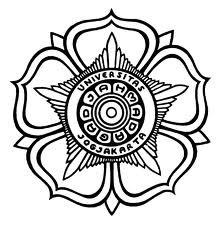
**RESEARCH PLAN**

**IMPLEMENTASI DAN ANALISIS KINERJA RIP PADA JARINGAN SDNMENGGUNAKAN RASPBERRY PI DAN PENGENDALI POX**

***IMPLEMENTATION AND PERFORMANCE ANALYSIS OF RIP ON SDN NETWORK USING RASPBERRY PI AND POX CONTROLLER***

****

**Diajukan oleh :**

**BUDI INDRA GUNAWAN**

15/384761/SV/09118

**PROGRAM STUDI SARJANA TERAPAN TEKNOLOGI REKAYASA INTERNET**

**DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA**

**SEKOLAH VOKASI**

**UNIVERSITAS GADJAH MADA**

**YOGYAKARTA**

**2018**

# BAB I

**PENDAHULUAN**

## Latar Belakang

Pengguna internet terus mengalami peningkatan setiap tahunnya. Menurut data yang dirilis oleh Asosiasi Penyelenggara Jasa Internet Indonesia, pada tahun 2016 terdapat 132,7 orang pengguna internet di Indonesia (APJII, 2016). Pertumbuhan pengguna yang disertai berkembangnya layanan baru seperti virtualisasi server, cloud computing, IPTV, dan internet of things menyebabkan meningkatnya kinerja routing yang pada jaringan konvensional merupakan bagian terpenting karena berfungsi sebagai pengatur lalu lintas jaringan dan memiliki kelemahan yaitu konfigurasi harus dilakukan secara individual dengan mengacu pada masing-masing vendor penyedia perangkat jaringan. Hal tersebut menyebabkan sebuah jaringan menjadi tidak fleksibel terhadap perubahan yang terjadi.

Beberapa tahun terakhir diajukanlah konsep Software Designed Network (SDN) yaitu pendekatan baru untuk mendesain, mengelola, dan mengimplementasikan arsitektur jaringan yang memisahkan antara sistem kontrol (control plane) dan sistem forwarding (data plane) pada perangkat jaringan (Eueng, 2015). SDN memungkinkan suatu jaringan secara dinamis menyesuaikan lingkungan unutk kebutuhan aplikasi maupun kebutuhan pengguna, menyederhanakan manajemen, meningkatkan skalabilitas, dan perangkat dari vendor yang berbeda juga dapat saling berkomunikasi karena dikelola menggunakan protokol yang disebut OpenFlow (Hikam, 2017).

Saat ini terdapat tiga pilihan metode untuk menerapkan arsitektur SDN yaitu penggunaan emulator Mininet, perangkat net-FPGA, dan switch yang berbasis OpenFlow. Metode pertama yaitu Mininet adalah sebuah emulator SDN yang didalamnya terdapat berbagai macam virtual end-host, switch, dan perangkat lainnya sehingga memungkinkan mendesain virtual link tanpa menggunakan perangkat sungguhan. Metode kedua yaitu menggunakan perangkat net-FPGA untuk menerapkan SDN pada perangkat sungguhan, namun memiliki kelemahan yaitu membutuhkan biaya yang besar dan kompleksitas karena menggunakan bahasa pemrograman khusus. Metode ketiga adalah menggunakan switch berbasis OpenFlow yang pada penelitian sejenis menggunakan perangkat-perangkat seperti mikrotik RB951-2n dan TP-Link WR-1043ND v2.

Penulis tertarik untuk menerapkan metode ketiga yaitu mengimplementasikan jaringan SDN skala kecil dengan menggunakan perangkat Raspberry pi dengan protokol routing RIP. RIP (*Routing Information Protocol*) merupakan sebuah protokol *distance vector* yang paling simpel, bandwidth yang dibutuhkan kecil, dan mudah diimplementasikan sehingga sangat cocok digunakan pada jaringan SDN skala kecil. Sedangkan Raspberry pi adalah sebuah komputer papan tunggal (*single-board circuit*) berharga murah dengan sistem operasi berbasis linux yang dapat diinstal pengendali POX dan OpenvSwitch sehingga dapat dirubah fungsinya menjadi sebuah *openflow switch*. Melalui topik ini diharapkan nantinya akan dihasilkan keluaran dengan parameter setara yang dapat digunakan untuk menentukan routing protokol dan penggunaan perangkat *open flow switch* berbiaya rendah yang terbaik untuk jaringan *Software Defined Network* skala kecil.

## Rumusan Masalah

Beberapa rumusan masalah pada penulisan proyek akhir ini adalah sebagai berikut :

1. Apakah routing protokol RIP dapat diterapkan pada jaringan *Software Defined Network* skala kecil dengan menggunakan perangkat Raspberry pi dan pengendali POX
2. Bagaimana kinerja routing protokol RIP pada implementasi menggunakan perangkat Raspberry dibandingkan dengan simulasi menggunakan Mininet

## Batasan Masalah

Beberapa batasan masalah yang mengacu pada tujuan penulisan proyek akhir ini adalah sebagai berikut :

Protokol routing yang dijalankan untuk penelitian ini adalah RIP dengan aplikasi RouteFlow

Kontroler yang digunakan untuk menjalankan SDN adalah POX

Implementasi topologi jaringan menggunakan X buah perangkat Raspberry pi B3 yang dijadikan sebagai openflow switch

Simulasi dilakukan pada satu perangkat komputer dengan aplikasi Mininet sebagai simulator jaringan SDN

Kinerja jaringan yang dianalisis yaitu QoS dan convergence time

Tidak membahas sistem keamanan jaringan

**BAB II**

**TUJUAN DAN MANFAAT PENELITIAN**

## Tujuan Penelitian

Tujuan dari penelitia dalam proyek akhir ini adalah sebagai berikut:

Implementasi protokol routing RIP pada jaringan Software Defined Network skala kecil dengan menggunakan perangkat Raspberry pi B3

Mendapatkan nilai latency, throughput, jitter, packet loss, dan convergence time untuk menganalisi performa kerja protokol routing RIP pada jaringan Software Defined Network skala kecil dengan menggunakan perangkat Raspberry pi B3 dibandingkan dengan simulasi menggunakan aplikasi Mininet

Mengetahui pengaruh topologi terhadap kinerja protokol routing RIP pada jaringan Software Defined Network

## Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian dan pengerjaan proyek akhir ini adalah sebagai berikut :

1. Informasi mengenai implementasi routing RIP pada jaringan SDN dengan menggunakan perangkat Raspberry pi dan pengendali POX
2. Informasi mengenai perbandingan kinerja routing RIP pada jaringan SDN menggunakan perangkat Raspberry pi dengan simulasi menggunakan aplikasi Mininet.
3. Memberikan pilihan dalam menentukan protokol routing RIP terbaik yang akan digunakan dalam membangun atau mengembangkan suatu jaringan.

**BAB III**

**TINJAUAN PUSTAKA**

SDN (Software Defined Network) adalah paradigma baru dalam perkembangan teknologi jaringan dengan memisahkan infrastruktur bagian *control plane* dan *data plane*. Banyak hal yang dapat diteliti dan dianalisa dari topik ini misalnya seperti routing, switching, dan teknologi lain yang sebelumnya biasa diimplementasikan pada jaringan konvensional, kini dicoba pada jaringan SDN. Penelitian jaringan SDN biasanya dilakukan pada software emulator Mininet. Software ini memungkinkan terciptanya jaringan *virtual host, switch, controller,* dan *link* dengan menjalankan standar perangkat lunak jaringan linux serta mendukung OpenFlow untuk *custom routing* yang sangat fleksibel.

Beberapa penelitian yang telah dilakukan pada jaringan SDN adalah adalah *Implementasi RIP pada Jaringan Berbasis SDN* (Hidayah et al.,2017). Penelitian ini menyimpulkah bahwa implementasi routing RIPv2 pada jaringan SDN menggunakan perangkat TP-Link WR-1043ND v2 dapat dilakukan dan dapat bekerja sebagaimana mestinya dengan memilih hop terkecil pada saat mengirimkan paket. Performa yang dihasilkan yaitu throughput dengan nilai rata-rata 100,21 Mbps, delay dengan nilai 47,43 ms, jitter dengan nilai 0,018 ms, packet loss dengan nilai 0%, dan convergence time dengan nilai 14,31 detik. Performa tersebut lebih baik dibanding dengan percobaan jaringan SDN pada emulator Mininet.

Penelitian lain yang berhubungan dengan penerapan routing protokol pada jaringan SDN adalah *Implementasi dan Analisis Kinerja Protokol Routing OSPF pada Jaringan SDN Berdasarkan Cost dengan Menggunakan RoutFlow* (Anam, 2017). Hasil penelitian ini adalah protokol routing OSPF beserta modifikasi nilai cost pada jaringan SDN dapat diimplementasikan dengan menggunakan perangkat Mikrotik RB951, perangkat lunak RouteFlow, dan Controller POX yang mendukung OpenFlow v1.0. Performa secara keseluruhan menunjukkan bahwa routing OSPF pada jaringan SDN memiliki kinerja yang lebih baik dibandingkan dengan jaringan konvensional.

Jika penelitian sebelumnya menggunakan routing internal, maka pada penelitian *Implementasi dan Analisis Kinerja Routing BGP pada Jaringan SDN* (Nugroho, 2017) mencoba menggunakan routing eksternal. Penelitian ini berupa implementasi pada perangkat Mikrotik RB951 dan kontroler POX dengan hasil protokol BGP dapat diterapkan pada jaringan SDN dengan topologi mesh. Performa yang dihasilkan berdasarkan parameter QoS memiliki nilai rata-rata terbaik delay (7,11597 ms), jitter (1,247303 ms), dan throughput (1,339281 Mbps). Nilai-nilai tersebut memiliki korelasi yakni mengalami peningkatan seiring dengan penambahan background traffic.

Sedangkan penelitian mengenai penggunaan Raspberry Pi pada jaringan SDN telah dilakukan dengan judul *Developing a Cost Effective OpenFlow Testbed for Small-scale Sofware Defined Networkintg* (Kim et al., 2014). Penelitian ini melakukan perbandingan performa menggunakan tiga metode yang berbeda yaitu penggunaan emulator Mininet, penggunaan perangkat khusus SDN yaitu net-FPGA, dan penggunaan Raspberry Pi. Hasil dari penelitian ini adlaah perangkat Raspberry Pi dapat difungsikan sebagai OpenFlow switch dengan performa mendekati perangkat net-FPGA, namun hanya bisa digunakan pada jaringan SDN skala kecil.

Penelitian lain yang berhubungan yaitu *Implementasi Switch OpenFlow Berbasis Software dengan Memanfaatkan Raspberry Pi untuk Infrastruktur SDN* (Santoso et al., 2018). Penelitian ini dilakukan dengan eksperimen membangun infrastruktur SDN dengan menggunakan Raspberry Pi sebagai OpenFlow Switch, Sebuah Server VM untuk Controller Floodlight, dan tiga buah PC dalam bentuk VM sebagai host. Hasil yang didapatkan adalah Raspberry Pi dapat digunakan sebagai OpenFlow Switch, namun performa QoS mengalami penurunan seiring dengan bertambahnya klien

**BAB IV**

**LANDASAN TEORI DAN HIPOTESIS**

## LANDASAN TEORI

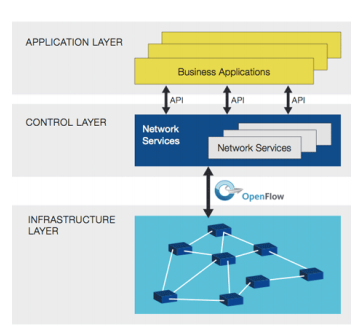
### SOFTWARE DEFINED NETWORK (SDN)

Software Defined Network (SDN) adalah paradigma baru dalam mendisain, mengelola, dan mengimplementasikan jaringan dimana sistem pengkontrol (control plane) dipisahkan dengan arus datanya (forwarding plane) (ONF, 2013). Ada tiga aspek penting dari SDN yaitu :

Adanya pemisahan secara fisik/eksplisit antara forwarding plane dengan control plane.

Control plane yang terpusat secara logika yaitu adanya sebuah siste operasi jaringan yang mampu membentuk peta logika dari seluruh jaringan yang kemudian dipresentasikan melalui API (Application Programming Interface).

Virtualisasi dimana beberapa sistem operasi jaringan dapat mengontrol bagian-bagian dari perangkat yang sama.



Gambar 4.1 Arsitektur SDN (ONF)

Gambar 4.1 menunjukan arsitektur SDN beserta komponen dan interaksinya terdiri dari tiga lapis/bidang yaitu :

1. Layer Aplikasi (*Application Layer*)

Layer Aplikasi berada pada lapisan teratas merupakan sebuah aplikasi end-user untuk mengatur dan mengoptimalkan jaringan secara baik dan fleksibel via NorthBound Interface (NBI).

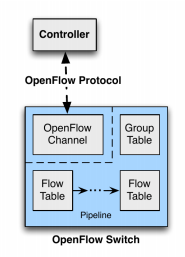
1. Layer Kontrol (*Control Layer*)

Layer Kontrol menyediakan fungsionalitas kontrol secara terpusat yang mengawasi perilaku jaringan forwarding melalui open interface.

1. Layer Infrastruktur (*Infrastructure Layer*)

Layer infrastruktur terdiri dari elemen-elemen jaringan dan perangkat keras yang menjalakan fungsi packet switching dan forwarding.

### OpenFlow

OpenFlow merupakan protokol utama pada jaringan SDN untuk menghubungkan controller layer dan infrastructure layer akibatnya dimungkinkan untuk melakukan pengaturan routing serta pengiriman paket melalui switch. Hal ini bisa terjadi karena switch hanya berfungsi untuk meneruskan paket melalui suatu port tanpa mampu membedakan tipe protokol data yang sedang dikirimkan. Openflow juga memungkinkan untuk mengakses serta memanipulasi forwarding plane secara langsung dari perangkat-perangkat jaringan seperti switch dan router baik secara fisik maupun virtual (Izzatul Ummah, 2016).

Gambar 4.2 Arsitektur OpenFlow

Gambar 4.2 menunjukkan arsitektur OpenFlow yang terdiri dari tiga bagian yaitu :

1. Controller

Controller adalah aplikasi SDN yang mengelola flow control untuk mengaktifkan intelligence networking yang bekerja berdasakan protokol seperti OpenFlow sehingga memungkinkan server memberitahu kemana paket dikirimkan. Perangkat meneruskan paket data yang diterima berdasarkan aturan yang diterapkan dari controller (Anggara, 2015).

|  |  |  |
| --- | --- | --- |
| **Nama** | **Interface** | **Versi OpenFlow** |
| NOX | C++ | 1.0 |
| POX | Python | 1.0 |
| Beacon | Java | 1.0 |
| Floodlight | REST, Java | 1.0, 1.3 |
| OpenDayLight | REST, OSGi | 1.0, 1.3 |
| Ryu | Python, REST, RPC | 1.0, 1.2, 1.3, 1.4 |
| FlowVisor | OpenFlow | 1.0 |

Tabel 4.1 Daftar Controller SDN

Tabel 4.1 menampilkan informasi mengenai Controller SDN yang tersedia saat ini termasuk bahasa pemrograman yang digunakan untuk membangunnya dan versi OpenFlow yang dapat berjalan.

Proyek akhir ini menggunakan POX yang merupakan kontroler pertama yang diciptakan untuk Openflow di Standford mUniversity enggunakan bahasa pemrograman Python. Alasan penggunaan POX dikarenakan kontroler ini mudah, ringan, dan cepat untuk penelitian yang tidak membutuhkan header fields tambahan.

1. OpenFlow Switch

OpenFlow Switch terdiri dari dua bagian sebagai berikut :

1. Sebuah flow table menyimpan segala aturan yang berkaitan dengan bagaimana sebuah paket harus dialirkan. Controller dapat merubah, menambah, dan menghapus aturan tersebut yang dinamakan sebagai flow entries.
2. Sebuah OpenFlow Channel merupakan saluran yang aman digunakan untuk menghubungkan switch dengan controller.
3. OpenFlow Protocol

OpenFlow protocol merupakan penyedia standar komunikasi terbuka antar Controller dengan Switch. OpenFlow Protocol menentukan ke interface manakah flow akan diterapkan dari flow table.

### RouteFlow

RouteFlow adalah sebuah proyek open source untuk menyediakana layanan virtual IP routing pada perangkat yang mengaktifkan OpenFlow. Proyek ini terbentuk atas penggabungan OpenFlow dengan routing engine Quagga. RouterFlow terdiri dari tiga komponen yaitu sebagai berikut:

1. RFServer adalah standalone application yang menjadi pusat kontrol jaringan dengan cara mengatur virtual machine yang berjalan pada RFClient dan mengatur logic processnya yaitu event processing, VM mapping, resource management.
2. RFProxy adalah kontroler POX yang bertugas meneruskan kebijakan protokol seperti update route dan konfigurasi dari RFServer ke dataplane.
3. RFClient adalah daemon pada VM yang bertugas mendeteksi perubahan informasi routing dan memberitahunya ke RFServer.

### Routing Information Protocol (RIP)

Routing Information Protocol (RIP) adalah salah satu jenis protokol routing yang menggunakan routing distance vector sebagau referensi dalam penentuan jalurnya. RIP efektif diguankan dalam jaringan skala kecil dan memiliki karakteristik utama sebagai berikut :

1. Perhitungan hop digunakan sebagai metric untuk memilih jalur.
2. Jika jumlah hop pada suatu jaringan lebih dari lima belas hop, maka RIP tidak dapat menampung rute pada jaringan tersebut.
3. Routing update di broadcast atau multicast setiap tiga puluh detik, timeout timer setiap 180 detik, dan garbage-collection timer setiap 120 detik.

### Parameter Quality of Service

Quality of Service (QoS) adalah efek kolektif dari kinerja layanan yang menentukan kepuasan pengguna terhadap sebuah jaringan. Salah satu hal yang perlu diperhatikan untuk mengukur QoS adalah performa kerja yang terdiri dari uji latency dan uji throughput. Namun penelitian ini juga ditambahkan dua variable tambahan yang akan diamati untuk mendukung penilaian performa kerja yakni variable jitter dan packet loss.

1. **Latency**

Latency didefinisikan sebagai interval waktu antara rangsangan dan tanggapan dari beberapa perubahan fisik dalam sistem yang diamati(). Untuk dapat mencari nilai latency digunakan persamaan sebagai berikut:

Standarisasi latency yang dibuat oleh Telecomunication Internet Protocol Harmonization Over Network (TIPHON) dapat dilihat pada tabel 4.1 berisi mengenai indeks dan kategori latency yang menentukan kualitas layanan.

|  |  |
| --- | --- |
| **Kategori Latency** | **Latency** |
| Sangat Bagus | < 150ms |
| Bagus | 150ms s/d 300ms |
| Sedang | 300ms s/d 450ms |
| Jelek | >450ms |

Tabel 4.1 Performa Jaringan berdasarkan Latency (TIPHON)

1. **Throughput**

Throughput adalah jumlah bit yang sukses diterima dari suatu titik jaringan ke titik jaringan lainnya dibandingkan dengan total waktu pengiriman(). Nilai throughput didapatkan menggunakan persamaan berikut :

Standarisasi throughput yang dibuat oleh Telecomunication Internet Protocol Harmonization Over Network (TIPHON) dapat dilihat pada tabel 4.2 tentang indeks dan kategori throughput yang menentukan kualitas layanan.

|  |  |
| --- | --- |
| **Kategori Throughput** | **Throughput** |
| Sangat Bagus | 100% |
| Bagus | 75% |
| Sedang | 50% |
| Jelek | < 25% |

Tabel 4.2 Performa Jaringan berdasarkan Throughput (TIPHON)

1. **Jitter**

Jitter adalah variasi delay yang diakibatkan oleh panjang antrian dalam suatu pegolahan data dan reassemble paket data di akhir pengiriman akibat kegagalan sebelumnya(). Besarnya nilai jitter dipengaruhi oleh beban trafik dan tumbukan antar pake (congestion) yang ada pada jaringan. Semakin besar beban trafik, maka semakin besar peluang terjadi congestion, sehingga nilai jitter akan semakin besar yang menandakan nilai QoS semakin turun.Nilai jitter didapatkan dari persamaan berikut :

Pada standarisasi menurut Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON), diberitahukan mengenai indeks dan kategori berdasar nilai jitter yang menentukan kualitas layanan seperti terlihat pada Tabel 4.3

|  |  |
| --- | --- |
| **Kategori Jitter** | **Peak Jitter** |
| Sangat Bagus | 0 ms |
| Bagus | 75 ms |
| Sedang | 125 ms |
| Jelek | > 225 ms |

Tabel 4.3 Performa Jaringan berdasarkan Jitter (TIPHON)

1. **Packet Loss**

Packet Loss Ratio (PLR) adalah jumlah paket yang hilang pada saat pengirian paket data ke tujuan. Semakin kecil nilai packet loss, maka semakin baik kualitas jaringan tersebut (). Nilai packet loss didapatkan dari persamaan berikut :

Secara umum terdapat empat kategori penurunan performansi jaringan berdasarkan nilai packet loss menurut Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) seperti pada Tabel 4.4

|  |  |
| --- | --- |
| **Kategori** | **Packet Loss** |
| Sangat Bagus | 0 % |
| Bagus | 3 % |
| Sedang | 15 % |
| Jelek | > 25 % |

## HIPOTESIS

Berdasar pada sub bab terdahulu, seperti perumusan masalah, tinjauan pustaka, dan landasan teori, maka dapat dikemukakan hipotesisnya yaitu sebagai berikut:

1. Routing Protocol RIP dapat diimplementasikan dalam jaringan SDN dengan perangkat raspberry pi B3 menggunakan kontroler POX
2. Kinerja raspberry pi B3 sebagai OpenFlow Switch tidak jauh berbeda dengan simulasi Mininet

**BAB V**

**CARA PENELITIAN**

## Alat dan Bahan

Bab ini menjelaskan proses perancangan untuk terbentuknya simulasi dan implementasi pada analisis kinerja routing protokol RIP pada jaringan SDN yang terdiri dari peralatan dan bahan baik berupa perangkat keras (hardware) ataupun perangkat lunak (software).

### Peralatan

Adapun peralatan yang digunakan dalam penelitian ini diantaranya sebagai berikut:

1. Laptop sebagai Controller dengan spesifikasi:

* CPU : 2.00 GHz
* Hardisk : 500 GB
* RAM : 4 GB
* OS : Windows 10

1. PC sebagai host sebanyak 4 unit dengan spesifikasi:

* CPU : 2.00 GHz
* Hardisk : 500 GB
* RAM : 2 GB
* OS : Windows 10

1. Raspberry Pi 3 Model B sebagai OpenFlow Switch sebanyak 4 unit dengan spesifikasi:

* CPU : 1.2 GHz quad-core ARM
* Memory : 1 GB LPDDR2-900 SDRAM
* USB Port : 4
* Network : 10/100 Mbps Ethernet, 802.11 n Wireless LAN

1. USB to Ethernet Converter sebanyak 10 unit
2. Switch sebanyak 1 unit
3. Access Point sebanyak 1 unit
4. Kabel LAN sebanyak 14 unit

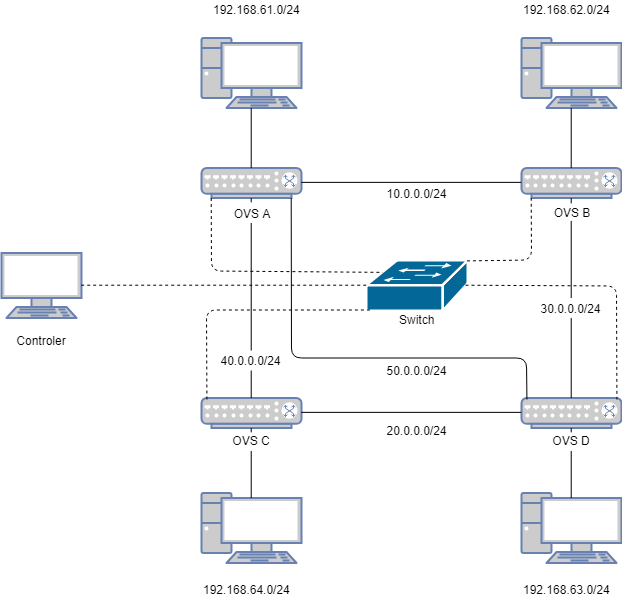
### Bahan

Perangkat lunak opensource dan freeware yang digunakan adalah sebagai berikut:

1. Ubuntu Server 12.04
2. Mininet
3. Oracle VM Virtual Box
4. POX dan RouteFlow Controller
5. Wireshark
6. D-ITG dan Iperf

## Prosedur Penelitian

1. Rancangan Topologi

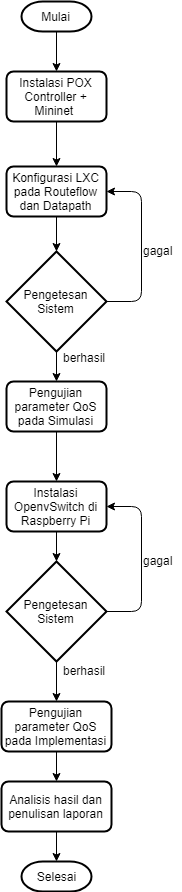


Gambar 5.1 Rancangan Topologi Jaringan

Rancangan topologi jaringan penelitian ini seperti yang ditunjukkan pada gambar 5.1 menggunakan sebuah laptop sebagai controller, empat buah Raspberry Pi 3 Model B sebagai OpenFlow Switch, dan empat buah PC atau Laptop sebagai host.

1. Metode Penelitian

Metode penelitian yang dilakukan dapat dilihat pada flowchart berikut :



Gambar 5.2 Flowchart Metode Penelitian

## Analisis Hasil

Sebelum dilakukan analisis hasil terlebih dahulu dilakukan perancangan dan implementasi routing RIP pada jaringan SDN menggunakan software emulator Mininet dan perangkat Raspberry Pi B3 yang dijadikan sebagai OpenFlow Switch. Selanjutnya data QoS yang meliputi meliputi latency, throughput, jitter, dan packet loss diambil dari hasil pengiriman paket antara 2 host melalui tools aplikasi iPerf. Data-data tersebut kemudian diolah dan disajikan dalam bentuk grafik dan akan dihasilkan sebuah analisa perbandingan pada setiap skenario yang diberi perlakukan sesuai dengan parameter pengukuran yang telah ditentukan. Hasil dari analisis tersebut diharapkan dapat memberikan informasi mengenai performa kinerja routing RIP pada jaringan SDN yang menggunakan Raspberry Pi B3 sebagai switch OpenFlow dan dapat memenuhi tujuan dan manfaat proyek akhir yang telah dituliskan.

## Jadwal Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahap Kegiatan | Bulan | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | 2 | | | | 3 | | | | 4 | | | | 5 | | | | 6 | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| **Persiapan** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Identifikasi Masalah |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan Topologi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Seminar Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Pelaksanaan** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Simulasi** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Install VM Virtual Box |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Install Ubuntu Server 14.04 + Controller POX |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Install Mininet emulator |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Konfigurasi OpenFlow |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian RIP pada simulasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Implementasi** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Instal OpenFlow pada Raspberry Pi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pemasangan Jaringan dan Konfigurasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengujian RIP pada implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Penyelesaian** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analisis Hasil |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan Laporan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Submit Publikasi (Seminar) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyelesaian Laporan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |