup. Hal ini memperlambat kemajuan inovasi dalam teknologi jaringan dan menyebabkan peningkatan manajemen jaringan serta biaya operasi jaringan, ketika ada layanan baru atau ada perangkat dengan teknologi baru yang akan diterapkan pada jaringan yang sudah ada [1]. Manajemen jaringan menjadi semakin rumit mengatur lalu-lintas Internet seoptimal mungkin untuk memberikan pelayanan yang berkualitas [2], seorang teknisi memungkinkan harus mengakses secara langsung pada perangkat jaringan (*router*, *switch*).

Namun pada teknologi konvensional saat ini, kapasitas dan kompleksitas dari banyak pengembangan modern dianggap masih kurang bagi penyedia layanan Telekomunikasi. Maka dari itu, meningkatnya biaya pengoperasian peralatan jaringan, kebutuhan inovasi baru dalam teknologi jaringan dan secara khusus meningkatnya tuntutan modernisasi *data center* untuk manajemen jaringan secara terpusat. Hal ini mendorong teknologi jaringan dari metode dan protokol konvensional menuju paradigma SDN (*Software Defined Network*) yang lebih terbuka dan berinovasi [3].

SDN merupakan suatu arsitektur jaringan yang mengoptimalkan dan menyederhanakan operasi jaringan dengan lebih interaksi (*provisioning*, *messaging*, and *alarms*) antara aplikasi, layanan jaringan dan perangkat. Arsitektur SDN dapat dibangun dengan menggunakan pusat *control* secara *logical* dan *programmable*, yang sering direalisasikan sebagai *controller* SDN untuk manajemen jaringan, menyederhanakan operasi jaringan dan

*) penulis korespondensi: Rifqi Fauzan
Email: rifqi.fzn@gmail.com

memfasilitasi komunikasi antara *layer application* dengan *layer infrastructure* [4]. Berdasarkan adanya pemisahan antara *control plan* dan *data plan*, arsitektur SDN memberikan kemudahan pada administrator dalam pengelolaan jaringan serta membuat arsitektur jaringan menjadi lebih efisien, hemat biaya dan inovatif [5].

Aruba VAN Controller menyediakan pemusatan *control* dalam suatu arsitektur jaringan SDN untuk manajemen jaringan Internet secara terpusat pada halaman web yang menggunakan pemrograman bahasa *java*. Protokol *OpenFlow* sebagai *interface* antara fisik dan virtual, sehingga dapat menyederhanakan manajemen jaringan, *provisioning* dan *orchestrates*. Controller ini sebagai perangkat lunak mengelola perilaku *packet forwarding*. Port jaringan, *link* dan topologi semuanya dapat terlihat pada halaman web, sehingga memungkinkan kebijakan pengelolaan jaringan secara terpusat dan pemilihan jalur lebih efektif berdasarkan pada pandangan global saat ini dan dinamis daripada arsitektur jaringan konvensional [6]. Pada Aruba VAN Controller memiliki fitur aplikasi Narmox spear sebagai *platform network* administrator yang memungkinkan administrator dapat mengkonfigurasi suatu aktivitas jaringan SDN tanpa menggunakan Bahasa pemrograman, hal tersebut memberikan fungsi baru pada Aruba VAN Controller untuk menyediakan *platform* siap pakai atas dasar kondisi dan tindakan pada masalah jaringan yang lebih kompleks. Aplikasi narmox spear memberikan skema program jaringan serbaguna yang dapat membuat *flow tabel* dan memungkinkan mendefinisikan *script* untuk lingkungan jaringan tertentu.

Hasil penelitian yang dilakukan oleh Muhammad Hikam dan Nur Rohman Rosyid, bahwa Arsitektur SDN dapat diterapkan ke dalam perangkat MikroTik menggunakan berbagai jenis topologi. Selain itu arsitektur SDN juga memiliki fleksibilitas dan skalabilitas yang baik, meskipun semakin besar jaringan yang dimiliki maka performansi juga semakin menurun [7]. Skenario penelitian tersebut melakukan analisis perbandingan arsitektur jaringan konvensional dengan arsitektur jaringan SDN yang menggunakan *controller OpenDaylight* dan menggunakan 3 topologi yang berbeda pada masing-masing arsitektur jaringannya. Penelitian tersebut belum dilakukan mengenai pengaruh *controller* lainnya dan hanya berfokus pada performansi jaringan dari perbandingan arsitektur jaringan konvensional dengan arsitektur jaringan SDN. Maka dari itu penulis akan melakukan penelitian dengan judul “Analisis Performansi Jaringan SDN (Software Defined Network) menggunakan Aruba VAN Controller”, penelitian ini diharapkan dapat mengetahui performansi jaringan dan karakteristik dari *controller* terbaru yang dikembangkan oleh Hewlett Packard Enterprise Company.

II. PENELITIAN YANG TERKAIT

Penelitian yang dikaji oleh Ameer Mosa Al-Sadi, Ali Al-Sherbaz, James Xue dan Scott Turner pada tahun 2019 yang berjudul “Developing an Asynchronous Technique to Evaluate the Performance of SDN HP Aruba Switch and OVS” [8], melakukan perbandingan antara jaringan SDN menggunakan (OVS) *Open Virtual Switch* di dalam *emulator Mininet* dengan jaringan SDN menggunakan perangkat *switch* HP Aruba 3810M dan Aruba HP VAN SDN versi 2.7 sebagai *controller*. Pengujian dalam penelitian tersebut mengukur QoS (*Quality of Service*) seperti *delay*, *throughput*, *packet*

loss dan *jitter*, kemudian diberikan beban trafik (*background traffic*) pada jaringan SDN. Penelitian tersebut berfokus pada pengujian dengan sejumlah protokol ICMP, TCP dan UDP mengevaluasi *latency* pada jaringan SDN secara akurat, dengan teknik *asynchronous* untuk menunjukkan hasil yang lebih tepat. Selain itu, berfokus pada *flow setup latency* yang disebabkan oleh *Controller* SDN saat mengatur aturan kepada perangkat *switch* sebagai *packet forwarding*. Hasil penelitian ini menunjukkan perbedaan bahwa perangkat *switch* HP Aruba 3810M memiliki *latency* yang lebih tinggi karena pada saat *flow setup time* adanya tambahan sumber *delay*, yang tidak dimiliki pada *emulator Mininet*.

Penelitian Romi Afan melakukan perbandingan antara kontroler RYU dan POX pada performansi jaringan SDN [9]. Penelitiannya menggunakan tiga skenario topologi *full mesh* yang berjumlah *switch* 6, 8 dan 10 dan masing-masing *switch* mempunyai 2 *host* yang terhubung. Berdasarkan simulasi yang dilakukannya nilai QoS yang didapat untuk kedua kontroler masih dalam standar ITU-T G.1010 yaitu *delay* kecil dari 15s untuk data, kecil dari 150ms untuk Voip dan kecil dari 10s untuk video.

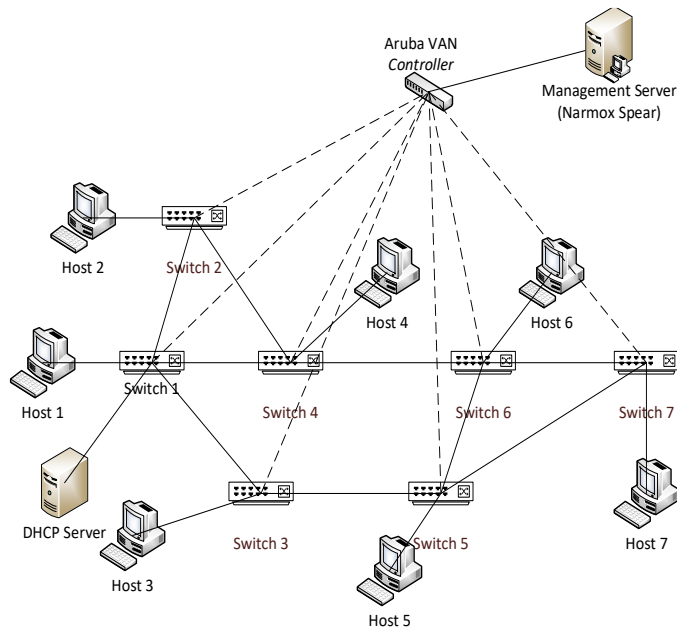
Penelitian yang dikaji oleh I Putu Eka Pratama dan I Made Adhiarta Wikantyasa [10], melakukan Analisis *performance device* Openflow *switch* pada jaringan SDN yang dibuat dengan menggunakan ONOS *Controller*. Topologi *custom* yang dibuat menggunakan Bahasa pemrograman python pada mininet API dengan 18 *host* dan 10 *switch*, kemudian dilakukan pengujian QoS menggunakan Iperf3 dan untuk keduanya melakukan analisis performansi dari *device* Openflow *switch* pada jaringan SDN. Hasil penelitiannya mendapatkan nilai QoS dari pengujian menggunakan tools Iperf3 dengan nilai rata-rata *jitter* 0,1585 ms, *packet loss* 7,428 %, *throughput* UDP 2,3819 bit/sec, *throughput* UTP 11,912 bit/sec dan rata-rata *bandwidth* 9,96 Mbps dari setiap 18 *host*. Hasil pada analisis terhadap *device* Openflow *switch* yang dimana menyaring seluruh *packet* yang melewati 10 *switch*, memperoleh jumlah total data *packet received* 2,549,271 ; *packet send* 17,137,632 ; *byte received* 58,277,697,419 ; *byte send* 229,525,181,061.

III. METODE PENELITIAN

Pada penelitian ini membutuhkan beberapa perangkat sebagai pendukung untuk melakukan simulasi jaringan SDN. Perangkat yang digunakan terbagi menjadi 2 komponen yaitu perangkat keras dan perangkat lunak. Perangkat keras merupakan komponen fisik dari sebuah perangkat *virtualisasi* jaringan SDN, perangkat keras yang digunakan yaitu laptop dengan prosesor Inter® Core™ i5-8265U CPU @ 1,60 GHz 1,80 GHz, RAM 8 GB dan memiliki sistem operasi Windows. Perangkat lunak merupakan sebuah program untuk berinteraksi antar *user* dengan perangkat keras, adanya perangkat lunak dapat membangun suatu simulasi jaringan SDN yang menerapkan Aruba VAN Controller. Perangkat lunak pada penelitian ini, yaitu VMware, Mininet, D-ITG, Iperf, Putty, Xming dan SDN Narmox Spear.

Topologi yang digunakan pada penelitian ini menggunakan topologi *custom*, namun pada topologi ini biasanya disebut dengan Topologi Abiline. Topologi Abilene merupakan topologi yang memiliki jaringan *backbone* berkemampuan tinggi (*high performance*), topologi ini direkomendasikan oleh Project Internet2 [11]. Topologi jaringan akan diilustrasikan pada Gambar 1, dimana topologi

tersebut menggunakan 7 buah switch, 7 buah host, 1 buah perangkat controller dan 1 buah host DHCP Server.



Gbr. 1 Topologi Abilene

Pada Gambar. 1 menampilkan topologi Abilene dengan masing-masing switch terhubung pada satu host. Perangkat Aruba VAN Controller berfungsi untuk mengontrol s1, s2, s3, s4, s5, s6 dan s7, dimana setiap switch terhubung dengan h1, h2, h3, h4, h5, h6 dan h7, s1 terhubung dengan s2, s3 dan s4, s4 terhubung dengan s6, s3 terhubung dengan s5, kemudian untuk s5, s6 dan s7 saling terhubung. Management server berperan sebagai Network Administrator yang sudah terintegrasi dengan Aruba VAN Controller, sehingga dapat melakukan tindakan untuk memprogram jaringan menggunakan aplikasi Narmox Spear. DHCP Server yang terhubung pada s1 berfungsi untuk memberikan alamat IP address secara otomatis kepada setiap host yang sudah saling terhubung dengan host lainnya.

Pada penelitian ini dilakukan pengujian koneksi jaringan yang menerapkan DHCP Server dengan perintah “*nx-pingall*” pada emulator mininet untuk mengetahui koneksi jaringan, selanjutnya melakukan pengujian performansi jaringan SDN. Penelitian ini menggunakan dua skenario untuk dilakukan pengujian performansi jaringan SDN, yaitu skenario pertama *bandwidth link* sebesar 100 Mbps dengan diberikan variasi *background traffic* sebesar 50 Mbps, 100 Mbps, 150 Mbps dan 200 Mbps. Skenario kedua *bandwidth link* sebesar 200 Mbps dengan diberikan variasi *background traffic* sebesar 50 Mbps, 100 Mbps, 150 Mbps dan 200 Mbps. Pengujian performansi dilakukan pengiriman paket menggunakan protokol UDP (*User Datagram Protokol*) dengan besar paket yang dikirim sebesar 1000 Bytes, jumlah paket yang dikirim perdetik sebanyak 100 packet/s, kemudian waktu pengiriman dilakukan selama 10 detik. Skenario pengujian pada penelitian ini dilakukan pengiriman paket sebanyak 30 kali pada masing-masing percobaan, kemudian secara bersamaan diberikan variasi *background traffic*.

TABEL I

ALOKASI BACKGROUND TRAFFIC TERHADAP BANDWIDTH LINK		
Skenario Ke-	Background Traffic	Bandwidth Link
1.	50 Mbps	100 Mbps
	100 Mbps	
	150 Mbps	
	200 Mbps	
2.	50 Mbps	200 Mbps
	100 Mbps	
	150 Mbps	
	200 Mbps	

Berdasarkan pada Tabel 1. dapat diketahui nilai variasi *background traffic* terhadap dua *bandwidth link* yang berbeda sebagai *variable* peubah untuk mendapatkan nilai parameter QoS pada skenario pengujian, sehingga kedua skenario pengujian tersebut dapat dilakukan analisis untuk mengetahui bagaimana kinerja sebuah jaringan SDN dengan *bandwidth link* yang berbeda berdasarkan nilai parameter QoS dari hasil data yang didapatkan pada tahap pengambilan data.

Pada tahap pengambilan data menggunakan perangkat lunak D-ITG yang dapat menghasilkan *log file* berisi nilai parameter QoS. Nilai parameter QoS tersebut yang akan diamati yaitu *delay*, *jitter*, *throughput* dan *packet loss*, kemudian dilakukan analisis dengan mengacu kepada standar TIPHON untuk mengetahui hasil analisis performansi jaringan SDN pada penelitian ini. Pengambilan data menggunakan alat bantu aplikasi Xming untuk memunculkan CLI *host* yang dapat berjalan bersamaan dengan aplikasi Putty yang berfungsi untuk meremote server melalui protokol SSH, sehingga dapat melakukan pengiriman paket data dari satu *node* ke *node* lainnya. Setelah kedua aplikasi tersebut dapat dijalankan, maka melakukan perintah “*xterm [nama host]*” pada CLI emulator mininet. Pada penelitian ini, penulis menjalankan aplikasi Iperf dan aplikasi D-ITG secara bersamaan pada masing-masing CLI *host* pengirim dan penerima. Iperf digunakan untuk membangkitkan *background traffic* dari h1 sebagai pengirim kepada h7 sebagai penerima, secara bersamaan dilakukan pengiriman paket data UDP menggunakan D-ITG.

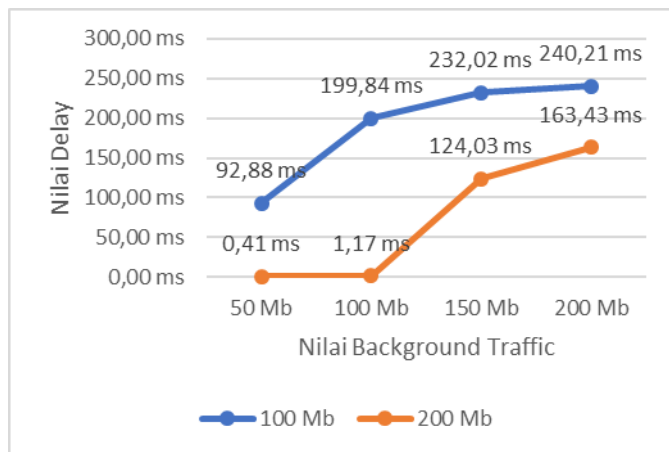
IV. HASIL DAN PEMBAHASAN

Pada penelitian ini telah dilakukan pengukuran performansi jaringan SDN, pengujian dilakukan dengan protokol UDP yang disalurkan dengan metode *client-side* yaitu *host* h1 pada penerima dan *server-side* yaitu *host* h7 pada pengirim, skenario pengujian telah ditentukan pada bab sebelumnya. Pengukuran performansi jaringan SDN telah dilakukan sebanyak 30 kali setiap skenario pengujian dengan jumlah variasi *background traffic* dan *bandwidth link* yang berbeda, hal ini digunakan sebagai *variable* pengubah, sehingga dapat diketahui hasil data perbandingannya. Acuan standar QoS yang digunakan adalah ETSI (*European Telecommunications Standard Institute*) TR 101 329 V2.1.1 (1999-06) tentang *Telecommunications and Internet Protocol Harmonization Over Network (TIPHON); General aspects of Quality of Service*.

A. Parameter Delay

Delay merupakan jeda waktu yang ditempuh, terhitung ketika paket dikirimkan oleh *transmitter* sampai kepada tujuan yang diterima oleh *receiver*. Besarnya nilai parameter

delay dipengaruhi oleh media transmisi dan ukuran paket yang akan dikirimkan. Semakin besar ukuran paket yang dikirimkan, maka semakin tinggi nilai parameter *delay* yang dihasilkan.



Gbr. 2 Grafik *delay* terhadap *Background Traffic*

Analisa berdasarkan grafik hasil pengujian parameter *delay* terhadap variasi nilai *background traffic* seperti pada Gambar 2, menunjukkan ketika *bandwidth link* sebesar 100 Mbps mengalami perubahan peningkatan nilai parameter *delay* pada jaringan SDN, karena pada saat nilai *background traffic* yang diberikan sama dengan besarnya nilai *bandwidth link* pada jaringan tersebut yaitu masing-masing sebesar 100 Mbps dengan hasil yang didapatkan nilai parameter *delay* sebesar 199,84 ms, ketika nilai *background traffic* melebihi batas nilai *bandwidth link*, maka nilai parameter *delay* terus meningkat seiring bertambah besarnya nilai *background traffic* yang diberikan. Nilai parameter *delay* tertinggi terdapat pada saat nilai *background traffic* sebesar 200 Mbps yaitu mencapai 240,21 ms, hal tersebut terjadinya *overload* yang dikarenakan adanya antrian panjang saat pengiriman paket atau pengambilan *route* lain untuk menghindari kemacetan.

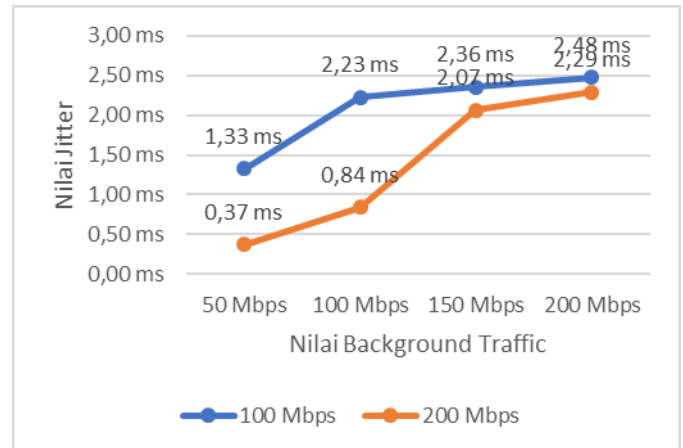
Pada grafik dengan nilai *bandwidth link* sebesar 200 Mbps mengalami perubahan nilai parameter *delay* pada saat diberikan nilai *background traffic* sebesar 150 Mbps dengan hasil nilai yang didapatkan parameter *delay* yaitu sebesar 124,03 ms, karena besarnya *background traffic* yang diberikan mendekati ambang batas nilai dari besarnya *bandwidth link* yaitu 200 Mbps.

Hasil pengujian performansi pada parameter *delay* dengan diberikan nilai variasi *background traffic* terhadap nilai *bandwidth link* yang digunakan membuktikan bahwa semakin besar beban trafik, maka semakin tinggi nilai parameter *delay* yang dihasilkan. Semakin besar *bandwidth link* yang digunakan, maka semakin kecil nilai parameter *delay* yang dihasilkan. Namun, secara keseluruhan dari hasil pengukuran pada parameter *delay* tidak terlalu signifikan melebihi nilai standar QoS TIPHON, yaitu masih dalam kategori baik dengan nilai indeks yaitu 3, oleh karena itu protokol UDP mengutamakan kecepatan pengiriman paket meskipun keutuhan paket tersebut diabaikan atau sama dengan nilai *packet loss* besar.

B. Parameter Jitter

Nilai parameter *jitter* didapatkan dari waktu panjang antrian dalam suatu jaringan, yaitu pada saat perbedaan

selang waktu pengiriman paket kepada sisi penerima. Hal tersebut dapat menimbulkan penundaan yang mempengaruhi *end-to-end delay* dan *jitter* dapat didefinisikan dengan variasi *delay*. Semakin besar beban trafik dalam sebuah jaringan akan menyebabkan semakin besar peluang terjadinya tumbukan antar paket dalam jaringan tersebut, hal ini menyebabkan nilai parameter *jitter* semakin besar. Pada protokol UDP tidak akan terjadi antrian yang cukup besar atau nilai parameter *jitter* tinggi, hal ini dikarenakan tidak ada pengurutan ulang paket – paket yang telah dikirimkan sampai di sisi penerima, apabila terjadi *overload* pada suatu jaringan.



Gbr. 3 Grafik *jitter* terhadap *Background Traffic*

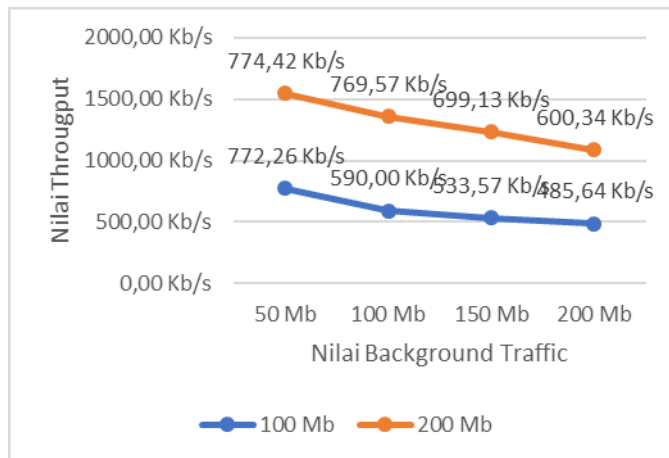
Analisa berdasarkan grafik hasil pengujian parameter *jitter* terhadap variasi nilai *background traffic* seperti pada Gambar 3, menunjukkan terjadinya kemacetan (*congestion*) yang cukup tinggi pada saat beban trafik sebesar 100 Mbps dan *bandwidth link* sebesar 100 Mbps dengan nilai parameter *jitter* yaitu sebesar 2,23 ms, hal tersebut terjadi *overload* didalam jaringan tersebut, sehingga antrian pengiriman paket pada suatu jaringan cukup besar. Sedangkan pada saat beban trafik sebesar 100 Mbps dan *bandwidth link* sebesar 200 Mbps, dihasilkan nilai parameter *jitter* yaitu sebesar 0,84 %, hal tersebut terjadi karena nilai *bandwidth link* lebih besar dibandingkan beban trafik pada jaringan tersebut.

Berdasarkan hasil nilai parameter *jitter*, semakin besar beban trafik maka semakin tinggi nilai parameter *jitter*, namun jika sudah melebihi kapasitas *bandwidth link* yang sudah di tentukan, maka perubahan nilai parameter *jitter* tidak terlalu signifikan, oleh karena itu pada protokol UDP apabila terjadi *overload* pada suatu jaringan, maka antrian dari pengiriman suatu paket akan diabaikan dan akan terjadi pengiriman paket yang tidak sempurna urutannya di sisi penerima, karena adanya paket yang hilang. Nilai parameter *jitter* tertinggi pada saat beban trafik sebesar 200 Mbps dan *bandwidth link* sebesar 100 Mbps dihasilkan nilai parameter *jitter* yaitu sebesar 2,48 ms, sedangkan untuk nilai parameter *jitter* terendah pada saat beban trafik sebesar 50 Mbps dan *bandwidth link* sebesar 200 Mbps dihasilkan nilai parameter *jitter* yaitu sebesar 0,37 ms, namun dari keseluruhan nilai parameter *jitter* yang dihasilkan masih berkisaran < 75 ms dengan nilai indeks, yaitu 4 dengan kategori (*perfect*) sesuai dengan standarisasi QoS TIPHON.

C. Parameter Throughput

Nilai parameter *throughput* merupakan jumlah bit yang diterima dengan sukses perdetik melalui sebuah sistem

jaringan telekomunikasi atau kemampuan sebenarnya dalam suatu jaringan pada saat melakukan pengiriman paket. Aspek utama *throughput* yaitu berkisar pada ketersediaan *bandwidth* untuk menjalankan suatu aplikasi dengan baik, hal ini menentukan besarnya trafik yang dihasilkan pada saat melewati suatu jaringan.



Gbr. 4 Grafik throughput terhadap Background Traffic

Pada parameter *throughput*, jumlah paket yang dikirimkan perdetik (pps) dan besar paket yang dikirimkan (Bytes) serta nilai *bandwidth link* pada suatu jaringan dapat mempengaruhi nilai parameter *throughput* dan *packet loss*. Pada bab sebelumnya sudah menentukan jumlah paket dan besar paket yang dikirimkan untuk dilakukan pengujian serta nilai *bandwidth link* yang sudah ditentukan. Analisa berdasarkan grafik hasil pengujian parameter *throughput* terhadap variasi nilai *background traffic* seperti pada Gambar 4, pada saat nilai *bandwidth link* sebesar 200 Mbps dan beban trafik sebesar 50 Mbps dihasilkan nilai parameter *throughput* tertinggi yaitu sebesar 774,42 Kb/s, sedangkan nilai parameter *throughput* terendah yaitu sebesar 485,64 Kb/s. Nilai parameter *throughput* terendah disebabkan karena melebihi batas nilai *bandwidth link*, yaitu ketika beban trafik sebesar 200 Mbps dan nilai *bandwidth link* hanya sebesar 100 Mbps.

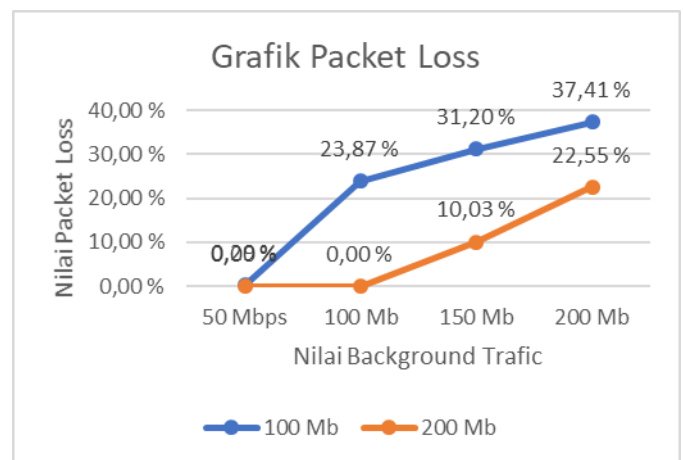
Secara keseluruhan dari nilai parameter *throughput* yang dihasilkan, seiring bertambahnya beban trafik yang diberikan mengalami penurunan terhadap nilai parameter *throughput*. Penurunan nilai parameter *throughput* lebih besar pada saat nilai *bandwidth link* sebesar 100 Mbps dibandingkan dengan nilai *bandwidth link* sebesar 200 Mbps. Namun, pada saat beban trafik sebesar 50 Mbps terdapat kesamaan nilai parameter *throughput* dengan selisih nilai parameter *throughput* yaitu sebesar 2.16 Kbps, hal itu dikarenakan beban trafik dalam suatu jaringan masih belum melewati batas maksimal pada nilai *bandwidth link* yang sudah ditentukan, yaitu sebesar 100 Mbps dan 200 Mbps.

Hasil pengujian performansi pada parameter *throughput* dengan diberikan nilai variasi *background traffic* terhadap nilai *bandwidth link* yang digunakan membuktikan bahwa semakin besar nilai *bandwidth link* yang ditentukan dalam suatu jaringan, maka semakin besar nilai parameter *throughput* yang dihasilkan. Semakin besar beban trafik dalam suatu jaringan, maka semakin besar nilai *bandwidth link* yang dibutuhkan. Hal tersebut terjadi karena beban trafik mempengaruhi nilai parameter *throughput* yang dihasilkan,

tetapi pada saat pengiriman paket menggunakan protokol UDP dalam suatu jaringan yang memiliki batas nilai *bandwidth link* dengan beban trafik yang semakin besar, maka kepadatan trafik meningkat yang akan berdampak kepada meningkatnya potensi *packet loss* dalam suatu jaringan, sehingga menyebabkan nilai parameter *throughput* yang dihasilkan akan semakin kecil disisi *destination*.

D. Parameter Packet Loss

Nilai parameter *packet loss* merupakan banyaknya paket yang hilang dalam satuan persentase selama proses pengiriman paket sampai kepada tujuan, misalnya pada penelitian ini terjadi adanya paket bertabrakan (*collision*) atau kemacetan trafik pada saat pengiriman paket (*congestion*) dalam suatu jaringan, hal tersebut terjadi disebabkan ketika beban trafik tidak dapat memenuhi kapasitas dari nilai *bandwidth link* yang tersedia. Selain itu akan mengakibatkan suatu paket tersebut rusak (*corrupt*), sehingga tidak sampai pengiriman paket kepada penerima.



Gbr. 5 Grafik packet loss terhadap Background Traffic

Analisa berdasarkan grafik hasil pengujian parameter *packet loss* terhadap variasi nilai *background traffic* seperti pada Gambar 5, ketika beban trafik sebesar 100 Mbps dan *bandwidth link* sebesar 100 Mbps, menunjukkan adanya tabrakan (*collision*) paket yang dikirimkan pada suatu jaringan, secara signifikan nilai parameter *packet loss* tinggi yaitu sebesar 23,87 %, hal tersebut terjadi *overload* didalam jaringan tersebut, sehingga banyak paket yang rusak (*corrupt*) dan paket tidak utuh pada sisi penerima. Sedangkan pada saat beban trafik sebesar 100 Mbps dan *bandwidth link* sebesar 200 Mbps, mendapatkan nilai parameter *packet loss* yaitu sebesar 0,00 %, hal tersebut terjadi karena nilai *bandwidth link* lebih besar dibandingkan beban trafik pada suatu jaringan tersebut.

Berdasarkan hasil nilai parameter *packet loss* yang didapatkan, seiring dengan bertambahnya beban trafik pada suatu jaringan, maka nilai parameter *packet loss* semakin meningkat. Nilai parameter *packet loss* tertinggi yaitu sebesar 37,41 %, pada saat beban trafik sebesar 200 Mbps dan nilai *bandwidth link* sebesar 100 Mbps. Nilai parameter *packet loss* terendah yaitu sebesar 0,00 %, pada saat beban trafik sebesar 50 Mbps dan nilai *bandwidth link* sebesar 200 Mbps.

Hasil pengujian performansi pada parameter *packet loss* dengan diberikan nilai variasi *background traffic* terhadap nilai *bandwidth link* yang digunakan membuktikan bahwa

semakin besar beban trafik, maka semakin tinggi nilai parameter *packet loss* yang didapatkan. Semakin besar *bandwidth link* yang digunakan, maka semakin kecil nilai parameter *packet loss* yang didapatkan. Secara keseluruhan nilai parameter *packet loss* dikategorikan jelek dengan nilai indeks yaitu 4 (*poor*), apabila beban trafik melebihi kapasitas nilai *bandwidth link* yang sudah ditentukan. Oleh karena itu pengiriman paket menggunakan protokol UDP membutuhkan nilai *bandwidth link* yang besar.

V. KESIMPULAN

Kesimpulan penelitian ini dari hasil pengujian performansi jaringan SDN dengan pengiriman paket menggunakan protokol UDP, untuk besar paket yang dikirimkan yaitu sebesar 1000 Bytes dan untuk jumlah paket yang dikirim perdetik sebanyak 100 paket/s. Hasil pengujian didapatkan ketika beban trafik melebihi kapasitas nilai *bandwidth link*, maka kepadatan trafik meningkat yang akan berdampak kepada meningkatnya potensi *packet loss* dalam suatu jaringan, sehingga menyebabkan nilai parameter *throughput* yang dihasilkan akan semakin kecil disisi *destination*, yaitu sebesar 485,64 Kb/s. Semakin kecil nilai parameter *throughput*, maka kecepatan pengiriman paket semakin menurun, sehingga dapat mempengaruhi nilai parameter *delay* yang semakin meningkat, yaitu sebesar 240,21 ms. Jika nilai parameter *delay* meningkat, maka memperpanjang antrian pada pengiriman paket tersebut yang akan mengakibatkan meningkatnya nilai parameter *jitter*, yaitu sebesar 2,48 ms. Ketika terjadi *overload* dalam suatu jaringan tersebut, maka akan ada *packet loss* yang disebabkan oleh paket yang bertabrakan (*collision*), sehingga paket tersebut rusak (*corrupt*) dan mengakibatkan paket tidak utuh di sisi penerima. Nilai parameter *packet loss* tertinggi, yaitu sebesar 37,41 %. Namun, ketika beban trafik meningkat dan tidak melebihi kapasitas *bandwidth link* yang sudah ditentukan, maka ketika *bandwidth link* sebesar 200 Mbps dihasilkan nilai parameter *throughput* tertinggi yaitu sebesar 774,42 Kb/s, sehingga nilai parameter *delay* ikut menurun dengan nilai terendah yaitu sebesar 0,41 ms, kemudian untuk nilai parameter *jitter* terendah yaitu sebesar 0,37 ms dan untuk nilai parameter *packet loss* terendah, yaitu sebesar 0,00 %.

Penulis memberikan saran atas kekurangan dan keterbatasan penelitian ini, yaitu dilakukan pengujian pengiriman paket menggunakan protokol TCP dengan *bandwidth link* yang *unlimited* dan membandingkan ketika adanya penambahan *switch* dan *host* dalam suatu jaringan SDN, kemudian menambahkan *flow tabel* yang dapat dilakukan pada aplikasi Narmox Spear untuk pemilihan *route* pengiriman paket. Selain itu dilakukan DHCP *Snooping*, agar tidak terjadinya DHCP *server illegal* yang dapat memberikan IP address kepada *client*.

DAFTAR PUSTAKA

- [1] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 1955–1980, 2014.
- [2] J. Kurbalija, *Sebuah Pengantar Tentang Tata Kelola Internet*, Edisi Ke I., vol. 1. Switzerland: APJII-www.apjii.or.id, 2011.
- [3] A. . Fallis, *Software Defined Networks A Comprehensive Approach*, vol. 53, no. 9. 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann, 2013.

- [4] C. S. Li and W. Liao, "Software defined networks," *IEEE Communications Magazine*, vol. 51, no. 2, p. 113, 2013.
- [5] M. A. Aulia and R. Tulloh, "Simulation of Software Defined Network Using OSPF Protocol and POX as Controller," *Open Libr. Telkom Univ.*, 2018.
- [6] HPE, "Aruba VAN SDN Controller Software," 1st ed., vol. 1, no. March, America: Hewlett Packard Enterprise Company, 2017, p. 8.
- [7] M. H. Hidayat and N. R. Rosyid, "Analisis Kinerja dan Karakteristik Arsitektur Software-Defined Network Berbasis OpenDaylight Controller," *Citee*, no. 2085–6350, pp. 194–200, 2017.
- [8] A. M. Al-Sadi, A. Al-Sherbaz, and J. Xue, "Developing an Asynchronous Technique to Evaluate the Performance of SDN HP Aruba Switch and OVS," *Intell. Comput.*, vol. 1, pp. 251–264, 2019.
- [9] R. Afan, I. A. Virgono, and R. Rumani, "Analisis Efek Penggunaan Kontroler Ryu Dan POX pada Performansi Jaringan SDN," vol. 5, no. 14, pp. 68–82, 2018.
- [10] I. P. A. E. Pratama and I. M. A. Wikantyasa, "Implementasi dan Analisis Simulasi QOS dan Performance Device dengan Menggunakan ONOS dan Iperf3," *J. Inform. Univ. Pamulang*, vol. 4, no. 2, p. 57, 2019.
- [11] F. Ramadhan, R. Pramananda, and W. Yahya, "Implementasi Routing Berbasis Algoritme Dijkstra Pada Software Defined Networking Menggunakan Kontroler Open Network Operating System," *J. Pengemb. Teknol. Inf. dan Ilmu Komputer*; Vol 2 No 7, vol. 2, no. 7, pp. 2531–2541, 2017.