



Pengoptimasian *Traffic* pada Jaringan *Wide Area Network* Menggunakan *Application Aware Routing* Berbasis SD-WAN

Ahmad Firdausi¹, Deni Ramdani^{2*}

¹*Teknik Elektro, Universitas Mercu Buana,
Jl. Meruya Selatan, Jakarta 11650, Indonesia*

²*PT NTT Indonesia Solutions,*

Jl. Prof. DR. Satrio, Jakarta 12940, Indonesia

*Email Penulis Koresponden: deni.ramdani@outlook.com

Abstract:

Seiring dengan berkembangnya zaman, pengguna internet saat ini terus bertambah. Pengguna internet dunia yang diprediksi akan mencapai 66% populasi dunia pada tahun 2023, perkembangan tersebut juga didukung oleh munculnya berbagai jenis aplikasi dan komputasi berbasis awan. Tentunya dengan seiring bertambahnya pengguna internet dan penggunaan berbagai aplikasi tersebut diperlukan kehandalan jaringan WAN (*Wide Area Network*) dalam menangani *traffic*. Saat ini masih terdapat beberapa tantangan dalam menangani *traffic* diantaranya terbatasnya bandwidth, latensi tinggi, kemacetan data dan hilangnya paket data pada jaringan. Oleh karena itu penulis mengusulkan implementasi *application aware routing* berbasis SD-WAN yang diharapkan dapat menjadi solusi ketika terdapat gangguan pada jaringan WAN yang dapat menurunkan performansi *traffic* data dan suara sehingga memberikan pengalaman akses yang lebih baik. Berdasarkan pengukuran dan simulasi sistem yang telah diimplementasikan pada simulator jaringan GNS3 terhadap performansi *traffic* suara dan data, performansi *Quality of Service traffic* suara saat *peak traffic* menggunakan SD-WAN *application aware routing* lebih baik dibandingkan dengan jaringan *Border Gateway Protocol Multihomed* dengan rata-rata *latency* 20,77 ms, rata-rata *jitter* 27,26 ms dan *packet loss* 3,7%. Performansi *Quality of Service traffic* suara tersebut masih dalam kategori “Baik” dan “Sangat Baik” berdasarkan standar TIPHON. Sedangkan performansi *traffic* data saat *peak traffic* menggunakan SD-WAN *application aware routing* juga lebih baik dibandingkan dengan *Border Gateway Protocol Multihomed* dengan rata-rata waktu tanggap 146.812 ms, dan TCP *retransmission* 15 paket.

This is an open access article under the [CC BY-NC](https://creativecommons.org/licenses/by-nc/4.0/) license



Kata Kunci:

SD-WAN;
Routing;
VOIP;
Performansi;

Riwayat Artikel:

Diserahkan 24 Maret 2022
Direvisi 14 Juli 2022
Diterima 25 Juli 2022
Dipublikasi 18 Agustus 2022

DOI:

10.22441/incomtech.v12i2.15180

1. PENDAHULUAN

Teknologi jaringan WAN yang menggunakan teknik *circuit switching* yang telah ada diantaranya adalah *Integrated Service Digital Network* (ISDN). Sedangkan untuk *packet switching* menggunakan teknologi frame relay, *Asynchronous Transfer Mode* (ATM) dan IP MPLS (*Multiprotocol Label Switching*) [1]. Untuk menangani lalu lintas data melalui jaringan WAN melalui keempat teknologi tersebut terdapat beberapa hambatan diantaranya terbatasnya *bandwidth*, latensi tinggi, *congestion* dan *packet loss* yang tinggi [2]. Selain itu, permasalahan yang dihadapi dengan jaringan MPLS (*Multiprotocol Label Switching*) saat ini hanya digunakan untuk *traffic* secara lokal, sedangkan tantangan bisnis saat ini memerlukan akses yang *reliable* ke *cloud* [3].

Dengan mengimplementasikan SD-WAN (*Software Defined - Wide Area Network*), perusahaan dapat menentukan jenis koneksi WAN yang akan digunakan secara fleksibel disesuaikan dengan keperluan baik secara lokal maupun *cloud* dan berbiaya murah [1]. Namun seiring dengan fleksibilitas pemilihan jenis koneksi, koneksi WAN seperti publik *direct internet access* tidak terjamin dari gangguan [4]. Oleh karena itu diperlukan metode untuk manipulasi pemilihan jalur *forwarding data* apabila koneksi WAN tidak normal. Untuk menghadapi tantangan tersebut, berbagai metode manipulasi *traffic* telah dikembangkan diantaranya yaitu dengan menggunakan *policy-based routing* [5], *traffic engineering* [6], maupun *bandwidth management* [7].

Mengimplementasikan *policy-based routing* dapat melakukan *failover* namun terjadi ketika salah satu jalur *down*, namun tidak dapat mendeteksi kualitas QoS untuk melakukan *failover* [5]. Begitu pula dengan teknik manajemen *bandwidth* yang dapat mengoptimalkan penggunaan *bandwidth* namun tidak mendeteksi gangguan pada jaringan [7]. Sedangkan untuk mengimplementasikan *traffic engineering* teknik algoritma yang kompleks sehingga memerlukan sumber daya jaringan yang cukup besar [8]. Oleh karena itu, penulis mengusulkan metode dengan mengimplementasikan *application-aware-routing* pada jaringan SD-WAN dengan melakukan pemantauan QoS berupa *latency*, *jitter* dan *loss* sebagai penentu perutean *traffic* secara dinamis yang dapat diimplementasikan berdasarkan aplikasi yang dilewatkan pada jaringan dengan algoritma dan protokol yang tidak memerlukan sumber daya yang besar.

2. METODE

2.1 Studi Literatur

Telah ada beberapa penelitian yang telah dilakukan sebelumnya yang berkaitan dengan penelitian ini. Penelitian pertama oleh Van Adrichem menggunakan perangkat lunak lisensi terbuka yang dinamai OpenNetMon untuk memantau penggunaan secara aktual dari *bandwidth*, *delay* dan *packet loss* pada jaringan SDN (*Software Defined Network*) berbasis OpenFlow [9]. Penelitian tersebut hanya melakukan pemantauan saja terhadap parameter QoS namun belum melakukan metode *failover* berdasarkan hasil pemantauan QoS. Penelitian tersebut menjadi dasar dalam mengimplementasikan rekayasa *traffic*.

Penelitian kedua yang dilakukan oleh U-chupala mengimplementasikan perutean *traffic data* berdasarkan *bandwidth* dan latensi pada jaringan SDN berbasis *open flow* serta membandingkan performansi *bandwidth* dan *latency* dengan *routing* tradisional [10]. Penelitian tersebut sudah mampu memilih jalur perutean berdasarkan *bandwidth* dan latensinya, namun dengan dua parameter tersebut masih belum mempertimbangkan QoS aksesnya.

Penelitian ketiga oleh Pambudi dan Aziz yang melakukan manipulasi *traffic* dengan metode *policy-based routing* untuk membagi jalur akses internet WAN menggunakan perangkat Mikrotik dan dua jalur ISP (*Internet Service Provider*) [5]. Penelitian tersebut hanya membagi jalur akses internet melalui dua jalur dan akan *failover* apabila salah satu jalur *down* saja tanpa mempertimbangkan faktor QoS nya, meskipun QoS pada jalur ISP jelek maka tidak akan terjadi *failover* sehingga pengguna akan terkena dampak dari jeleknya QoS tersebut.

Penelitian keempat oleh Ramadhan yaitu perancangan dan analisis parameter QoS pada *traffic VOIP* (*Voice over IP*) menggunakan *routing* protokol BGP (*Border Gateway Protocol*) [11]. Penelitian tersebut tidak mengusulkan metode *failover* jalur berdasarkan hasil QoS-nya.

Penelitian kelima oleh Badrul dan Akmaludin mengimplementasikan manajemen *bandwidth* dengan HTB (*Hierarchical Token Bucket*) dengan *queue* yang berhirarki pada perangkat Mikrotik [7]. Manajemen *bandwidth* memberikan pengaruh pada user saat terjadinya *traffic* tinggi yang dapat meminimalisir terjadinya kemacetan *traffic* namun tidak mendeteksi gangguan pada jaringan atau penurunan QoS sehingga pengguna akan terkena dampak dari jeleknya QoS.

Penelitian keenam oleh Satriawan menganalisis performansi *Quality of Service* meliputi *delay*, *jitter* dan *throughput* dari jaringan MPLS (*Multi Protocol Label Switching*) dan jaringan SD-WAN (*Software Defined Wide Area Network*) dalam aplikasi layanan VOIP (*Voice over Internet Protocol*) [12]. Penelitian tersebut menunjukkan SD-WAN memiliki QoS jaringan suara yang lebih baik daripada jaringan MPLS, namun belum menguji dampak jaringan SD-WAN terhadap komunikasi *traffic data*.

Penelitian ketujuh oleh Shaikh membandingkan teknologi *high availability* dan konvergensi dalam teknologi jaringan IP (*Internet Protocol*) [13]. Dalam penelitian tersebut, terdapat protokol BFD (*Bidirectional Forwarding Detection*) memiliki waktu kurang dari 1 detik sehingga protokol BFD sangat handal untuk digunakan sebagai protokol yang mendukung *high availability* dan pemantauan jaringan.

Penelitian kedelapan oleh Lin mengusulkan pendekatan *path label routing* dalam WAN berbasis SDN dan menggabungkannya dengan *routing* konvensional menggunakan OpenFlow [6]. Penelitian kesembilan oleh Zhang & Zhao membuat jaringan SDN dengan *QoS-aware routing* yang dinamai dengan LSEA untuk *event Live-Soccer video*. Dari segi *routing* mengimplementasikan *improved Dijkstra routing algorithm* and *SDN-based disjoint routing algorithm*. Penelitian kedelapan dan kesembilan dapat memperbaiki parameter QoS, namun memerlukan sumber daya yang besar dan kompleks dalam mengimplementasikannya.

Berdasarkan beberapa penelitian-penelitian tersebut, maka penulis mengusulkan metode untuk meminimalisir dampak adanya gangguan pada jaringan WAN dengan mengimplementasikan *application-aware-routing* pada jaringan SD-WAN dengan melakukan pemantauan QoS berupa *latency*, *jitter* dan *loss* sebagai penentu perutean *traffic* secara dinamis yang dapat diimplementasikan berdasarkan aplikasi

yang dilewatkan pada jaringan dengan algoritma dan protokol BFD (*Bidirectional Forwarding Detection*) yang tidak memerlukan sumber daya yang besar.

2.2 Perangkat Lunak dan Perangkat Keras

Pada penelitian ini terdapat beberapa perangkat lunak dan perangkat keras yang digunakan ditunjukkan pada [Tabel 1](#) dan [Tabel 2](#).

Tabel 1 Perangkat Lunak

No	Nama Perangkat Lunak	Versi	Fungsi
1	GNS3	2.2.22	Emulator Jaringan
2	VMware ESXi	6.7.0	Hypervisor
3	Wireshark	3.6.5	Packet Capture Tool
4	NETem	6.4	Bandwidth Emulator
5	Startrinity SIP Tester	2021-06-24	Softphone Tester
6	Zoiper Softphone	5.4.12	Softphone
7	iPerf	2.0.13	Traffic Generator
8	HTTTPing	2.4	HTTP Traffic Generator
9	Viptela OS	19.2.3	SD-WAN Device OS
10	Cisco IOSv	15.5(2)T	Router & Switch OS
11	Ubuntu OS	18.04	Server & Client OS

Tabel 2 Perangkat Keras

No	Nama Perangkat Keras	Spesifikasi
1	Server	8 CPU, 24 GB Memory, 500 GB Storage
2	Laptop Client	Intel(R) Core i5 2.4 GHz, 16 GB RAM

2.3 Parameter Penelitian

Quality of Service (QoS) adalah suatu cara mengatur kualitas terhadap layanan tertentu dalam lalu lintas data untuk mencapai tingkat kualitas layanan tertentu. Terdapat beberapa parameter yang sering digunakan sebagai indikator tingkat kualitas jaringan diantaranya yaitu *delay*, *jitter* dan *packet loss*.

Delay (*Latency*) merupakan lamanya waktu yang dibutuhkan data untuk berpindah dari alamat IP asal ke alamat IP tujuan [14]. *Delay/Latency* didapatkan melalui persamaan berikut:

$$\text{Delay (s)} = \frac{\text{Packet Length (bit)}}{\text{Link Bandwidth (bit/s)}}$$

Menurut standar TIPHON [15], *delay* dikategorikan seperti pada [Tabel 3](#) berikut.

Tabel 3 Kategori *Delay* Standar TIPHON

Kategori <i>Delay</i>	Besar <i>Delay</i> (ms)
Sangat Bagus	< 150 ms
Bagus	150 ms s/d 300 ms
Sedang	300 ms s/d 450 ms
Jelek	300 ms s/d 450 ms

Jitter adalah variasi dari *delay* atau selisih antara *delay* pertama dengan *delay* selanjutnya [16]. *Jitter* didapatkan melalui persamaan berikut:

$$Jitter (s) = \frac{Total\ Delay\ Variation (s)}{Total\ Received\ Packet}$$

Menurut standar TIPHON [15], *jitter* dikategorikan seperti pada Tabel 4 berikut.

Tabel 4 Kategori *Jitter* Standar TIPHON

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

Packet loss adalah suatu parameter QoS yang menunjukkan presentase total paket yang hilang selama proses mentransmisikan data [16]. *Packet loss* dapat terjadi karena *collision* atau *congestion* pada jaringan [14]. *Packet loss* didapatkan melalui persamaan berikut:

$$Packet\ Loss\ (\%) = \frac{(Packet\ Sent - Packet\ Received)}{Packet\ Sent} \times 100\%$$

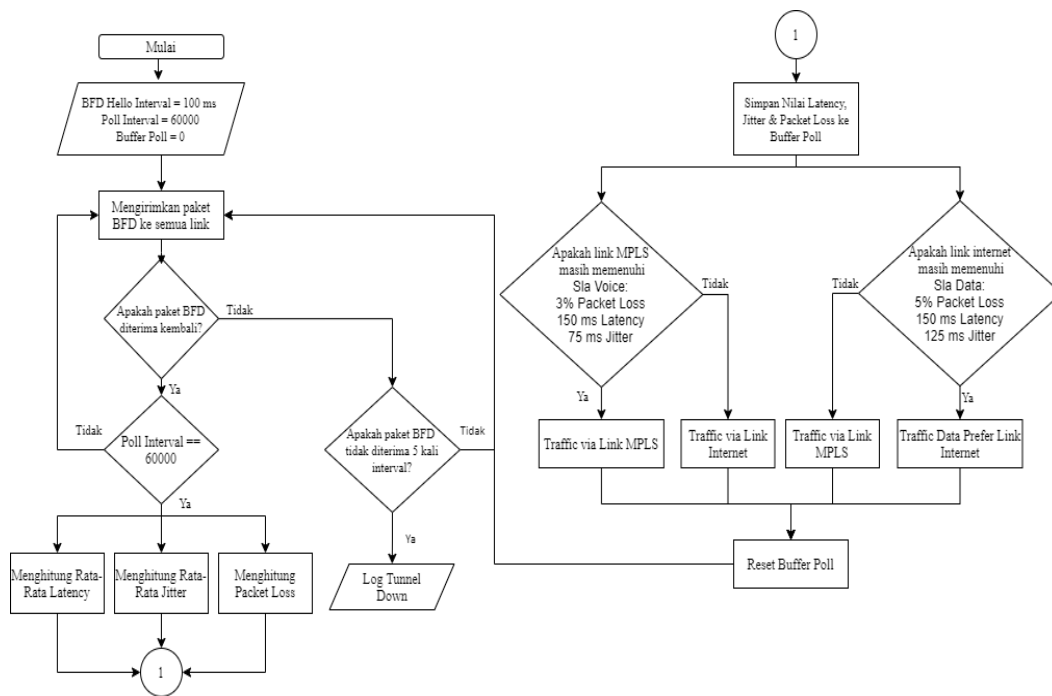
Menurut standar TIPHON [15], *packet loss* dikategorikan seperti pada Tabel 5 berikut.

Tabel 5 Kategori *packet loss* standar TIPHON

Kategori <i>Packet Loss</i>	Besar <i>Packet Loss</i> (%)
Sangat Bagus	0-2
Bagus	3-14
Sedang	15-24
Jelek	> 25

2.4 Perancangan *Policy Application Aware Routing*

Application aware routing dirancang untuk melakukan perpindahan *link* apabila salah satu QoS *link* tidak memenuhi nilai *threshold*. Terdapat 2 *link* yang diimplementasikan yaitu *link* MPLS dan *link* Internet. Setiap *link* memiliki nilai *threshold* yang berbeda untuk tiap parameter *latency*, *jitter* dan *packet loss*. Nilai parameter QoS tersebut dipantau secara terus-menerus menggunakan protokol BFD yang diimplementasikan pada perangkat SD-WAN. Proses *application aware routing* ditunjukkan pada Gambar 1.

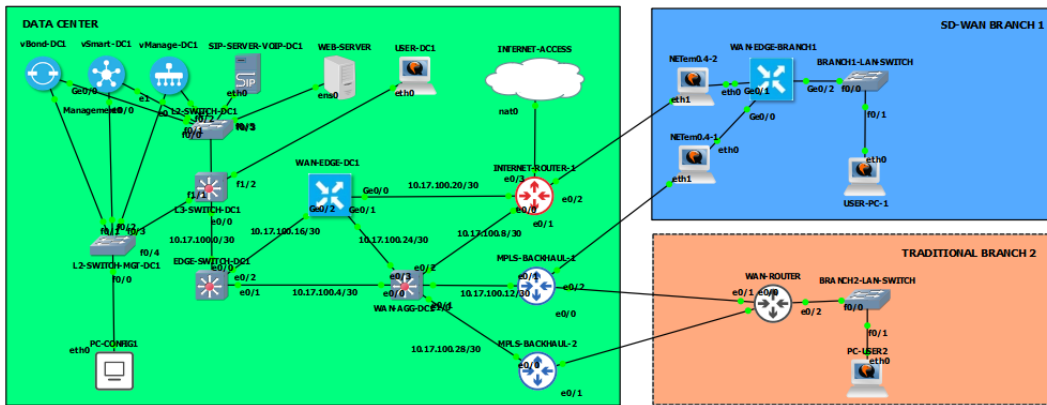


Gambar 1 Diagram Alir *Application Aware Routing*

Protokol BFD (*Bidirectional Forwarding Detection*) akan terus mengukur kualitas *link* dengan mengirimkan *hello interval* setiap 100 ms. Apabila salah satu *peer* tidak mengirimkan 5x *hello* maka *link* akan dinotifikasi *Down*. Setelah interval 60000 ms, parameter QoS akan dihitung rata-ratanya dan dibandingkan dengan nilai *threshold* masing-masing *traffic*. Nilai *threshold traffic voice* yaitu 3% *packet loss*, 150 ms *latency* dan 75 ms *jitter*. Sedangkan untuk nilai *threshold traffic data* yaitu 5% *packet loss*, 150 ms *latency* dan 125 ms *jitter*. *Application aware routing policy* ini nantinya diimplementasikan pada SD-WAN *controller vSmart*. Setelah *policy* tersebut aktif maka seluruh *forwarding traffic* pada jaringan SD-WAN akan diproses berdasarkan *policy* tersebut.

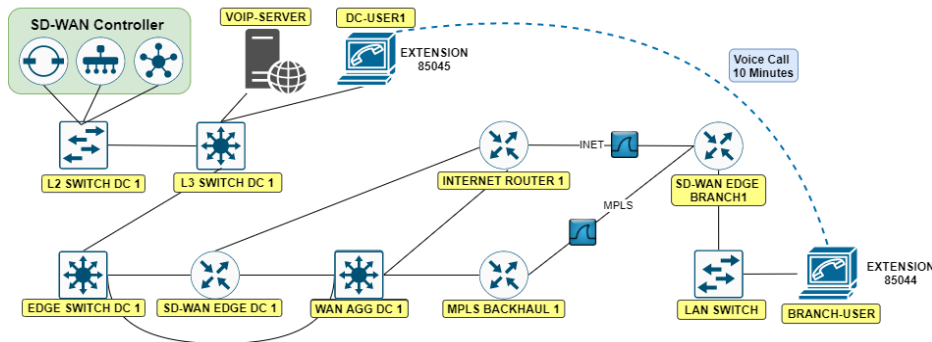
2.5 Skenario Pengujian

Skenario pengujian terhadap sistem yang telah dirancang pada Gambar 2. Terdapat 3 skenario pengujian yang akan dilakukan yaitu menguji manipulasi *traffic*, menguji parameter QoS *traffic voice* dan menguji performansi *traffic* HTTP. Pengujian performansi *traffic* data dan *voice* dilakukan pada *site Branch* SD-WAN yang telah mengimplementasikan *application aware routing* dan *site Branch* Non-SD-WAN yang menggunakan BGP *Multihomed*.



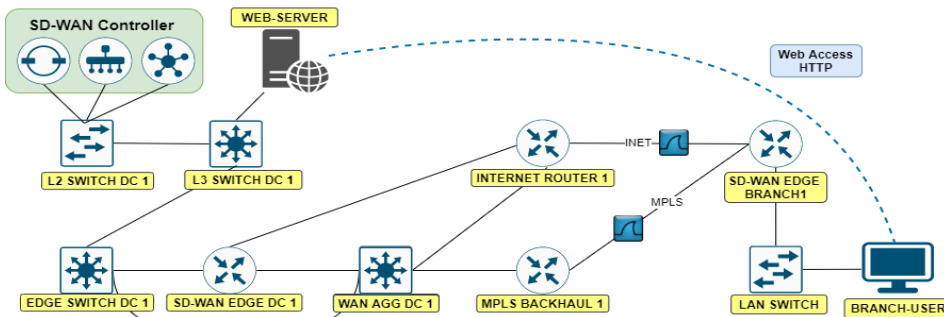
Gambar 2 Topologi Jaringan Keseluruhan

Skenario pengujian manipulasi *traffic voice* ditunjukkan oleh Gambar 3 dengan melakukan panggilan dari Branch-User nomor ekstensi 85044 ke DC-User1 nomor ekstensi 85045 selama 10 menit dalam keadaan *traffic* pada kedua link internet dan MPLS normal. Keluaran dari pengujian ini untuk menguji manipulasi *traffic* yang dirancang yaitu *traffic voice* harus menggunakan link MPLS ketika kualitas link MPLS masih dibawah nilai *threshold*.



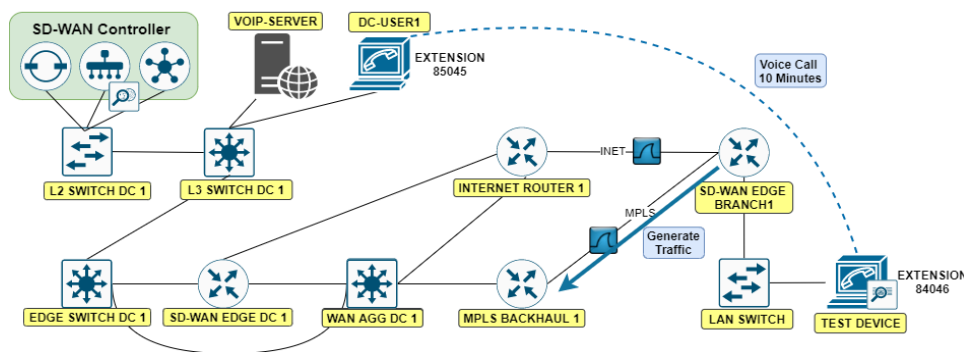
Gambar 3 Skenario Pengujian Manipulasi Traffic Voice

Skenario pengujian manipulasi *traffic data* ditunjukkan pada Gambar 4 dengan melakukan akses ke *web server* dengan alamat <http://172.17.20.31/> dari Branch-User dalam keadaan *traffic* pada kedua link internet dan MPLS normal. Keluaran dari pengujian ini untuk menguji manipulasi *traffic* yang dirancang yaitu *traffic data* protokol HTTP harus menggunakan link internet ketika kualitas link internet masih dibawah nilai *threshold*.



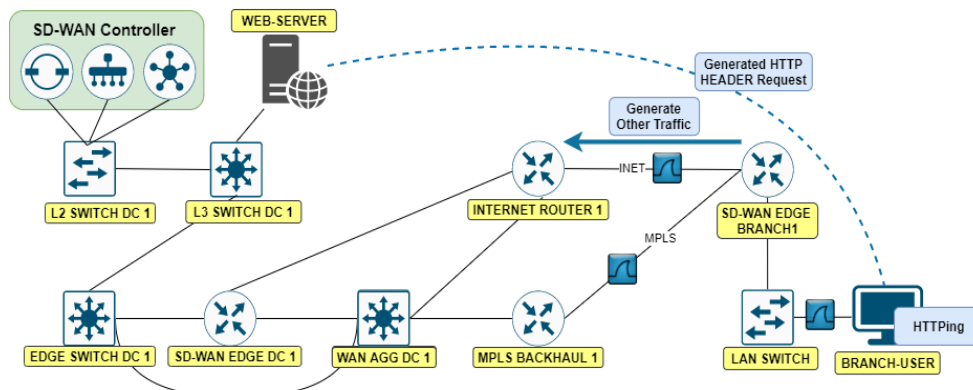
Gambar 4 Skenario Pengujian Manipulasi Traffic Data

Skenario pengujian untuk menguji QoS *traffic voice* adalah dengan melakukan panggilan dengan durasi 10 menit dari nomor ekstensi DC-User1 85045 ke nomor ekstensi Tester 84046 yang telah terinstall aplikasi Startrinity SIP Tester seperti pada [Gambar 5](#). Setelah durasi panggilan 5 menit, membangkitkan *background traffic* UDP melalui aplikasi iperf dari perangkat SD-WAN Edge. *Branch 1* ke IP address link MPLS-Backhaul 1 192.168.10.5 dengan durasi 3 menit. Persentasi *traffic* yang dikirimkan melalui iperf pada saat pengujian dibangkitkan secara bertahap dari 10% sampai dengan 90% *total bandwidth* yaitu 5 Mbps sehingga terdapat 9 kali pengujian. Disaat yang bersamaan melakukan pengukuran pada kedua *link* dengan menggunakan perangkat lunak Wireshark saat panggilan dilakukan dan mengukur kualitas QoS (*delay, jitter, packet loss*) secara *end-to-end* pada koneksi SIP *server*.



Gambar 5 Skenario Pengujian Performansi *Traffic Voice*

Metode pengujian yang dilakukan untuk menguji performansi *traffic data* yaitu melakukan HTTP *Header request* dari PC *Branch-User* dengan menggunakan perangkat lunak HTTPing yang dijalankan dengan interval 1 detik dengan total koneksi 300 *request* seperti pada [Gambar 6](#). Setelah 1 menit HTTPing berjalan membangkitkan *background traffic* UDP melalui aplikasi iperf dari perangkat SD-WAN Edge ke *link* Internet. Persentasi *traffic* yang dikirimkan melalui kedua *link* dibangkitkan secara bertahap dari 10% sampai dengan 90% *total bandwidth*. Disaat yang bersamaan melakukan *packet capture* pada kedua *link* WAN dan *interface* ke PC *Branch-User* dengan menggunakan perangkat lunak Wireshark.



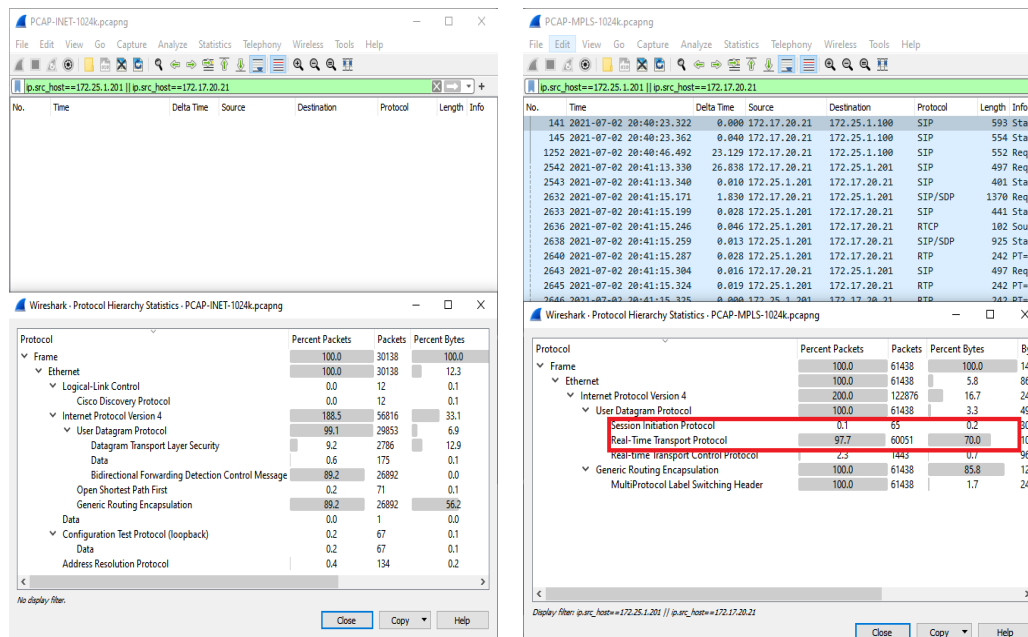
Gambar 6 Skenario Pengujian Performansi *Traffic Data*

3. HASIL DAN PEMBAHASAN

Hasil pengujian terbagi menjadi 3 bagian yaitu hasil pengujian manipulasi *traffic*, parameter QoS *traffic voice* dan performansi *traffic data*.

3.1 Hasil Pengujian Manipulasi Traffic

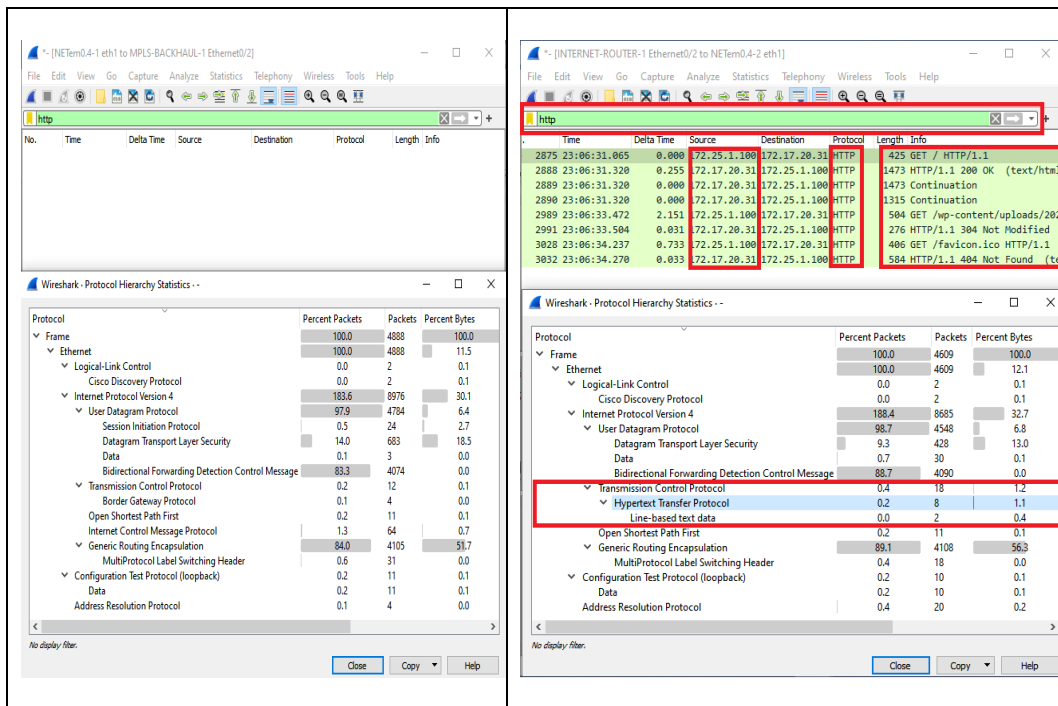
Hasil pengujian manipulasi *traffic voice* berdasarkan skenario pada Gambar 3, *traffic voice* yang ter-capture menggunakan perangkat lunak Wireshark menggunakan *protocol SIP (Session Initiation Protocol)* dan *RTP (Real Time Protocol)*. Dari hasil *packet capture* pada Gambar 7 pada *link internet* saat dilakukan panggilan tidak terdapat *traffic* dengan protokol SIP dan RTP.



Gambar 7 Hasil Packet Capture Manipulasi Traffic Voice

Sedangkan dari hasil *packet capture link MPLS* pada Gambar 7 terdapat 0,1 % *traffic* protokol SIP dan 97,7% *traffic* protokol RTP dari total paket UDP yang melewati *link MPLS*. Dari kedua hasil *packet capture* tersebut menunjukkan bahwa perangkat SD-WAN mampu mengklasifikasikan dan mengalirkan *traffic* dengan baik dan sesuai dengan yang dirancang.

Hasil pengujian manipulasi *traffic data* berdasarkan skenario pada Gambar 4, *traffic data* yang tercapture pada *link MPLS* pada saat pengujian Branch-User1 mengakses *web server* `http://172.17.20.31` melalui *web browser* yang ditunjukkan pada Gambar 8 tidak terdapat statistik adanya *traffic* dengan protokol HTTP yang melewati *link MPLS*. Sedangkan pada *link internet*, melalui statistik *protocol hierarchy* Wireshark menunjukkan adanya paket data dengan protokol HTTP GET *request* dari *source* IP 172.25.1.100 ke *web server* dengan IP 172.17.20.31 dan HTTP *response* 200 OK dari *web server*. Dari hasil *packet capture* pada Gambar 8 menunjukkan bahwa manipulasi *traffic* untuk melewatkan *traffic* HTTP melalui *link internet* berjalan dengan baik sesuai dengan yang dirancang.

Gambar 8 Hasil *Packet Capture* Manipulasi *Traffic* Data

3.2 Hasil Pengujian QoS *Traffic Voice*

Hasil pengujian QoS *traffic voice* berdasarkan skenario pada Gambar 5 pada jaringan SD-WAN dan BGP *Multihomed* dengan adanya *background traffic* yang dibangkitkan secara bertahap dari 10% sampai dengan 90% dari *total bandwidth* 5 Mbps pada saat panggilan berlangsung selama 10 menit yang terukur pada Wireshark di *interface Server VOIP* ditunjukkan pada Tabel 6.

Tabel 6 Hasil Pengukuran QoS *Traffic Voice*

<i>Background Traffic</i> (Kbps)	SDWAN			BGP <i>Multihomed</i>		
	<i>Average Latency</i> (ms)	<i>Average Jitter</i> (ms)	<i>Packet Loss</i> (%)	<i>Average Latency</i> (ms)	<i>Average Jitter</i> (ms)	<i>Packet Loss</i> (%)
512	20.00	10.26	0	20.00	10.41	0
1024	20.00	11.02	0	20.02	10.26	0
1536	20.00	10.94	0	20.00	10.40	0
2048	20.00	11.29	0	20.00	10.62	0
2560	20.03	26.73	0.2	21.14	12.15	4.9
3072	20.08	32.45	0.4	21.65	11.67	7.5
3584	20.11	30.14	0.5	22.65	12.67	11.6
4096	20.52	27.22	2.4	23.27	13.17	14
4608	20.77	27.26	3.7	23.62	12.77	15.3

Hasil pengujian QoS *traffic voice* dari segmen *branch* dengan menggunakan SD-WAN *application aware routing* diperoleh nilai rata-rata *latency* terbesar didapatkan pada saat *background traffic* 90% dari *total bandwidth* yaitu sebesar 20,77 ms. Sedangkan untuk nilai rata-rata *jitter* terbesar didapatkan pada saat *background traffic* 60% dari *total bandwidth* yaitu 32,45 ms. Nilai *jitter* yang cukup

besar pada jaringan SD-WAN terjadi karena adanya mekanisme *buffer latency* sebelum dan sesudah terjadinya *link failover*. Nilai *jitter* yang besar dan tidak stabil pada jaringan suara dapat menyebabkan delay waktu penerimaan suara. Lalu *packet loss* terbesar yaitu 3,7% didapatkan pada saat *background traffic* sebesar 90% dari total *bandwidth*. Dari hasil pengujian tersebut, QoS *traffic* jaringan SD-WAN *application aware routing* masih dalam kategori “Bagus” dan “Sangat Bagus” berdasarkan standar TIPHON. Kecilnya *packet loss* pada saat terjadi *high traffic* pada jaringan SD-WAN karena terdapat mekanisme *link failover* saat parameter QoS melebihi *threshold*.

Sedangkan hasil pengujian QoS *traffic voice* dari segmen *branch* dengan menggunakan BGP *multihomed* diperoleh hasil nilai rata-rata *latency* terbesar didapatkan pada saat *background traffic* 90% dari *total bandwidth* yaitu sebesar 23,62 ms. Sedangkan untuk nilai rata-rata *jitter* terbesar didapatkan pada saat *background traffic* 80% dari *total bandwidth* yaitu 13,17 ms. Lalu *packet loss* terbesar yaitu 15,3% didapatkan pada saat *background traffic* sebesar 90% dari total *bandwidth*. Dari hasil pengujian tersebut, QoS *traffic* jaringan BGP Multihomed berdasarkan standar TIPHON masih dalam kategori “Sangat Bagus”, “Bagus” dan “Sedang” berdasarkan standar TIPHON. Namun nilai *packet loss* lebih dari 5% - 15,3% cukup besar untuk *traffic voice* dan dapat mengganggu komunikasi *voice*.

3.3 Hasil Pengujian Performansi Traffic Data

Hasil pengujian performansi *traffic data* berdasarkan skenario pada Gambar 6 pada jaringan SD-WAN dan BGP *Multihomed* dengan adanya *background traffic* yang dibangkitkan secara bertahap dari 10% sampai dengan 90% dari *total bandwidth* 5 Mbps pada saat HTTP *request* berlangsung dengan interval 1 detik dengan total koneksi 300 *request* yang terukur pada wireshark di *interface PC Branch-User* ditunjukkan pada Tabel 7. Pengukuran yang dilakukan berupa waktu respon, dan jumlah TCP *retransmission*.

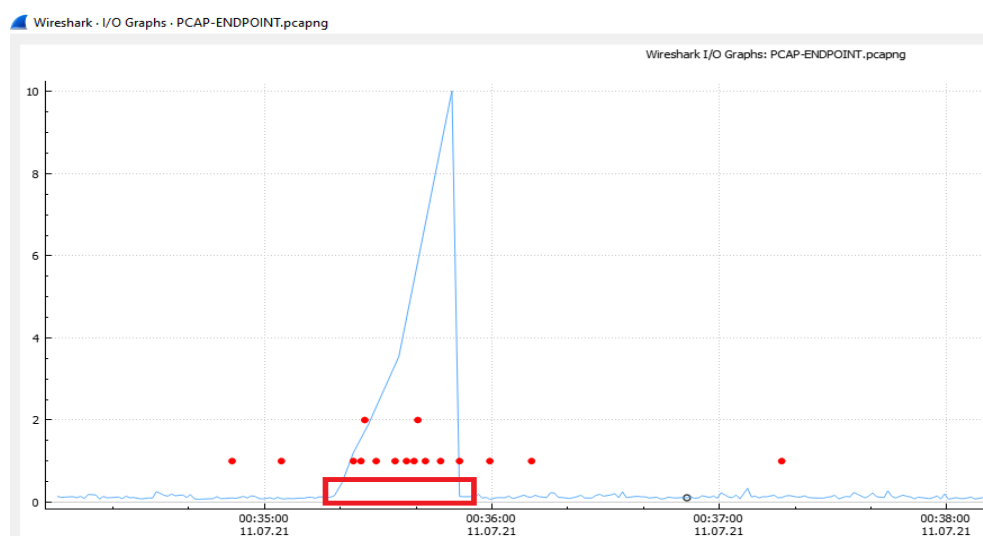
Tabel 7 Hasil Pengujian Performansi *Traffic Data*

<i>Background Traffic</i> (Kbps)	SD-WAN		BGP Multihomed	
	<i>Average Response Time</i> (ms)	Jumlah Paket TCP Retransmit	<i>Average Response Time</i> (ms)	Jumlah Paket TCP Retransmit
512	94.222	6	113.969	4
1024	96.558	12	126.069	7
1536	126.964	7	114.683	1
2048	95.891	10	129.598	9
2560	105.525	11	124.403	2
3072	136.191	7	185.561	7
3584	179.720	20	373.203	79
4096	169.861	18	455.309	83
4608	146.812	15	454.936	74

Hasil pengujian performansi HTTP pada jaringan SD-WAN menunjukkan rata-rata respon waktu maksimum terukur melalui Wireshark saat *background traffic* pada *link internet* 70% yaitu 179,720 ms. Sedangkan adanya paket TCP *retransmission* terbesar juga terjadi pada saat *background traffic* pada *link internet*

70%. Hal ini menunjukkan bahwa saat pengiriman paket HTTP *request* 300 paket dengan interval 1 detik dan *background traffic* lebih dari 50% waktu respon dan TCP *retransmission* yang didapatkan cukup tinggi. Respon waktu yang tinggi dan TCP *retransmission* dapat mempengaruhi kecepatan user untuk mengakses web HTTP.

Apabila dianalisis melalui hasil *packet capture* jaringan SD-WAN saat *background traffic* 3584 Kbps pada Gambar 9 sempat terdapat *respon time* yang tinggi yang ditunjukkan dengan grafik biru dan TCP *retransmission* yang ditunjukkan dengan titik merah yang cukup banyak pada periode proses perpindahan *traffic* dari *link* internet ke *link* MPLS yaitu 29 detik.



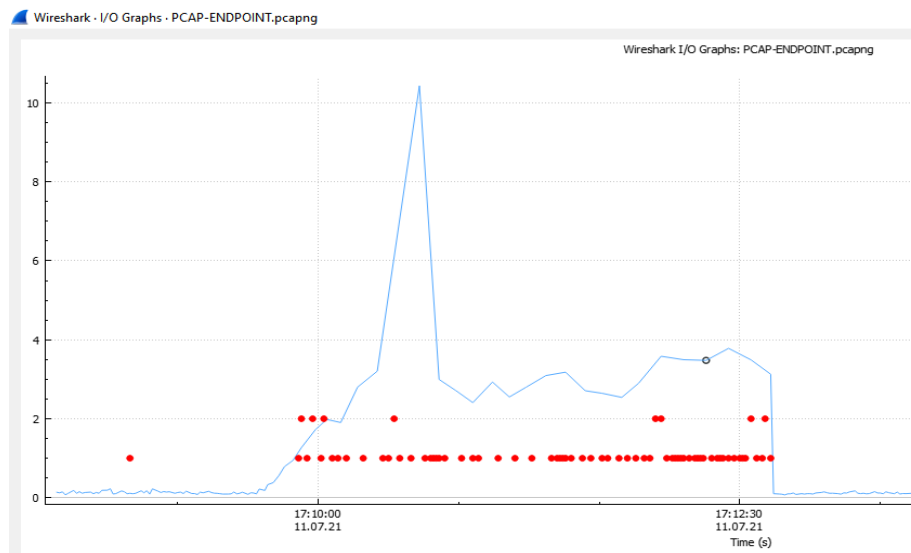
Gambar 9 Hasil *Packet Capture* SD-WAN *Background Traffic* 3584 Kbps

Sedangkan hasil pengujian performansi HTTP pada jaringan BGP *multihomed* menunjukkan rata-rata respon waktu maksimum terukur melalui wireshark saat *background traffic* pada *link* MPLS 1 90% yaitu 454,936 ms. Paket TCP *retransmission* terbanyak terjadi pada saat *background traffic* pada *link* MPLS 1 80%. Apabila dianalisis melalui hasil *packet capture* jaringan BGP *multihomed* saat *background traffic* 3584 Kbps pada Gambar 10 sempat terdapat *respon time* yang tinggi dan TCP *retransmission* yang cukup banyak selama periode *background traffic* dibangkitkan yaitu selama 3 menit

Hal ini menunjukkan bahwa baik jaringan SD-WAN dan BGP saat pengiriman paket HTTP *request* 300 paket dengan interval 1 detik dan *background traffic* lebih dari 60% waktu respon dan TCP *retransmission* yang didapatkan cukup tinggi. Hal ini disebabkan karena terjadinya *bottleneck* pada *link* yang digunakan oleh *traffic* HTTP. Namun periode waktu terjadinya *respon time* tinggi dan TCP *retransmission* pada jaringan SD-WAN lebih singkat apabila dibandingkan dengan BGP.

Hasil *response time* terbesar yang terukur pada saat pengujian yaitu 10 detik dan TCP *retransmission* terjadi sebanyak 77 kali selama periode 3 menit terjadinya *traffic congestion* pada *link* MPLS 1. Berdasarkan perbandingan respon waktu dan TCP *retransmission* antara jaringan SD-WAN dan MPLS pada Gambar 9 dan

Gambar 10 menunjukkan mekanisme *application aware routing* yang telah diimplementasikan dapat mengurangi impact menurunnya performansi akses web HTTP response time dan TCP *retransmission* akibat adanya gangguan kualitas latency, jitter dan packet loss pada salah satu *link* WAN.



Gambar 10 Hasil *Packet Capture* BGP *Background Traffic* 3584 Kbps

4. KESIMPULAN

Mekanisme optimasi *traffic voice* dan *data* menggunakan metode *application aware routing* berhasil diimplementasikan dan bekerja dengan baik. *Application aware routing* yang diimplementasikan dapat mengurangi *impact* akibat adanya penurunan performa pada salah satu *link* WAN baik berupa penurunan kualitas QoS maupun *traffic congestion*. Performansi QoS *traffic voice* saat *peak traffic* menggunakan SD-WAN *application aware routing* lebih baik dibandingkan dengan BGP *Multihomed* dengan rata-rata *latency* 20,77 ms, rata-rata *jitter* 27,26 ms dan *packet loss* 3,7%. Performansi QoS *traffic voice latency* saat *peak traffic* menggunakan SD-WAN masih dalam kategori “Sangat Bagus” sedangkan untuk *jitter* dan *packet loss* masih dalam kategori “Bagus” berdasarkan standar TIPHON (*Telecommunication and Internet Protocol Harmonization Over Network*). Performansi *traffic data* HTTP saat *peak traffic* menggunakan SD-WAN *application aware routing* lebih baik dibandingkan dengan BGP *Multihomed* dengan rata-rata *response time* 146.812 ms, dan TCP *retransmission* 15 paket.

REFERENSI

- [1] S. Badotra and S. N. Panda, “A survey on software defined wide area network,” *Int. J. Appl. Sci. Eng.*, vol. 17, no. 1, pp. 59–73, 2020, doi: 10.6703/IJASE.202003_17(1).059.
- [2] Y. Zhang, N. Ansari, M. Wu, and H. Yu, “On wide area network optimization,” *IEEE Commun. Surv. Tutorials*, vol. 14, no. 4, pp. 1090–1113, 2012, doi: 10.1109/SURV.2011.092311.00071.
- [3] P. A. Dhakulkar, P. S. Dubey, A. A. Gaikwad, S. P. Dhokane, and P. G. A. Dashmukhe, “Software Defined Wide Area Network,” vol. 3, no. 6, pp. 27–32, 2018, Computer Science and Engineering DES’s college of engineering and technology Dhamangaon Rly., India.
- [4] D. Zad Tootaghaj, F. Ahmed, P. Sharma, and M. Yannakakis, “Homa: An Efficient

- Topology and Route Management Approach in SD-WAN Overlays,” *Proc. - IEEE INFOCOM*, vol. 2020-July, pp. 2351–2360, 2020, doi: 10.1109/INFOCOM41043.2020.9155503.
- [5] R. Pambudi and M. A. Muslim, “Implementasi Policy Base Routing dan Failover Menggunakan Router Mikrotik untuk Membagi Jalur Akses Internet di FMIPA Unnes,” *J. Teknol. dan Sist. Komput.*, vol. 5, no. 2, p. 57, 2017, doi: 10.14710/jtsiskom.5.2.2017.57-61.
- [6] W. Lin, Y. Niu, X. Zhang, L. Wei, and C. Zhang, “Using path label routing in wide area software-defined networks with openflow,” *Proc. - 2016 Int. Conf. Netw. Netw. Appl. NaNA 2016*, pp. 149–154, 2016, doi: 10.1109/NaNA.2016.40.
- [7] M. Badrul and Akmaludin, “Implementasi Quality of Services (Qos) Untuk Optimalisasi Manajemen Bandwidth,” *Prosisko Vol. 6 No. 1 Maret 2019*, vol. 6, no. 1, pp. 1–9, 2019, [Online]. Available: <http://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/download/1120/931>.
- [8] E. R. Huddiniah, E. M. Safitri, S. A. Priyambada, M. Nasrullah, and N. D. Angresti, “Optimasi Rute Untuk Software Defined Networking-Wide Area Network (SDN-WAN) Dengan Openflow Protocol,” *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 13, no. 1, p. 7, 2018, doi: 10.30872/jim.v13i1.1006.
- [9] N. L. M. Van Adrichem, C. Doerr, and F. A. Kuipers, “OpenNetMon: Network monitoring in OpenFlow software-defined networks,” *IEEE/IFIP NOMS 2014 - IEEE/IFIP Netw. Oper. Manag. Symp. Manag. a Softw. Defini. World*, 2014, doi: 10.1109/NOMS.2014.6838228.
- [10] P. U-Chupala *et al.*, “Application-oriented bandwidth and latency aware routing with open flow network,” *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, vol. 2015-Febru, no. February, pp. 775–780, 2015, doi: 10.1109/CloudCom.2014.90.
- [11] E. Ramadhan, A. Firdausi, and S. Budiyanto, “Design and analysis QoS VoIP using routing Border Gateway Protocol (BGP),” *2017 Int. Conf. Broadband Commun. Wirel. Sensors Powering, BCWSP 2017*, vol. 2018-Janua, pp. 1–4, 2018, doi: 10.1109/BCWSP.2017.8272556.
- [12] M. A. Satriawan, I. B. Rahmat, and S. Si, “Analisi Performansi Layanan VOIP Pada Jaringan SD WAN dengan Teknonogi Virtualisasi.” 2019, Universitas Telkom, Bandung.
- [13] M. Shaikh and R. Vadivel, “Available Online at www.ijarcs.info International Journal of Advanced Research in Computer Science,” *Int. J. Adv. Res. Comput. Sci.*, vol. 9, no. 2, pp. 742–744, 2018.
- [14] R. Wulandari, “Analisis QoS (Quality of Service) Pada Jaringan Internet (Studi Kasus : UPT LOKA Uji Teknik Penambangan Jampang Kulon – LIPI),” *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 162–172, 2016, doi: 10.28932/jutisi.v2i2.454.
- [15] ETSI, “Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS),” *Etsi Tr 101 329 V2.1.1*, vol. 1, pp. 1–37, 1999.
- [16] Hasanul Fahmi, “Analisis Qos (Quality of Service) Pengukuran Delay, Jitter, Packet Lost Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik,” *J. Teknol. Inf. dan Komun.*, vol. 7, no. 2, pp. 98–105, 2018.