

# Studi Performansi Protokol *Routing IS-IS* Pada Arsitektur Jaringan *Software Defined Network* (SDN)

1<sup>st</sup> Muhammad Khatami  
Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia  
muhmmdkhtmi@students.telkomuniversity.ac.id

2<sup>nd</sup> Siti Amatullah Karimah  
Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia  
karimahsiti@telkomuniversity.ac.id

3<sup>rd</sup> Muhammad Arief Nugroho  
Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia  
arif.nugroho@telkomuniversity.ac.id

**Abstrak-Software-Defined Networking** (SDN) merupakan arsitektur jaringan yang memisahkan antara *control plane* dan *data plane* sebuah jaringan, sehingga jaringan dapat dikendalikan dari sebuah SDN *Controller*. SDN membawa perubahan bentuk jaringan dengan konfigurasi minimal oleh administrator jaringan. Dengan perkembangan teknologi saat ini dibutuhkan jaringan yang memiliki performansi tinggi. Salah satu yang penting adalah *network routing*. Tugas akhir menggunakan protokol *routing IS-IS* (*Intermediate System - Intermediate System*). Metode yang digunakan dalam penelitian tugas akhir ini adalah Studi literatur tentang SDN dan *IS-IS*, Pengembangan protokol *routing IS-IS* pada SDN, Pengujian performansi dan analisis. Parameter yang dijadikan pengujian performansi dan analisis adalah *packet loss*, *jitter*, *throughput*, *Network Convergence Time*. Dari hasil pengujian dapat disimpulkan bahwa *routing protocol IS-IS* pada arsitektur jaringan konvensional mendapatkan nilai parameter yang lebih baik dibandingkan dengan arsitektur jaringan SDN dalam parameter *throughput*, *packet loss*, dan *network convergence time*. Namun dari segi *jitter* dapat disimpulkan bahwa *routing protocol IS-IS* menggunakan arsitektur jaringan SDN mendapatkan nilai lebih baik dibandingkan jaringan konvensional.

Kata kunci- *routing*, SDN, *IS-IS*, *controller*, performansi, jaringan. QoS

**Abstrak-Software-Defined Networking** (SDN), is a network architecture that separates the *control plane* and *data plane* of a network, thus enabling network control of an SDN *Controller*. SDN allows changing the shape of the network with minimal configuration by the

*network administrator*. With the development of technology today, a network that has high performance is needed. One of the important ones is the *network routing*. final task using the *IS-IS* (*Intermediate System - Intermediate System*) *routing protocol*. The method used in this final project research is a literature study of SDN and *IS-IS*, Development of *IS-IS* *routing protocols* on SDN. Performance testing and analysis. The parameters used as performance testing and analysis are *packet loss*, *jitter*, *delay*, *throughput*, and *network Convergence Time*. From the test results, it can be concluded that the *IS-IS* *routing protocol* in conventional network architectures gets better parameter values compared to SDN network architectures in *throughput* parameters, *packet loss*, and *network convergence time*. However, in terms of *jitter*, it can be concluded that the *IS-IS* *routing protocol* using the SDN network architecture gets better scores than conventional networks.

**Keywords:** *routing*, SDN, *IS-IS*, *controller*, performance, networking

## I. PENDAHULUAN

### A. Latar Belakang

Seiring perkembangan zaman teknologi seperti saat ini, banyak aspek yang telah berkembang, termasuk teknologi di bidang arsitektur jaringan. Perubahan tersebut terjadi karena jaringan konvensional memiliki konfigurasi yang kompleks dan sulit dikelola. Salah satu alasannya karena kontrol dan bidang data yang saling terintegrasi. *Software-Defined Network* (SDN) hadir sebagai solusi untuk mengurangi kelemahan dari jaringan konvensional. SDN memiliki prinsip

sentralisasi, dimana semua konfigurasi dilakukan pada bidang kontrol. Ini terjadi karena konsep SDN memisahkan bidang kontrol dari bidang data. SDN memiliki protokol yang dapat mengontrol *data plane* melalui *control plane* dengan memanfaatkan kontroler, protokol tersebut disebut OpenFlow[1]. Kontroler pada SDN dapat secara tepat memantau status jaringan secara *real time* dan dapat menulis strategi pemrosesan yang berbeda sesuai dengan persyaratan yang berbeda.[2]

Jika dibandingkan dengan menggunakan perutean statis, perutean dinamis dapat memperbarui *routing table* sesuai dengan status jaringan secara *real time*. Saat ini, penelitian *dynamic routing algorithm* dengan menggunakan arsitektur jaringan SDN, berfokus menggunakan *shortest path* dengan menghitung jarak terpendek dari setiap titik atau *node*.

Untuk penelitian tugas akhir ini, menggunakan studi performansi perutean pada arsitektur jaringan *Software Defined Network* (SDN) dengan menggunakan *dynamic routing protocol*, *Intermediate System to Intermediate System* (IS-IS). IS-IS, salah satu *dynamic routing protocol* yang populer digunakan untuk perutean intra *Autonomous System* (AS) karena protokol ini dirancang dan dikembangkan untuk mengirim paket yang dialamatkan dengan menggunakan protokol jaringan berdasarkan *Open System Interconnection* (OSI) model serta mendukung protokol jaringan yang menggunakan *Transport Control Protocol/Internet Protocol* (TCP/IP) yang menjadi de- facto protokol jaringan saat ini. Parameter pengujian dan analisis pada Tugas Akhir ini adalah *packet loss*, *jitter*, *delay*, dan *throughput*.

#### B. Rumusan Masalah

Berdasarkan latar belakang sebelumnya, rumusan masalah diangkat menjadi beberapa pertanyaan sebagai berikut:

1. Bagaimana implementasi *routing protocol* IS-IS pada arsitektur jaringan SDN?
2. Apakah *routing protocol* tersebut dapat dijalankan pada arsitektur jaringan SDN?

3. Berapa jumlah nilai *Network Convergence Time* dan *Quality of Service* (QoS) *routing protocol* IS -IS pada arsitektur jaringan SDN dan non SDN?
4. Apakah *routing protocol* IS – IS menggunakan arsitektur jaringan SDN lebih baik daripada tanpa SDN?

#### C. Tujuan

1. Implementasi *routing protocol* IS-IS pada arsitektur jaringan SDN.
2. Menghitung nilai dari *Network Convergence Time* dan *Quality of Service* (QoS) *routing protocol* IS – IS pada arsitektur jaringan SDN dan non SDN.
3. Melakukan analisa serta membandingkan kedua arsitektur jaringan mana yang lebih baik.

#### D. Batasan Masalah

Sejumlah permasalahan yang dibahas pada penulisan tugas akhir ini dibatasi ruang lingkup batasan masalah sebagai berikut:

1. Parameter yang dilakukan pengujian yaitu, *Network Convergence Time* dan *Quality of Service* pada *Routing Protocol* IS-IS dengan SDN dan tanpa menggunakan SDN.
2. Pengujian hanya berfokus dengan arsitektur jaringan konvensional dan SDN.
3. Pengambilan data parameter QoS menggunakan Wireshark dan Iperf sebagai *background traffic*.
4. *Bandwidth* pada pengujian ini menggunakan *default bandwidth* yang sudah disediakan oleh Mininet.
5. *Controller* yang digunakan menggunakan *Open Networking Operating System* (ONOS) serta Mininet sebagai *data plane*.

## II. KAJIAN TEORI

### A. Studi Literatur

TABLE 1  
STUDI LITERATUR

No	Judul	Penulis / Tahun	Abstrak
1	Analisis Dan Simulasi Perbandingan Qos Di Routing Protokol Mpls Ospf Dan Mpls IS-IS Di Jaringan Ipv6 Menggunakan Gns3 Untuk Layanan Video	A.Munggaran, R.Munadi, D.Perdana/2018	IS-IS sangat mumpuni untuk kebutuhan perutingan yang lebih luas karena alamat yang lebih panjang
2	<i>Similar yet different: Protocol design choices in IS-IS and OSPF</i>	Yu Zhang, Alexander Afanasyev, Lan Wang, Lixia Zhang/2017	IS-IS memiliki struktur satu dimensi yang sederhana
3	<i>Performance evaluation of OSPFv3 and IS-IS routing protocol on ipv6 network</i>	A. N. H. Ahmad Jaafar, S Salim, L A. Tiron/2017	IS-IS melakukan proses konvergensi pada jaringan, lebih cepat daripada proses yang dilakukan oleh OSPF[3]
4	<i>Quality of Service (QoS) in Software Defined Networking (SDN)</i>	Karakus dan Durrezi	Membahas kemampuan QoS dari protokol OpenFlow dengan meninjau versinya bersama dengan beberapa versinya yang terkenal
5	<i>Review of Interior Gateway Routing Protocols</i>	Usman,M., Oberafo, E. E., , Abubakar, M. A., Aminu, T., Modibbo, A., dan Thomas, S	<i>Packet loss routing</i> IS-IS lebih rendah dibandingkan dengan OSPF, artinya dalam segi <i>Quality of Service (QoS)</i> jauh lebih baik menggunakan IS-IS daripada OSPF[4]
6	Analisis Simulasi Penerapan Algoritma OSPF Menggunakan RouteFlow pada Jaringan Software Defined Network (SDN)	Ridha Muldina Negara, Rohmat Tulloh	Hasil pengukuran <i>time convergence</i> menunjukkan bahwa penambahan jumlah <i>switch</i> mempengaruhi pertambahan waktu konvergensi, sedangkan untuk parameter <i>Quality of Service (QoS)</i> pada peningkatan topologi <i>switch</i> didapatkan hasil yang masih sesuai dengan standar ITU-T G.1010 namun apabila ditambahkan <i>background traffic</i> yang memenuhi 50% <i>bandwidth</i> jaringan maka QoS memburuk.[5]

## B. Dynamic Routing Protocol

*Dynamic routing protocol* merupakan protokol *router* yang memiliki dan membuat tabel rute secara otomatis, dengan mendengarkan lalu lintas jaringan dan saling berhubungan antara router lainnya. Protokol *routing* mengatur perutean sehingga dapat berkomunikasi satu dengan yang lain dan saling memberikan

informasi satu dengan yang lain dan saling memberikan informasi rute yang dapat mengubah isi *forwarding table*, tergantung keadaan jaringannya. Dengan cara ini, *router* mengetahui keadaan jaringan yang terakhir dan mampu meneruskan data ke arah yang benar. Dengan kata lain, dynamic routing adalah proses pengisian data routing di *table routing* secara otomatis.[6]

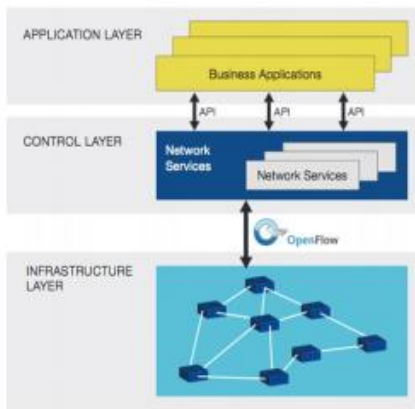
## C. Software Defined Network

Software Defined Network (SDN) memiliki kemampuan untuk mengelola, mendesain, dan mengimplementasikan sebuah arsitektur pada jaringan yang berfungsi untuk memisahkan *system control (control plane)* dan sistem *forwarding (forwarding plane)*. *Software Defined Network (SDN)* juga merupakan solusi masalah jaringan untuk mengefisienkan layanan aplikasi yang dapat disediakan baik secara lokal ataupun terpusat

bahkan dapat melalui komputasi awan[1]. Seluruh perangkat dapat diolah melalui satu media dimana media tersebut melewati satu protokol. *Software Defined Network (SDN)* dapat digunakan untuk konfigurasi *switch*, *router*, dan seluruh perangkat jaringan hanya dengan satu aplikasi saja.[7]

Arsitektur SDN membagi jaringan menjadi 3 layer yaitu application layer, control layer dan infrastructure/data layer. Application layer merupakan interface terhadap seorang admin atau peneliti dalam mengelola atau mengembangkan jaringan SDN.[1] *Control plane* berisi suatu kontroler bersifat terpusat dan *based on software*. *Subordinate hardware* dikontrol sepenuhnya oleh *controlling plane* atau kontroler dalam melakukan *forwarding decision*. Semua *subordinate* terhubung ke kontroler.[2]

Kontroler pada SDN dapat secara tepat memantau status jaringan secara *real time* dan dapat menulis strategi pemrosesan yang berbeda sesuai dengan persyaratan yang berbeda. Jika dibandingkan dengan menggunakan perutean statis, perutean dinamis dapat memperbarui *routing table* sesuai dengan status jaringan secara *real time*. [7]



GAMBAR 1  
ARSITEKTUR JARINGAN SDN

#### D. Intermediate System to Intermediate System (IS-IS)

*Intermediate System-to-Intermediate System* merupakan kepanjangan dari IS-IS. Adalah sebuah protokol yang hampir sama dengan OSPF, yang digunakan untuk menentukan cara terbaik untuk datagram oleh perangkat jaringan. Dengan linkstate database, IS-IS mengakomodasi kebutuhan protokol CLNP (*connectionless network protocol*) dan IP. Dimana router pertama akan mengirim paket hello dan akan menunggu paket hello balasan sebagai tanda ack dan juga menggunakan algoritma *Dijkstra*. IS-IS pun menggunakan sistem pembagian area untuk efisiensi pengiriman data.[8]

IS-IS juga merupakan hierarki protokol routing yang digunakan dengan perusahaan besar yang memungkinkan untuk mendefinisikan beberapa domain routing (zona) dan dengan demikian mengurangi ukuran tabel dan waktu konvergensi. Hal tersebut mendukung IPv4 dan IPv6, serta menggunakan algoritma SPF *Dijkstra* yang sama seperti OSPFv3 untuk menentukan jalur terpendek, dari ini ke itu untuk mencapai tujuan dalam jaringan. [6]

Keistimewaan IS-IS adalah, dapat membawa informasi jaringan IP, tetapi tidak menggunakan IP sebagai miliknya sebagai protokol transportasi. Karakteristik ini memungkinkannya menjadi sepenuhnya independen dari protokol yang dirutekannya (IPv6, IPv4). Tentang penanganan, IS-IS juga

memiliki format alamat yang berbeda dari salah satu OSPFv3. [3]

IS-IS menggunakan format ISO NET (*Network Entity Title*) untuk setiap router yang terlibat dalam proses perutean IS-IS, dan menggunakan dua "Level" dari perutean untuk router IS-IS: Level 1 (L1) untuk perutean intra-zona (hanya zona tempatnya berada diidentifikasi) dan Level 2 (L2) untuk perutean antar-zona (antara beberapa zona).[3]

#### E. Mininet

Mininet adalah emulator jaringan SDN yang dapat mensimulasikan kinerja antara end-host, switch, router, controller, dan link dalam sebuah kernel Linux. Mininet merupakan sebuah sistem virtualisasi yang dapat menggambarkan jaringan yang besar dengan hanya menggunakan sebuah laptop. Mininet bersifat open source, sehingga proyek yang telah dilakukan berupa source code, scripts, dan dokumentasi yang dapat dikembangkan oleh siapapun[9]

#### F. Open Networking Operation System (ONOS)

*Open Networking Operation System* (ONOS) merupakan *controller* SDN dengan basis *open source* untuk membangun SDN/NFV. ONOS dirancang untuk memenuhi kebutuhan sebagai *operator* dalam membangun dan mengembangkan layanan jaringan dinamis baru dengan antarmuka terprogram yang disederhanakan. ONOS mendukung konfigurasi dan kontrol real-time jaringan, menghilangkan kebutuhan untuk menjalankan perutean dan beralih protokol kontrol di dalam struktur jaringan.

#### G. Quality of Service (QoS)

*Quality of Service* (QoS) merupakan sebuah parameter yang terbagi menjadi beberapa bagian. Fungsi dari parameter diambil dari *packet* yang dikirimkan ini, digunakan untuk menentukan apakah kondisi jaringan stabil atau tidak. Adapun bagian parameter QoS sebagai berikut:

1. *Throughput* sebagai kecepatan arus *byte data per second* (Bps) sebagai jumlah total dari data yang sampai di penerima dalam rentang waktu pengiriman.[6]

$$Throughput = \frac{\sum \text{Ukuran Paket Diterima}}{\sum \text{Durasi pengukuran}} \quad (1)$$

2. *Jitter* merupakan dari variasi *delay* yang disebabkan variasi panjang antrian dalam pengolahan data, peningkatan traffic dan penyempitan *bandwidth*[6].

$$\text{Jitter} = \frac{\text{Total variasi delay}}{\sum \text{paket yang diterima}} \quad (2)$$

3. *Packet loss* adalah kondisi dimana paket yang telah dikirim hilang atau tidak sampai ke tujuan. Ada beberapa faktor yang menyebabkan hilangnya paket seperti penurunan sinyal, paket yang *corrupt*, atau *hardware failure*. [6]

$$\text{Packet Loss} = \frac{(\text{Packet yang dikirimkan} - \text{Packet yang diterima})}{\text{Packet yang dikirimkan}} \times 100\% \quad (3)$$

#### H. Network Convergence Time

*Network Convergence Time* merupakan waktu yang dibutuhkan semua *router* yang berada didalam jaringan untuk dalam keadaan *steady state* ketika terjadinya perubahan

kondisi jaringan. Perubahan kondisi ini dapat terjadi karena banyak hal, yaitu putusnya link antar dua *router*, *congestion*, *routing load processor* dan penyebab lainnya[10]

$$T_{\text{convergence}} = T_{\text{up}} - T_{\text{down}} \quad (4)$$

Keterangan :

$T_{\text{up}}$  = waktu ketika jaringan terhubung

$T_{\text{down}}$  = waktu ketika jaringan terputus

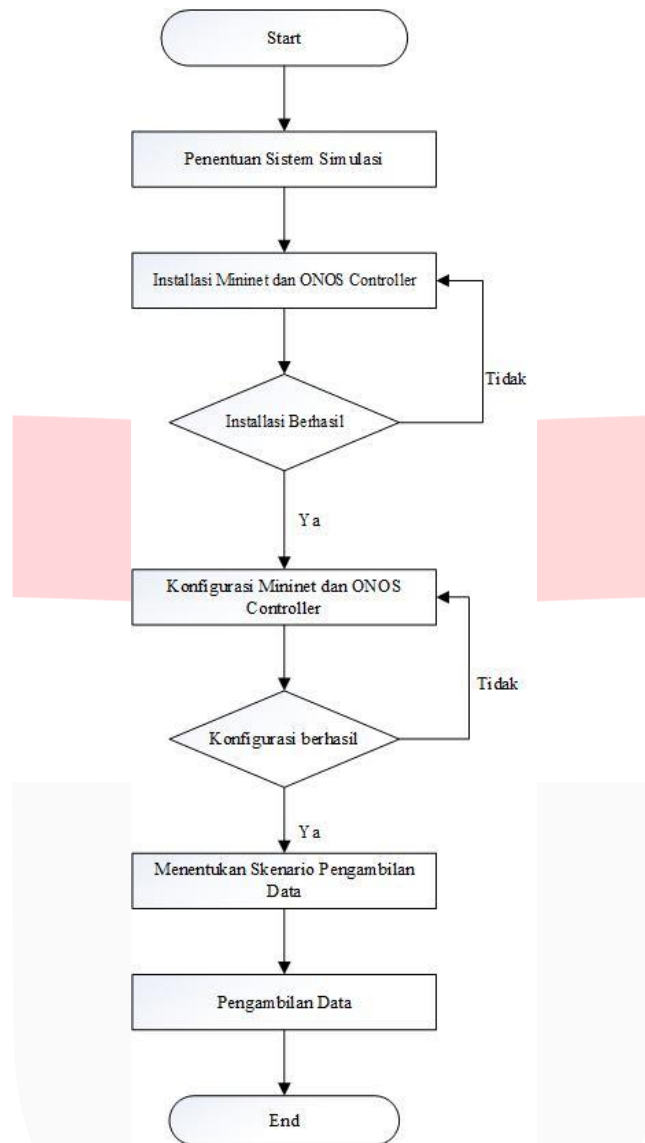
Pada bab ini dijelaskan beberapa hal yang berkaitan dengan perancangan dalam tugas akhir. Sub-bab terdiri dari 3.1 perancangan sistem simulasi, 3.2 topologi jaringan, 3.3 perangkat simulasi sistem, dan 3.4 skenario pengujian.

### III. METODE

#### A. Perancangan Sistem Simulasi

Pada sub-bab 3.1 ini dijelaskan tentang alur diagram dalam proses skenario penelitian tugas akhir. Berikut ini merupakan alur diagram skenario penelitian tugas akhir:



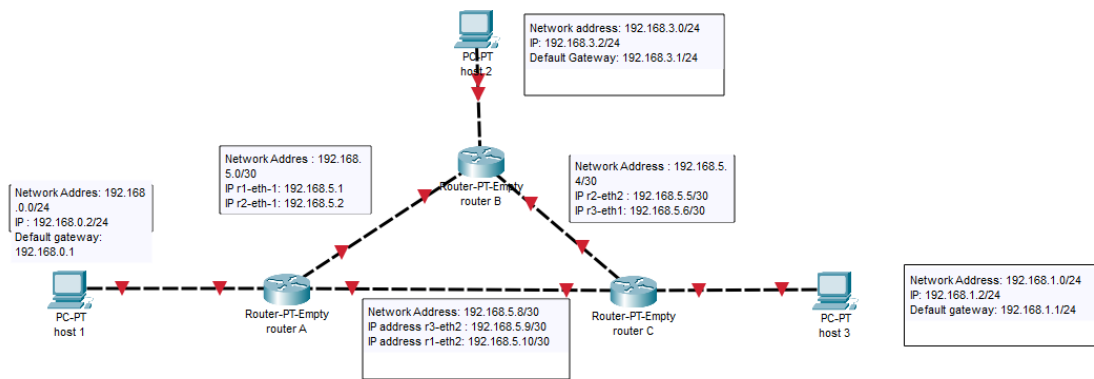


GAMBAR 2  
ALUR Pengerjaan Tugas Akhir

Pengerjaan tugas akhir dimulai dengan menentukan *control plane* dan *data plane* yang digunakan. Tugas akhir ini menggunakan OpenFlow *environment* sebagai *control plane* dan *emulator* Mininet bekerja sebagai *dataplane*. Kemudian dilakukan instalasi terhadap OpenFlow dan Mininet pada satu *virtual machine*. Pengecekan instalasi keduanya dilakukan dengan menjalankan file `QuaggaISIS.py`. File tersebut menjalankan topologi jaringan empat switch dan empat host yang menjalankan protokol IS-IS. Pengecekan

konektivitas jaringan dilakukan dengan *command pingall* pada Mininet. Tahap selanjutnya dilakukan modifikasi konfigurasi pada OpenFlow dan Mininet. Tugas akhir ini mengkonfigurasi topologi jaringan dengan 3-switch 3-host dengan protokol IS-IS. Jika konfigurasi sudah berjalan, maka ditentukan 6skenario pengambilan data.

B. Topologi Jaringan dan Konfigurasi Mininet



GAMBAR 3  
TOPOLOGI JARINGAN

Perancangan pada tugas akhir menggunakan tiga *router* serta tiga *host* untuk arsitektur jaringan SDN terdapat penambahan *controller* sehingga jaringan dapat dikontrol melalui sebuah pengontrol SDN. Pada konfigurasi datapath di dalam Mininet berupa pembuatan topologi. Hal ini dilakukan dengan cara inisialisasi *host* beserta alamat *gateway* yang ditujunya dan berapa banyak *router* yang digunakan pada Mininet. Kemudian menghubungkan setiap *router* dan *host* tersebut dengan mengatur link mana saja yang dibentuk antar-node.

### C. Perangkat Simulasi

Simulasi tugas akhir ini menggunakan sebuah *virtual machine*. Virtual machine digunakan sebagai *Control Plane* dan *Data Plane*. Pada setiap simulasi, *virtual machine* yang terpasang Mininet menggunakan *remote controller* yang ditujukan ke alamat IP virtual machine yang sudah terpasang ONOS *controller*, sehingga setiap *router* Mininet terhubung dengan kontroler di virtual machine.

Berikut adalah spesifikasi *virtual machine* yang digunakan sebagai *control plane* dan *data plane*:

TABLE 2  
PERANGKAT SIMULASI

OS	CPU	RAM	HDD
Ubuntu 20.04 LTS	2 Core	2 GB	50 GB

Penggabungan kontroler dengan Mininet pada satu *virtual machine* bertujuan untuk mendapatkan hasil simulasi yang maksimal dan lebih mudah dalam pengambilan data. Sehingga *virtual machine* menjalankan simulasi jaringan *data plane* dan *control plane*. Terdapat beberapa perangkat lunak yang digunakan untuk membantu pengambilan data, yaitu :

- Iperf, digunakan untuk membangkitkan background traffic UDP.
- Wireshark, digunakan untuk pengambilan data QoS

### D. Skenario Pengujian

#### 1. Skenario Uji Parameter *Throughput*

Skenario pengambilan data *throughput* menggunakan Iperf dengan *background traffic*

yang bervariasi, untuk ukurannya yaitu 40 Mbps, 60 Mbps, 100 Mbps, 200 Mbps, dan 300 Mbps serta dengan uji coba waktu yang dilakukan sebanyak 10, 20, 30, 40, dan 50 detik. Analisa *throughput* juga menggunakan *wireshark* sebagai *network analyzer*.

#### 2. Skenario Uji Parameter *Packet Loss*

Pengujian *packet loss* dilakukan dengan mengirimkan paket UDP dengan variasi *buffer size* 64 KB hingga 1024 KB dengan uji coba waktu yang dilakukan sebanyak 10, 20, 30, 40, dan 50 detik. Untuk *background traffic*nya menggunakan 100 Mbps setiap melakukan pengujian. Analisa pengujian *packet loss* menggunakan tool Iperf.

#### 3. Skenario Uji Parameter *Jitter*

Pengujian *jitter* dilakukan dengan mengirimkan paket UDP dengan variasi *buffer size* 64 KB hingga 1024 KB dengan uji coba waktu yang dilakukan sebanyak 10, 20, 30, 40, dan 50 detik. Untuk *background traffic*nya menggunakan 100 Mbps setiap melakukan pengujian. Analisa pengujian *jitter* menggunakan tool Iperf.[11]

#### 4. Skenario Uji Network Convergence Time

Pengujian *network convergence time* dilakukan dengan melakukan *link failure* antar dua *router*. Pengukuran ini bertujuan untuk bagaimana *routing protocol* IS-IS menjalankan *autonomous system* dengan memperbarui *routing table* secara otomatis. Sehingga ketika terjadi *link failure*, maka secara otomatis mencari rute terdekat.

### IV. HASIL DAN PEMBAHASAN

Pada Bab 4 ini menjelaskan hasil dari pengujian protokol *routing* IS-IS pada *Software Defined Network* (SDN) dan *routing*

konvensional. Hasil pengujian ini menggunakan dengan mengambil *background traffic*nya menggunakan Iperf dan untuk pengambilan parameter QoS menggunakan aplikasi Wireshark. Terdapat sub-bab yang terdiri 4.1 Hasil Evaluasi Parameter *Quality of Service* (QoS) dan 4.2 Hasil Evaluasi *Network Convergence Time*.

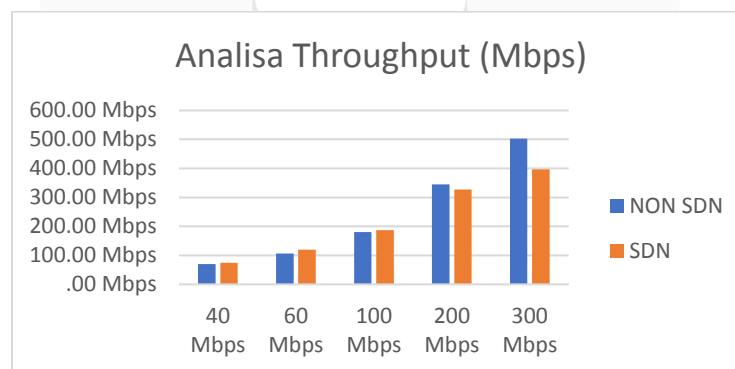
#### A. Hasil Evaluasi Parameter Quality of Service (Qos)

##### 1. Parameter Throughput

Untuk pengambilan *throughput* menggunakan Iperf dengan *background traffic* yang bervariasi, untuk ukurannya yaitu 40 Mbps, 60 Mbps, 100 Mbps, 200 Mbps, dan 300 Mbps serta dengan uji coba waktu yang dilakukan sebanyak 10, 20, 30, 40, dan 50 detik. Analisa *throughput* juga menggunakan *wireshark* sebagai *network analyzer*[11]. Berikut ini adalah hasil dari *throughput* pada topologi menggunakan non SDN dan menggunakan SDN:

TABLE 3  
HASIL ANALISA THROUGHPUT

Bandwidth	NON; SDN	SDN
40 Mbps	69.61 Mbps	74.49 Mbps
60 Mbps	106.23 Mbps	119.91 Mbps
100 Mbps	180.73 Mbps	187.46 Mbps
200 Mbps	345.02 Mbps	327.02 Mbps
300 Mbps	502.38 Mbps	397.14 Mbps



GRAFIK 1  
ANALISA THROUGHPUT

##### 2. Parameter Packet Loss

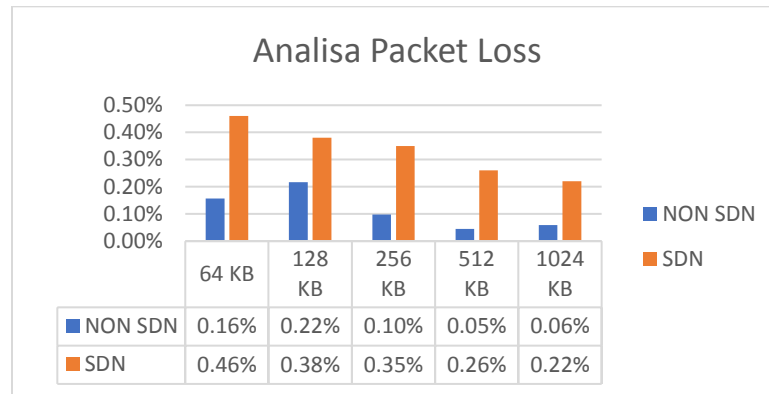
Pengujian *packet loss* dilakukan dengan mengirimkan paket UDP dengan variasi *buffer size* 64 KB hingga 1024 KB dengan uji coba waktu yang dilakukan sebanyak 10, 20, 30,

40, dan 50 detik. Untuk *background traffic*nya menggunakan 100 Mbps setiap melakukan pengujian. Analisa pengujian *packet loss* menggunakan tool Iperf.[11]



TABLE 4  
ANALISA PACKET LOSS.

Buffer; Size	NON; SDN	SDN
64 KB	0.16%	0.46%
128 KB	0.22%	0.38%
256 KB	0.10%	0.35%
512 KB	0.05%	0.26%
1024 KB	0.06%	0.22%



GRAFIK 2  
ANALISA PACKET LOSS

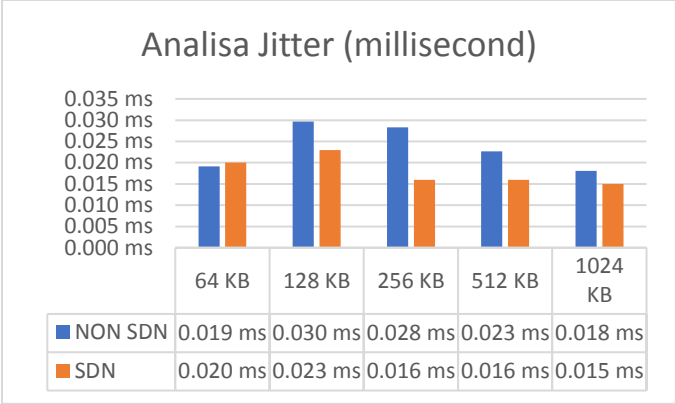
### 3. Parameter *Jitter*

Pengujian *jitter* dilakukan dengan mengirimkan paket UDP dengan variasi *buffer size* 64 KB hingga 1024 KB dengan uji coba waktu yang dilakukan sebanyak 10, 20, 30,

40, dan 50 detik. Untuk *background traffic*nya menggunakan 100 Mbps setiap melakukan pengujian. Analisa pengujian *jitter* menggunakan *tool* Iperf[11].

TABLE 5  
ANALISA JITTER

Buffer; Size	NON; SDN	SDN
64 KB	0.019 ms	0.020 ms
128 KB	0.030 ms	0.023 ms
256 KB	0.028 ms	0.016 ms
512 KB	0.023 ms	0.016 ms
1024 KB	0.018 ms	0.015 ms



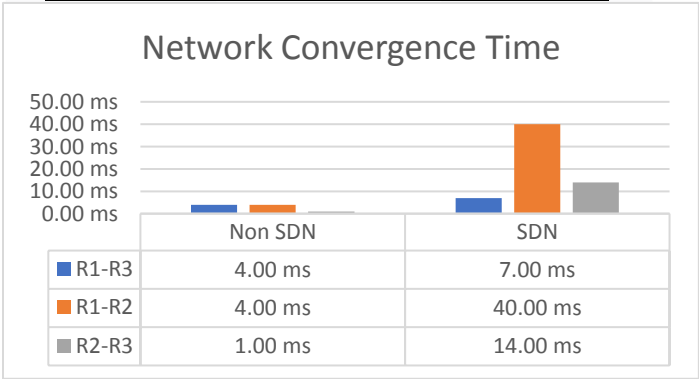
GRAFIK 3  
ANALISA JITTER

B. Hasil Evaluasi *Network Convergence Time*  
Pengujian *network convergence time* dilakukan dengan melakukan *link failure* antar dua *router*. Pengukuran ini bertujuan untuk bagaimana *routing protocol* IS-IS menjalankan *autonomous system* dengan memperbarui

*routing table* secara otomatis. Sehingga ketika terjadi *link failure*, maka secara otomatis mencari rute terdekat. Berikut adalah hasil dari pengukuran *network convergence time* pada topologi jaringan menggunakan SDN dan non SDN[5] :

TABLE 6  
ANALISA NETWORK CONVERGENCE TIME

Link; Failure	Non; SDN	SDN
R1-R3	4.00 ms	7.00 ms
R1-R2	4.00 ms	40.00 ms
R2-R3	1.00 ms	14.00 ms



GRAFIK 4  
ANALISA NETWORK CONVERGENCE TIME

V. KESIMPULAN

Protokol *routing* IS-IS dapat disimulasikan pada jaringan SDN maupun konvensional dengan menggunakan *virtual environment* OpenFlow pada SDN dan Mininet sebagai bagian yang mensimulasikan *data plane*, dengan hasil evaluasi performansi sebagai berikut:

- A. Hasil parameter QoS *routing protocol* IS – IS pada arsitektur jaringan konvensional mendapatkan nilai parameter yang cukup baik karena masih dalam rentang nilai yang sudah disesuaikan standarnya oleh ITU-T
- B. Hasil parameter QoS *routing protocol* IS-IS dengan menggunakan arsitektur

jaringan SDN, mendapatkan nilai parameter yang cukup baik dalam rentang nilai standar ITU-T namun jika dibandingkan dengan arsitektur jaringan konvensional dalam segi parameter *throughput* dan *packet loss* hasilnya jauh lebih baik dengan arsitektur jaringan konvensional. Parameter nilai *jitter* performansinya lebih baik dengan menggunakan arsitektur jaringan SDN

- C. Hasil parameter *network convergence time* pada *routing protocol IS-IS* dengan menggunakan arsitektur jaringan konvensional mendapatkan nilai parameter yang lebih baik dibandingkan dengan menggunakan arsitektur jaringan SDN. Hal ini disebabkan karena dalam arsitektur jaringan SDN, *control plane* harus menyesuaikan dengan *data plane* sementara jika jaringan konvensional masing – masing *router* memiliki *Autonomous System (AS)* sehingga tidak perlu lagi harus menunggu eksekusi dari *controller*.

#### REFERENSI

- [1] D. Medhi and K. Ramasamy, "Routing and Traffic Engineering in Software Defined Networks," in *Network Routing*, Elsevier, 2018, pp. 378–395. doi: 10.1016/b978-0-12-800737-2.00013-2.
- [2] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.
- [3] A. N. H. Ahmad Jaafar, S. Salim, L. A. Tiron, and Z. Mohd Hussin, "Performance Evaluation of OSPFv3 and IS-IS Routing Protocol on IPv6 Network."
- [4] M. Usman, E. Ehimeme Oberafo, M. Alhaji Abubakar, T. Aminu, A. Modibbo, and S. Thomas, "Review of Interior Gateway Routing Protocols," 2019.
- [5] R. M. Negara and R. Tulloh, "Analisis Simulasi Penerapan Algoritma OSPF Menggunakan RouteFlow pada Jaringan Software Defined Network (SDN)," *JURNAL INFOTEL*, vol. 9, no. 1, p. 75, Feb. 2017, doi: 10.20895/infotel.v9i1.172.
- [6] A. Putra Munggaran, M. T. Rendy Munadi, and D. Perdana, "ANALISIS DAN SIMULASI PERBANDINGAN QOS DI ROUTING PROTOKOL MPLS OSPF DAN MPLS IS-IS DI JARINGAN IPV6 MENGGUNAKAN GNS3 UNTUK LAYANAN VIDEO STREAMING ANALYSIS AND SIMULATION COMPARISON QOS OF ROUTING PROTOCOL MPLS OSPF AND MPLS IS-IS AT IPV6 USING GNS3 FOR VIDEO STREAMING SERVICES."
- [7] M. L. Naufal, M. Abdurrohman, and M. A. Nugroho, "Analisis High Availability Network Dengan Link Aggregation Pada Software Defined Network."
- [8] "A Comparison Between Two Routing Protocols: OSPF and IS-IS."
- [9] OPEN NETWORKING FOUNDATION, "OPEN NETWORKING FOUNDATION." <https://opennetworking.org/mininet/> (accessed Aug. 26, 2022).
- [10] V. Listiani, S. Naning Hertiana, R. M. Negara, K. Kunci: Ospf, and R. Routeflow, "ANALISIS PERFORMANSI SDN (SOFTWARE DEFINED NETWORK) MENGGUNAKAN PROTOKOL ROUTING OSPF (OPEN SHORTEST PATH FIRST) Analysis of Software Defined Network (SDN) Performance Using Open Shortest Path First (OSPF) Protocol Routing."
- [11] M. A. Fatahillah, M. Abdurrohman, and M. A. Nugroho, "ANALISIS PERBANDINGAN PERFORMANSI JARINGAN PADA ARSITEKTUR SOFTWARE-DEFINED NETWORK (SDN) DAN KONVENSIONAL COMPARATIVE ANALYSIS OF NETWORK PERFORMANCE ON SOFTWARE-DEFINED NETWORK

(SDN) ARCHITECTURE AND  
CONVENTIONAL,” 2016.

