

## **BAB 3**

### **METODE PENELITIAN**

#### **3.1 Teknik Pengumpulan Data**

Metode pengumpulan data yang digunakan penulis dalam menyusun laporan akhir ini antara lain:

##### **3.1.1 Studi Pustaka**

Studi Pustaka merupakan metode pengumpulan data yang dilakukan dengan membaca, mencari referensi-referensi yang berhubungan dengan topik yang dipilih dan yang berkaitan dengan objek penelitian. Referensi dapat berupa jurnal, buku, artikel maupun literatur-literatur tugas akhir lainnya. Studi Pustaka digunakan oleh penulis untuk penyusunan landasan teori, metode penelitian, dan cara analisis performansi *controller opendaylight* dan *ryu*.

##### **3.1.2 Perangkat Simulasi**

Pada tahap ini simulasi dilakukan untuk memperoleh data. Untuk melakukan simulasi, diperlukan beberapa alat pendukung guna terpenuhinya simulasi ini. Alat pendukung yang dimaksud berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

#### **3.2 Alat dan Bahan**

Dalam penelitian kali ini perangkat yang di gunakan dalam pengembangan sistem berupa perangkat keras dan perangkat lunak

##### **3.2.1 Perangkat Keras**

Perangkat keras yaitu perangkat yang berupa fisik yang akan digunakan dalam penelitian. Perangkat keras yang digunakan dalam penelitian ini adalah satu unit PC, berikut spesifikasi PC yang digunakan untuk mendukung simulasi yang dapat dilihat pada tabel 3.1. PC ini berfungsi untuk menjalankan aplikasi emulator jaringan yang akan digunakan, yaitu mininet.

**Tabel 3.1 Spesifikasi PC**

<i>Processor</i>	AMD Ryzen 3 3250U Radeon Graphics 2,6 GHz
RAM	4,00 GB
<i>Hardisk</i>	1,00 GB
OS	Windows 11

### **3.2.2 Perangkat Lunak**

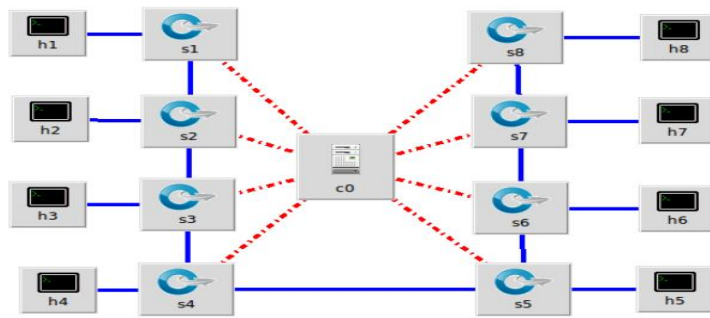
Perangkat lunak merupakan perangkat yang tidak dapat dilihat secara fisik. Perangkat lunak yang digunakan pada penelitian ini ialah sebagai berikut:

- a. *Virtual Box*
- b. *Mininet*
- c. *Ryu controller*
- d. *OpenDaylight Controller*
- e. *Software D-ITG*

### **3.3 Perancangan Topologi Jaringan**

Dalam simulasi ini, secara garis besar ditujukan untuk mengimplementasikan *Software Define Network (SDN)* dengan menggunakan kontroler *Ryu* dan *OpenDaylight* dan menggunakan protokol *Distributed-Internet Traffic Generator (D-ITG)* yang di mana selanjutnya *Quality of Service (QoS)* dari masing-masing kontroler akan dibandingkan sehingga diperoleh kontroler dengan performa terbaik.

Istilah topologi jaringan ini mengacu pada bagaimana bentuk perangkat jaringan yang saling terhubung apabila dilihat secara fisik dan membentuk pola tertentu. Untuk saling terhubung perangkat dalam jaringan memerlukan media transmisi kabel. Jenis topologi yang digunakan adalah topologi bus dengan menggunakan 8 *switch* dan 8 *router*. Diagram alur penelitian (*flowchart*) juga dapat diartikan proses secara bertahap yang dilakukan untuk memperoleh data yang akan dianalisis, seperti yang tertera pada gambar 3.1



**Gambar 3.1 Topologi Mininet**

Pada tabel 3.2 akan dijelaskan mengenai pemberian alamat pada *switch/router* dan *host* berdasarkan topologi yang telah dibuat sebelumnya.

**Tabel 3.2 Alamat IP Pada Jaringan**

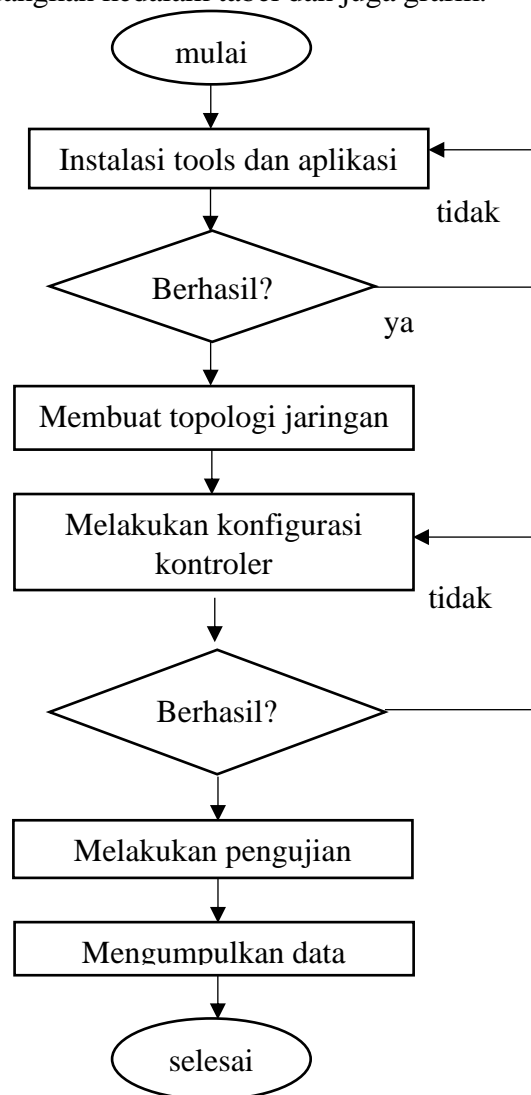
Perangkat	MAC	Perangkat	IP
<i>Switch 1</i>	00:00:00:00:01:01	<i>Host 1</i>	172.31.1.60
<i>Switch 2</i>	00:00:00:00:01:02	<i>Host 2</i>	172.31.1.70
<i>Switch 3</i>	00:00:00:00:01:03	<i>Host 3</i>	172.31.1.80
<i>Switch 4</i>	00:00:00:00:01:04	<i>Host 4</i>	172.31.1.90
<i>Switch 5</i>	00:00:00:00:01:05	<i>Host 5</i>	172.31.1.100
<i>Switch 6</i>	00:00:00:00:01:06	<i>Host 6</i>	172.31.1.110
<i>Switch 7</i>	00:00:00:00:01:07	<i>Host 7</i>	172.31.1.120
<i>Switch 8</i>	00:00:00:00:01:08	<i>Host 7</i>	172.31.1.130
<i>Controller</i>	-	-	127.0.0.1

### 3.3.1 Skenario Jaringan

Pada skenario jaringan dalam penelitian ini, penulis melakukan pengujian terhadap performa dari kontroler dalam analisis QoS untuk mengetahui kontroler mana yang memiliki performa terbaik, di mana kontroler yang digunakan dalam penelitian ini adalah kontroler *Ryu* dan kontroler *Opendaylight* dengan menggunakan *tools* D-ITG. Adapun alur dari skenario jaringan dapat dilihat pada gambar 3.2.

Setelah melakukan instalasi, maka selanjutnya adalah membuat topologi jaringan. Pada analisis kontroler *Ryu* dan *Opendaylight* pada topologi *bus* dengan jumlah *switch* 8 dan jumlah *host* 8 berdasar pada

parameter QoS seperti *throughput*, *delay*, *packet loss* dan *jitter* menggunakan mininet. Kemudian melakukan *input* dan *output* data yang diperlukan, seperti *command* atau perintah yang akan digunakan, *background traffic* yang dialirkan pada topologi, dan *output* yang merupakan hasil QoS yang akan dianalisis. Pengujian yang dilakukan dalam penelitian ini menggunakan variasi *background traffic* 50 MB, 100 MB, 150 MB, dan 200 MB, karena menurut (Rahmawan et al., 2020), variasi *background traffic* tersebut memberikan gambaran untuk mengetahui bagaimana pengaruhnya terhadap nilai dari parameter QoS. Hasil analisis kemudian akan diuraikan berupa nilai-nilai kecepatan dan waktu yang dituangkan kedalam tabel dan juga grafik.

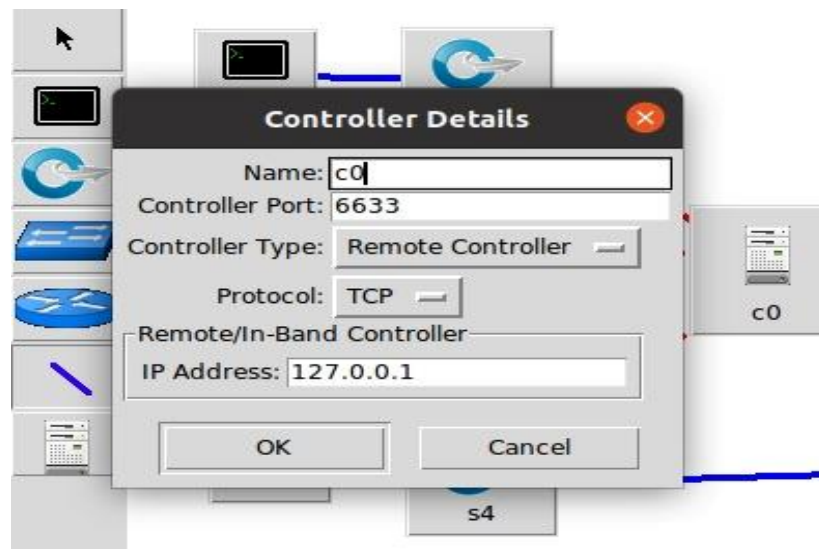


**Gambar 3.2 Flowchart simulasi**

### 3.3.2 Konfigurasi Jaringan

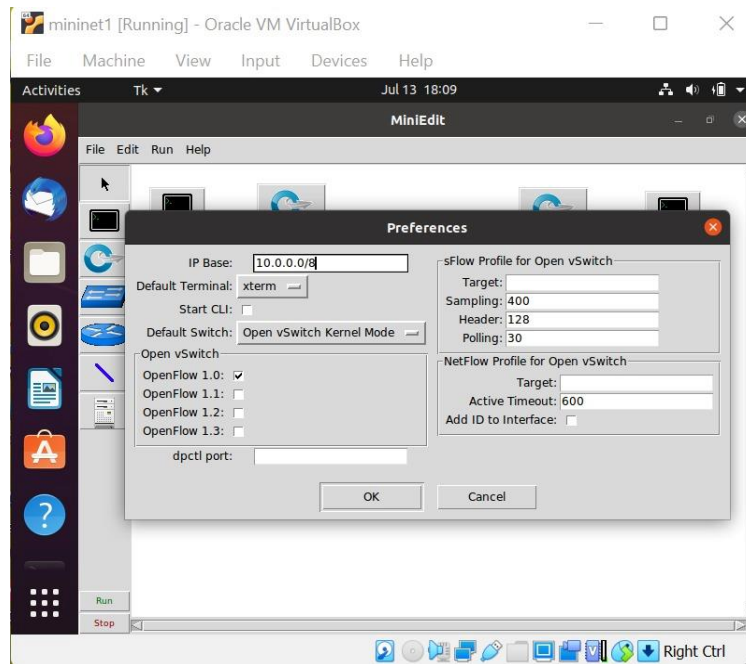
Pada tahap ini, dilakukan konfigurasi *controller* pada *miniedit* yang terdapat didalam *mininet*. Topologi yang dibuat terdiri dari 1 *controller Ryu* dan terhubung pada 8 *switch Openflow 1.0* yang masing-masing terhubung dengan 1 *host*.

Pada gambar 3.3 merupakan konfigurasi *controller* pada *miniedit*. Hal ini bertujuan agar *mininet* dapat terhubung pada *controller Ryu*, yaitu dengan cara klik kanan pada *controller*, kemudian pilih *properties* dan ubah *controller type* menjadi *remote controller*. Pada gambar tersebut juga dijelaskan bahwa *IP Address controller* yang digunakan adalah 127.0.0.1 dengan *controller port* 6633 (*controller port* pada umumnya adalah 6633 atau 6653).



**Gambar 3.3 Konfigurasi *remote controller***

Setelah melakukan konfigurasi *controller* dan topologi telah siap dijalankan, maka langkah selanjutnya adalah klik menu *edit* kemudian pilih *preferences* untuk konfigurasi *base*, dan kemudian centang *start CLI*. Untuk langkah selanjutnya, perhatikan gambar 3.4 berikut.



**Gambar 3.4 Edit preference controller**

Selanjutnya konfigurasi yang telah dibuat kemudian disimpan dengan klik *save* pada menu *bar file*. Export level 2 script merupakan pilihan untuk mengubah *file* menjadi *file* dalam bentuk *script python* dan *script miniedit*. Jika telah selesai merancang topologi, selanjutnya adalah menjalankan topologi yang telah dibuat. Apabila topologi telah berhasil dijalankan, maka langkah selanjutnya dapat dilaksanakan.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 h8
h6 -> h1 h2 h3 h4 h5 h7 h8
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 h5 h6 h7
*** Result: 0% dropped (56/56 received)
```

Pengujian yang dilakukan seperti yang dapat dilihat pada perintah diatas bertujuan untuk mengetahui apakah masing-masing *host* saling terhubung antara satu dengan yang lainnya. Perintah yang digunakan adalah *pingall*. Dari hasil yang telah diperoleh, yaitu 0% *dropped*, maka dapat

diketahui bahwa seluruh *host* telah terhubung satu dengan yang lain, karena tidak ada *broadcast* yang tidak tersampaikan. Jika koneksi ada yang gagal, maka proses pengujian tidak dapat berjalan.

```
mininet> links
h1 - eth0< - >s1 - eth1 (OK OK)
h2 - eth0< - >s2 - eth1 (OK OK)
h3 - eth0< - >s3 - eth1 (OK OK)
h4 - eth0< - >s4 - eth1 (OK OK)
h5 - eth0< - >s5 - eth1 (OK OK)
h6 - eth0< - >s6 - eth1 (OK OK)
h7 - eth0< - >s7 - eth1 (OK OK)
h8 - eth0< - >s8 - eth1 (OK OK)
s1 - eth2 < - > s2 - eth2 (OK OK)
s2 - eth2 < - > s3 - eth2 (OK OK)
s3 - eth2 < - > s4 - eth2 (OK OK)
s4 - eth2 < - > s5 - eth2 (OK OK)
s5 - eth2 < - > s6 - eth2 (OK OK)
s6 - eth2 < - > s7 - eth2 (OK OK)
s7 - eth2 < - > s8 - eth2 (OK OK)
```

Pengujian status *link* ini dilakukan untuk mengetahui apakah masing-masing *switch* telah terhubung dengan *host* dan telah terhubung dengan *switch* yang lainnya, serta untuk mengetahui status pada masing-masing *link ethernet* yang terhubung dalam *node* jaringan yang dibuat. Diketahui bahwa status pengujian *link* telah berjalan dengan baik, hal ini ditunjukkan oleh status (OK OK) pada CLI *mininet*.

```
mininet>net
h1 h1 -eth0:s1-eth1
h2 h2 -eth0:s2-eth1
h3 h3 -eth0:s3-eth1
h4 h4 -eth0:s4-eth1
h5 h5 -eth0:s5-eth1
h6 h6 -eth0:s6-eth1
h7 h7 -eth0:s7-eth1
h8 h8 -eth0:s8-eth1
s1 lo: s1-eth1:h1-eth0 s1-eth2:s2-eth2
s2 lo: s2-eth1:h2-eth0 s2-eth2:s1-eth2 s2-
eth3:s3-eth2
s3 lo: s3-eth1:h3-eth0 s3-eth2:s2-eth3 s3-
eth3:s4-eth2
s4 lo: s4-eth1:h4-eth0 s4-eth2:s3-eth3 s4-
eth3:s5-eth2
s5 lo: s5-eth1:h5-eth0 s5-eth2:s4-eth3 s5-
eth3:s6-eth2
```

```
s6 lo: s6-eth1:h6-eth0 s6:eth2:s5-eth3 s6-eth3:s7-eth2
s7 lo: s7-eth1:h7-eth0 s7:eth2:s6-eth3 s7-eth3:s8-eth2
s8 lo: s8-eth1:h8-eth0 s8:eth2:s7-eth3
co
```

Pada tahap ini, pengujian bertujuan untuk mengetahui dengan *detail* terkait koneksi antara *switch* dengan *switch*, *switch* dengan *host* dan *interface* apa yang digunakan. Diketahui bahwa *host* 1 (h1) terhubung ke *switch* 1 (s1) melalui *ethernet* 0 (eth0), sedangkan *switch* 1 (s1) terhubung ke *host* 1 (h1) melalui *ethernet* 1 (eth1), dan begitulah seterusnya.

### 3.4 Pengambilan Data

Pada proses ini, dilakukan pengujian dengan cara mengirimkan paket UDP menggunakan *tools* D-ITG dengan memberikan variasi *background traffic* sebesar 50 MB, 100 MB, 150 MB, dan 200 MB dengan masing-masing pengujian dilakukan sebanyak tiga kali, nilai yang nanti diperoleh berupa *delay*, *throughput*, *packet loss* dan *jitter*.

D-ITG akan berperan sebagai pengirim dan penerima paket. Untuk sisi pengirim, D-ITG akan melakukan *generate* data paket seperti yang telah diperintahkan dan kemudian mengirimnya kepada IP penerima. Untuk menjalankan program, maka perintah yang akan digunakan dapat dilihat pada perintah berikut.

```
./ITGSend -T UDP -a [alamat ip] -c 50000 -C 1000 -t 15000 -l
sender.log -x receiver.log
```

Setelah mengirim paket, maka D-ITG yang berperan sebagai sisi penerima akan melakukan listening terhadap paket data yang telah diterima. Untuk menjalankan program tersebut, maka perintah yang akan digunakan dapat dilihat pada perintah berikut .

```
./ITGRecv
```

Setelah semua program telah dijalankan, selanjutnya adalah membaca hasil yang telah diperoleh, kemudian akan diperoleh nilai dari



parameter yang sedang diuji, yaitu throughput, delay, jitter dan packet loss. Perintah yang digunakan dapat dilihat pada perintah berikut.

```
./ITGDec receiverXX.log
```

Perhitungan nilai throughput, delay, jitter dan packet loss akan dilakukan secara bersamaan