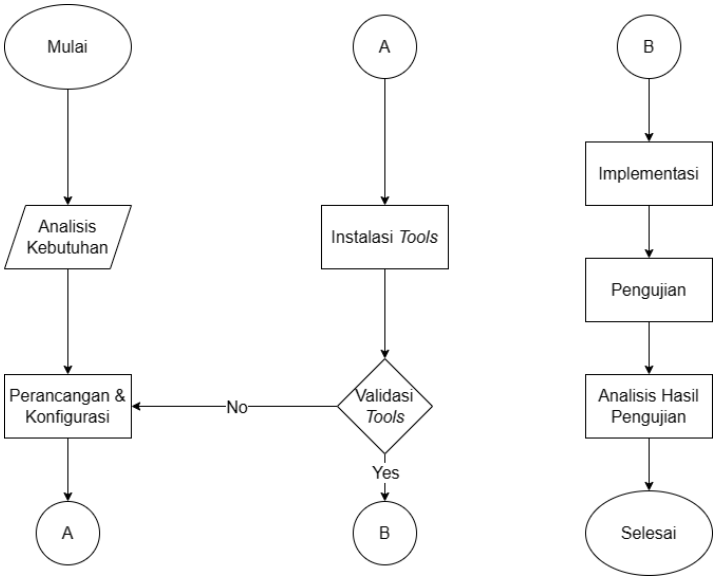
BAB III
METODE PENELITIAN

1.1 Metodologi

Penelitian ini menggunakan metode kuantitatif dengan pendekatan komparatif. Penelitian kuantitatif adalah metode penelitian yang melibatkan pengumpulan dan analisis data berdasarkan angka dan pengukuran numerik. Tujuan dari pendekatan komparatif adalah untuk menggambarkan, menjelaskan, dan menguji hubungan antara berbagai variabel menggunakan analisis statistik. Pendekatan komparatif umumnya bersifat *expost facto*, yaitu proses pengumpulan data akan dilakukan setelah fenomena yang menjadi titik permasalahan terjadi (Djollong, 2014).

Penelitian akan menghasilkan data kuantitatif berupa hasil perhitungan parameter QoS dan parameter *recovery time*. Data yang didapatkan tersebut akan dibandingkan antar kedua variabel, yakni *reactive forwarding* dan *intent-based reactive forwarding*. Hasil perhitungan dan perbandingan tersebut akan menjadi tolak ukur mekanisme manakah yang menampilkan performa dan skalabilitas yang paling tinggi.

Alur penelitian yang akan dijalankan ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Alur Penelitian

Pada tahapan analisis kebutuhan dikumpulkan landasan teori mengenai *Software Defined Network* (SDN), *controller*, dan mekanisme *routing*. Selain

mengenai variabel penelitian, diperlukan pula landasan teori terkait parameter pengujian serta *tools* yang akan digunakan dalam pengujian. Analisis dilakukan pula terkait kebutuhan fungsional penelitian seperti kebutuhan dalam perancangan sistem dan perkiraan mekanisme kerja dari rancangan sistem.

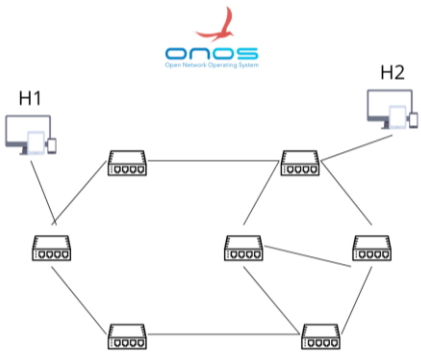
Pada tahapan perancangan, mulai dibangun lingkungan sistem yang akan digunakan pada sistem operasi. Maka dibutuhkan tahapan konfigurasi dan instalasi untuk fitur-fitur yang diperlukan oleh sistem seperti *controller* dan emulator jaringan. Setelah melakukan instalasi, perlu dipastikan apakah *tools* tersebut sudah bisa berjalan dengan baik ataukah terjadi *error* setelah instalasi. Hal ini penting untuk diperhatikan agar proses pengujian berjalan dengan baik dan hasil pengujian yang dihasilkan akurat.

Sistem yang sudah dirancang dan dikonfigurasikan perlu di implementasikan dan dipastikan apakah sudah siap untuk digunakan. Jika sistem sudah siap maka dapat dilakukan pengujian dengan skema yang sudah dirancang. Tahapan akhir dalam penelitian yaitu menganalisis hasil dan memberikan implikasi yang dihasilkan dari kajian teori dan kemudian disesuaikan dengan data yang didapatkan setelah melakukan pengujian.

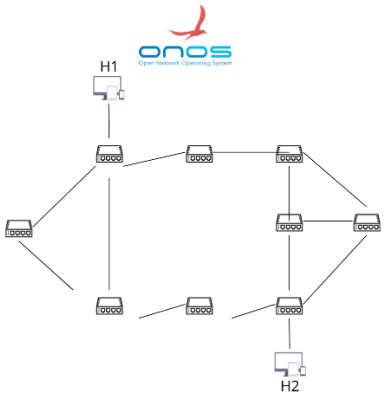
1.1 Alur Perancangan

1.1.1 Perancangan Topologi

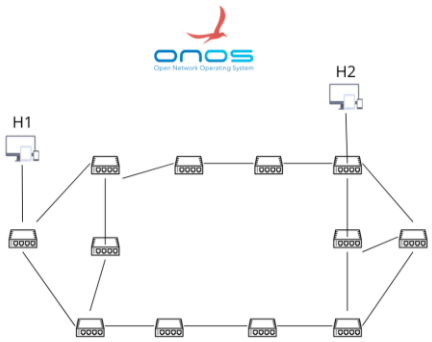
Pada penelitian ini akan menggunakan 3 topologi kustom yang berbeda, masing-masing topologi tersusun atas 2 *host* dan variasi jumlah *switch*. Topologi A memiliki 7 *switch* yang tersusun seperti pada Gambar 3.2, topologi B memiliki 9 *switch* yang tersusun seperti pada Gambar 3.3 dan topologi C memiliki 12 *switch* yang tersusun seperti pada Gambar 3.4.



Gambar 3. 2 Rancangan Susunan Topologi A



Gambar 3. 3 Rancangan Susunan Topologi B

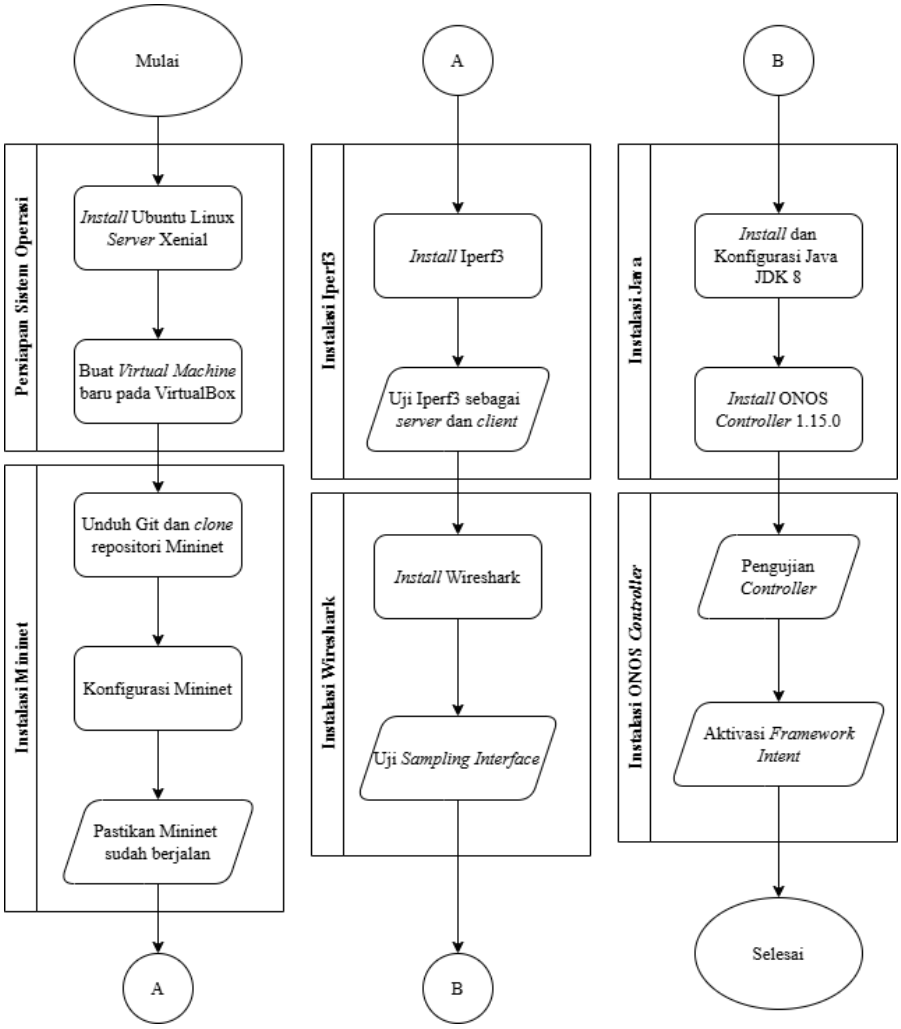


Gambar 3. 4 Rancangan Susunan Topologi C

Perbedaan skema pada topologi dan penggunaan jumlah switch yang berbeda bertujuan untuk meninjau performa kedua mekanisme *forwarding* baik *reactive forwarding* dan *intent-based reactive forwarding* apakah akan terpengaruh pada kompleksitas topologi dalam memanajemen trafik dan menangani kegagalan *link*.

1.1.2 Perancangan Sistem

Sistem yang akan dijalankan pada penelitian ini dapat dilihat pada Gambar 3.5.



Gambar 3. 5 Rancangan Sistem

Terlihat pada Gambar 3.5 bahwa sistem operasi Ubuntu Linux akan berjalan pada *Hypervisor* tipe II yaitu VirtualBox sebagai *Virtual Machine*. Ubuntu yang akan diinstall adalah Ubuntu Linux *Server* 16.04.3 (Xenial). Pemilihan versi Ubuntu ditentukan berdasarkan stabilitas dan kompatibilitas versi tersebut dengan seluruh komponen lainnya yang akan diinstall pada sistem.

Instalasi komponen pendukung lain seperti Java, Iperf3, Mininet, ONOS *Controller* dan Wireshark perlu dilakukan. Java diperlukan untuk mendukung berjalannya ONOS *Controller* dan *intent* didalamnya. Iperf3 diperlukan untuk mengirimkan beban trafik dalam jaringan. Wireshark digunakan sebagai alat untuk merekam dan *monitoring* transmisi data yang berjalan.

Pada masing-masing *virtual machine* akan menggunakan *bridged adapter* untuk memudahkan mengakses visualisasi simulasi.

1.2 Skenario Pengujian

Pada masing-masing skenario pengujian, mekanisme *forwarding* FWD dan IFWD dijalankan secara bergantian. Pada pengujian FWD harus mengaktifkan aplikasi *Reactive Forwarding* pada CLI ONOS dengan perintah berikut.

```
onos> app activate org.onosproject.fwd
```

Untuk menjalankan mekanisme IFWD diperlukan aktivasi aplikasi Proxy/ARP pada CLI ONOS. Setelah itu dapat dilakukan penambahan *host-to-host intent* pada masing-masing skema topologi. Susunan perintah yang dijalankan adalah sebagai berikut.

```
onos> app activate org.onosproject.proxyarp
onos> add-host-intent [Egress link H1] [Egress link H2]
```

Egress link pada H1 dan H2 disesuaikan dengan informasi perangkat yang ditampilkan setelah topologi dijalankan.

1.2.1 Pengujian QoS

Pengujian QoS ditujukan untuk menilai performa dan skalabilitas dari kedua mekanisme *forwarding* dalam menangani beban trafik yang tinggi diiringi dengan kompleksitas topologi yang digunakan. Hal ini ditujukan dalam rangka menilai sistem manakah yang memiliki kualitas tinggi dan dapat beroperasi dengan baik, sementara tetap mengelola sumber daya secara efisien.

Pengujian dilakukan secara bergantian antara mekanisme FWD dan IFWD. Masing-masing pengujian akan direkam oleh Wireshark dengan *interface* 'any' yang akan merekam seluruh transmisi data yang berjalan pada topologi tersebut.

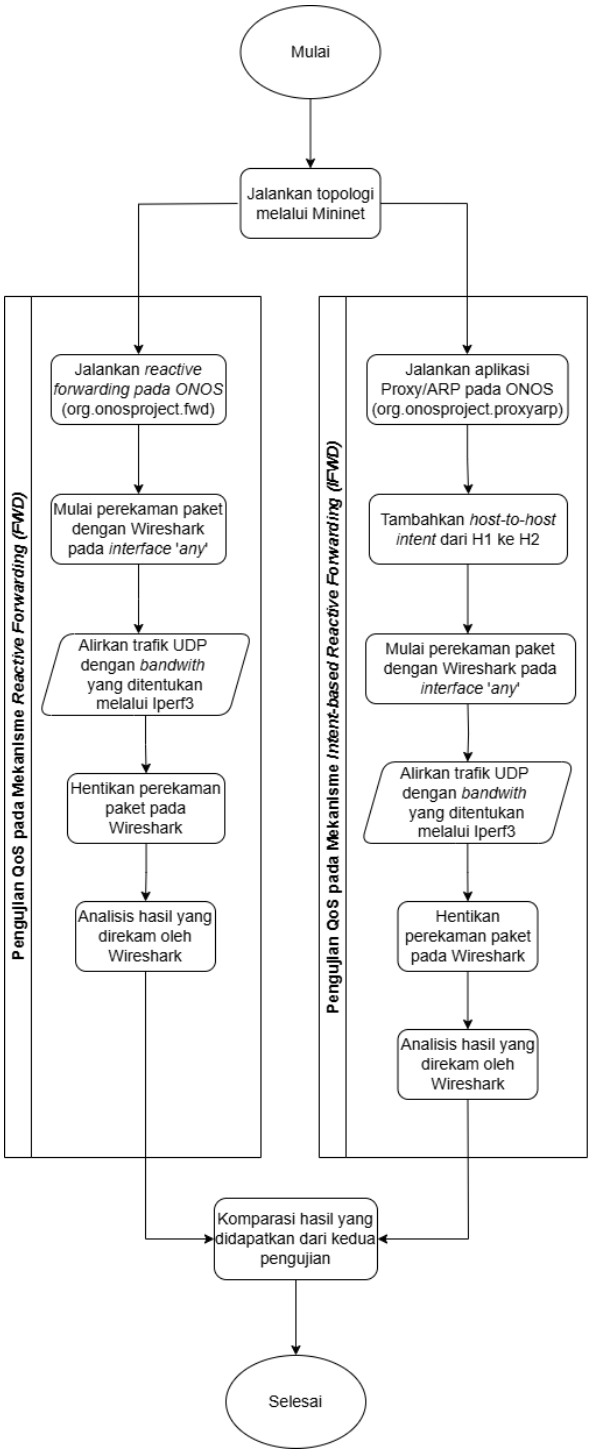
Pada CLI Mininet jalankan Xterm untuk H1 dan H2. Pada pengujian ini H1 akan berperan sebagai *server* dan H2 berperan sebagai *client*. Pada Xterm H2, jalankan perintah berikut untuk membanjiri trafik pada topologi.

```
iperf3 -c [IP tujuan] -u -b [besar bandwidth] -t 10
```

Keterangan:

- -c : Iperf3 yang dijalankan berperan sebagai klien
- -u : Mengirimkan paket UDP
- -b : Input besaran *bandwidth*
- -t : Waktu yang akan ditempuh untuk pengiriman paket

Alur pengujian parameter QoS yang dijalankan ditunjukkan pada Gambar 3.6.



Gambar 3. 6 Skenario Pengujian Parameter QoS

Masing-masing topologi dibanjiri trafik paket UDP yang meningkat secara bertahap sebesar 500 mbps, 1000 mbps, 1500 mbps, 2000 mbps dan 2500 mbps. *Bandwidth* maksimum pada *link* topologi akan diatur sebesar 1000 mbps. Skenario

pengujian trafik ini akan diterapkan pada 3 skema topologi, masing-masing topologi akan diuji coba sebanyak 15 kali repetisi untuk mendapatkan variasi data. Setelah mendapatkan data maka akan dirata-ratakan untuk mendapatkan hasil analisis performa.

1.2.2 Pengujian *Recovery Time*

Pada pengujian ini membutuhkan alat tambahan, yaitu Wireshark untuk memudahkan dalam analisis jaringan dan membantu dalam menganalisis lalu lintas jaringan secara *real-time*. Pengukuran membutuhkan 2 aplikasi Wireshark, aplikasi pertama akan mengukur di *interface 'any'* yang akan merekam seluruh aktivitas transmisi data yang berjalan. Sedangkan *interface* kedua akan difokuskan pada salah *switch* yang akan dilewati oleh jalur alternatif untuk melihat paket yang pertama masuk setelah pemutusan *link*.

Setelah topologi dan *controller* dijalankan maka IFWD dan FWD harus dijalankan secara bergantian. 30 paket ICMP akan dikirimkan melalui aplikasi Ping dengan perintah berikut.

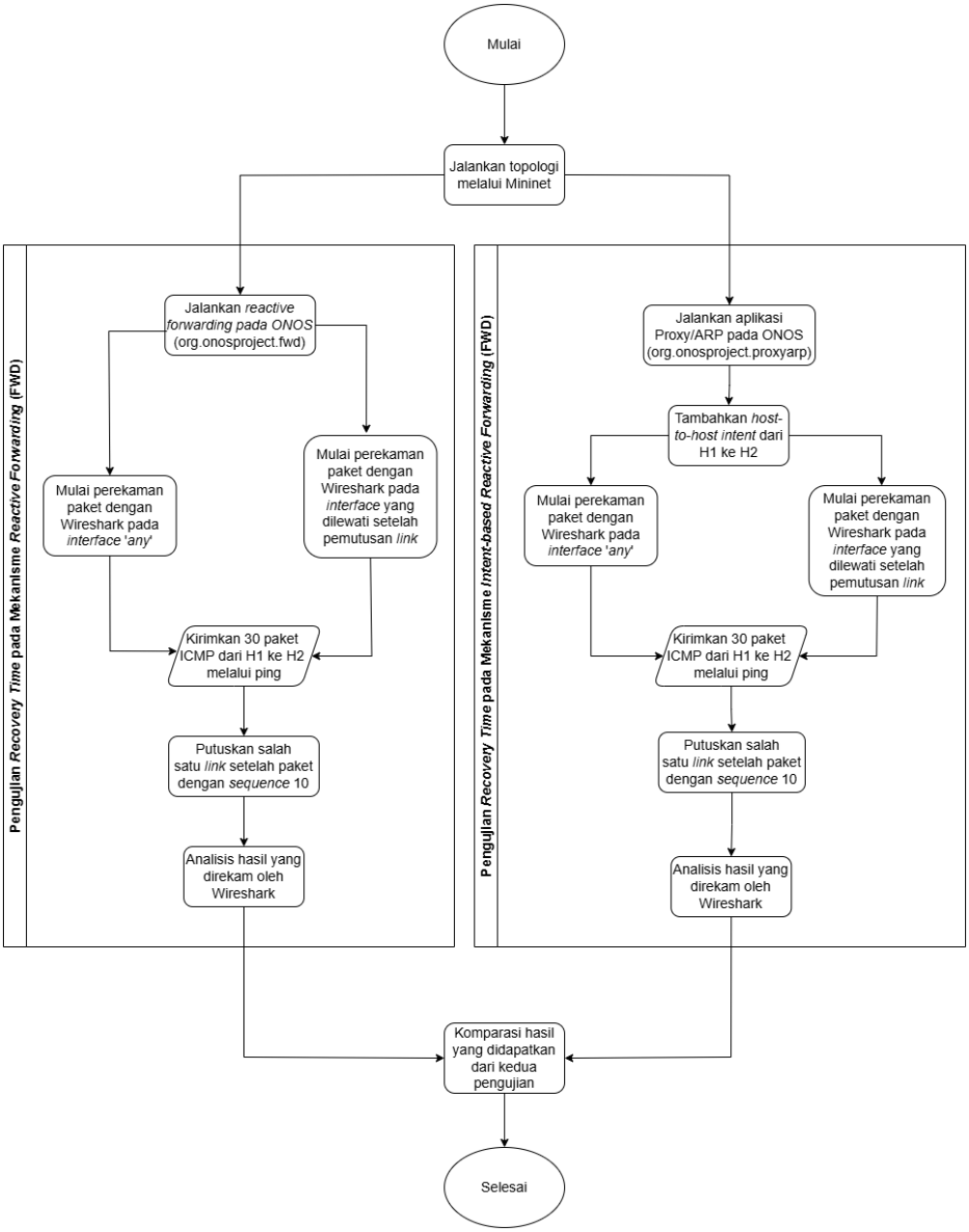
```
ping -c 30 [IP host tujuan]
```

Keterangan:

- -c 30 : Melakukan ping pada *client* dengan mengirimkan 30 paket ICMP

Pada paket dengan *sequence* 10 skema pemutusan *link* akan dilakukan. Ketika *link* diputus, mekanisme *forwarding* akan menentukan jalur alternatif yang menyebabkan perubahan rute. Pengukuran paket dilakukan dengan melihat paket pertama yang terekam pada Wireshark setelah terjadi pemutusan *link*. Pengujian *recovery time* akan dilakukan menggunakan dua skenario, *single-link failure* dan *multi-link failure*. Masing-masing skenario dijalankan sebanyak 10 kali repetisi.

Skenario *single-link failure* akan memutuskan salah satu *link* utama yang menjadi jalur transmisi data. Alur pengujian dengan skenaio *single-link failure* ditunjukkan pada Gambar 3.7.



Gambar 3. 7 Alur Pengujian *Recovery Time* dengan Skenario *Single-link Failure*

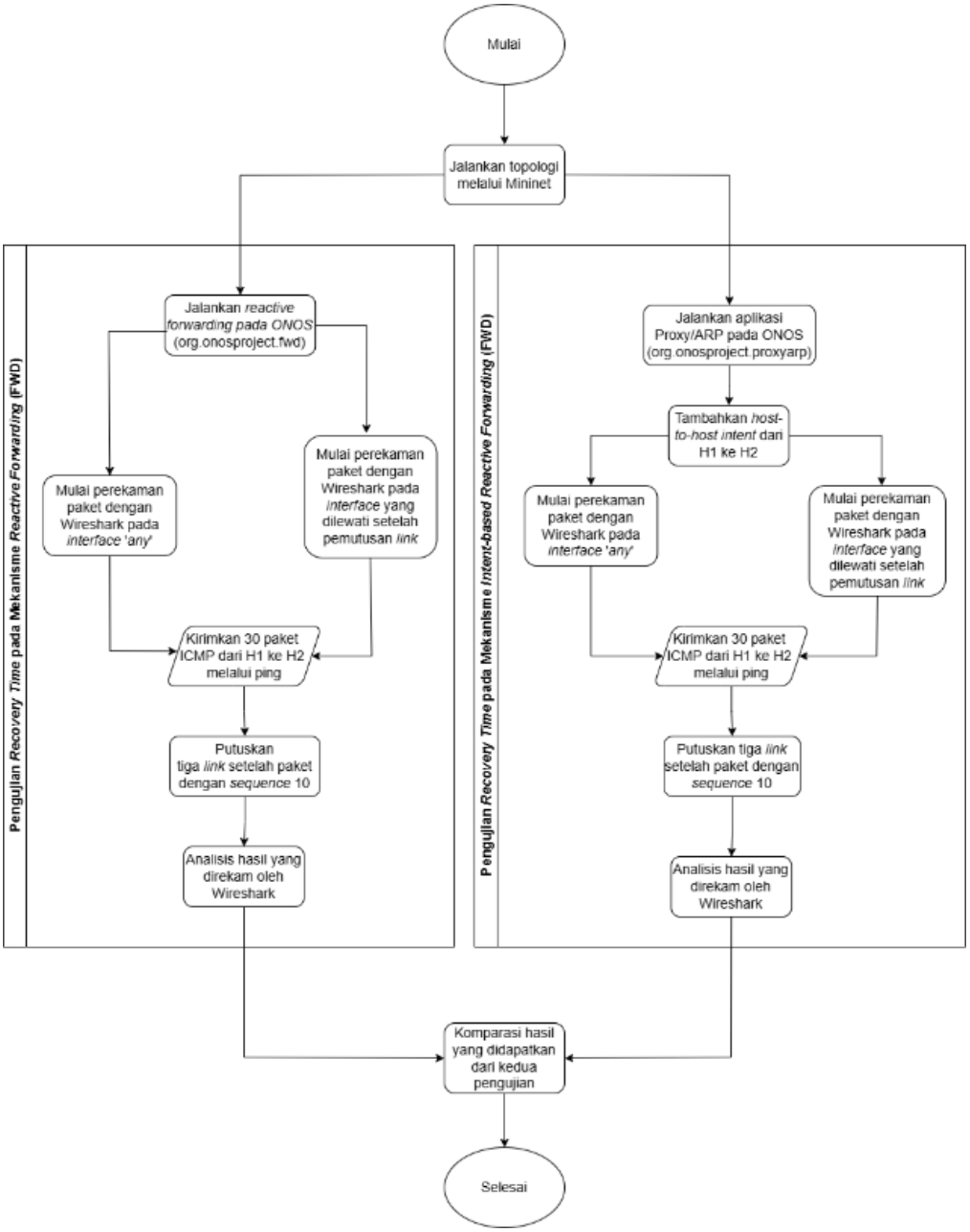
Pada skenario *single-link failure*, s1-eth2 akan menjadi *link* yang diputus. Pemutusan *link* dilakukan dengan menjalankan perintah berikut.

```
sudo ifconfig s1-eth2 down
```

Keputusan ini diambil karena pada masing-masing topologi, *link* s1-eth2 merupakan *link default* yang dilalui ketika menjalankan transmisi data. Maka dari itu, penting untuk menganalisis proses berjalannya transmisi data dalam kondisi normal sebelum terjadi pemutusan *link*. Hal ini juga ditujukan agar dapat

menentukan antarmuka yang akan digunakan untuk *monitoring* paket yang akan masuk pertama kali di jalur alternatif yang telah dibuat.

Skenario kedua adalah *multi-link failure* dimana ketika sedang terjadi proses transmisi data, dilakukan pemutusan pada 3 *link* secara bersamaan. Alur pengujian *recovery time* dengan skenario *multi-link failure* ditunjukkan pada Gambar 3.8.



Gambar 3. 8 Alur Pengujian *Recovery Time* dengan Skenario *Multi-link Failure*

Pemutusan tiga *link* secara bersamaan ini ditujukan untuk pembuktian lebih lanjut untuk mengukur keandalan mekanisme *forwarding* dalam menemukan jalur

alternatif setelah pemutusan *link*. Untuk memudahkan pemutusan *link* secara bersamaan, digunakan *script* dengan ekstensi .sh.

Perhitungan *recovery time* dapat dilakukan dengan mengurangi waktu yang ditunjukkan pada paket pertama yang diterima setelah pemutusan link dikurangi dengan waktu yang ditunjukkan pada paket *sequence* 10.