

BAB I

PENDAHULUAN

1.1 Latar Belakang

Software Defined Network (SDN) adalah konsep pendekatan baru untuk mendesain, membangun, dan mengelola jaringan komputer dengan memisahkan jalur kontrol (*control plane*) dan jalur data (*data plane*). Pemisahan yang dimaksud adalah pemisahan *router* atau *switch* yang ada pada *data plane* yang biasanya berfungsi untuk *forwarding* paket data dan memberikan tanggung jawab kepada *software* tertentu pada *control plane* sebagai *controller* yang secara logis terpusat untuk mengontrol perilaku sebuah jaringan [1]. *Controller* SDN yang bersifat *programmable* memungkinkan untuk mengimplementasikan suatu *routing* pada SDN.

Routing adalah proses penentuan rute dari *host* asal ke *host* tujuan. *Routing* menentukan kemana datagram akan dikirim agar mencapai tujuan yang diinginkan. Ada 2 metode *routing* yang dapat diterapkan dalam membangun suatu jaringan yaitu *single-path routing* dan *multipath routing*. *Multipath routing* merupakan metode *routing* menggunakan beberapa jalur yang tersedia. Berbeda dengan *single-path routing* yang hanya menggunakan satu jalur dari sumber ke tujuan. Pada algoritma *single-path routing*, keseluruhan jalur yang ada pada topologi jaringan yang disebut dengan kemampuan *multipath* tidak dapat sepenuhnya dimanfaatkan [2]. Dengan *multipath routing* topologi jaringan yang memiliki cabang redundan dapat dimanfaatkan sepenuhnya. Selain itu, *multipath routing* dapat mengurangi kemacetan pada jaringan dengan melakukan *load-balancing* sehingga *traffic* jaringan dapat didistribusikan dengan memilih jalur yang memiliki sumber daya tersedia paling besar [3].

Tujuan utama dari SDN sendiri adalah untuk mempermudah penerapan berbagai macam aplikasi jaringan yang dapat diprogram pada *controller*-nya, karena sifat *controller* SDN yang *programmable* tersebut. *Ryu* merupakan sebuah *framework* berbasis komponen SDN. *Ryu* menyediakan komponen *software* dengan API yang membudahkan *developer* untuk membangun aplikasi-aplikasi control dan manajemen jaringan [4]. Pengembangan aplikasi untuk *Ryu* dapat dilakukan dengan menggunakan Bahasa *Python*. *Ryu* mendukung berbagai protokol untuk

memanajemen jaringan antara lain *OpenFlow*. Kebutuhan atas *Ryu* sebagai *OpenFlow controller* adalah karena *Ryu* mendukung *OpenFlow* versi 1.0 hingga 1.5.

Algoritma Dijkstra adalah salah satu varian dari algoritma *greedy*, yaitu salah satu algoritma untuk pemecahan persoalan yang terkait dengan masalah optimasi yang cukup populer. Sesuai artinya *greedy*, yaitu rakus atau tamak tapi bukan dalam konteks yang negatif, algoritma ini hanya memikirkan solusi apa yang terbaik untuk dilakukan pada setiap langkahnya dan dipakai oleh pemrograman dinamis, jika solusi total optimal, maka bagian dari solusi sampai pada suatu tahap nilai optimum global [5]. Fungsi objektif algoritma Dijkstra akan memaksimal atau meminimalkan nilai solusi. Tujuannya adalah memilih satu saja solusi terbaik dari masing-masing anggota himpunan solusi (himpunan *node*).

Pada penelitian sebelumnya, membahas mengenai manajemen *Quality of Service* (QoS) pada SDN dengan menggunakan pendekatan rekayasa trafik ketika terjadi kemacetan pada jaringan. Algoritma yang diterapkan untuk mengatasi permasalahan tersebut berbasis algoritma Dijkstra. Pengujian dilakukan secara emulasi dan *real time*. Pengujian berkaitan dengan QoS dengan parameter seperti *packet loss*, *throughput*, dan *jitter* [6].

Pada Universitas Trunojoyo Madura (UTM) belum menggunakan SDN dalam pengontrolan jaringan, permasalahan yang timbul jika terjadi *error* maka perbaikan dilakukan konfigurasi pada setiap perangkat secara langsung satu persatu. Dengan adanya SDN maka konfigurasi terpusat yang dapat dilakukan lebih cepat dan efisien. Topologi UTM yang digunakan pada penelitian ini untuk melakukan pengujian *single-path* dan *multipath routing*. Pada topologi memungkinkan untuk dilakukan pengujian tanpa membuat topologi baru.

Berdasarkan latar belakang tersebut penelitian ini membahas tentang Analisis *Multipath Routing* Berbasis Algoritma Dijkstra Pada *Software Defined Network* Menggunakan *Ryu Controller*, dimana untuk menganalisa kinerja *multipath routing* menggunakan algoritma Dijkstra dan didasarkan pada parameter uji *Quality of Service* (QoS) meliputi *throughput*, *jitter*, dan *packet loss*.

1.2 Perumusan Masalah

1.2.1 Permasalahan

Berdasarkan latar belakang diatas, dapat dirumuskan suatu permasalahan penentuan rute yang akan dilalui untuk melakukan *multipath routing*. Bagaimana kualitas layanan QoS pada jaringan SDN dengan melakukan skenario jaringan *Single-path* dan *Multipath Routing*. Melakukan perbandingan *throughput*, *jitter*, dan *packet loss* untuk menghitung prosentase pada *single-path routing* dengan *multipath routing*.

1.2.2 Metode Usulan

Berdasarkan permasalahan diatas metode algoritma Dijkstra untuk melakukan pencarian rute terpendek harus dimodifikasi agar dapat melakukan *multipath routing*. Dan juga melakukan pengujian kelayakan sebuah jaringan menggunakan metode QoS yang memiliki beberapa parameter pengujian meliputi *throughput*, *jitter*, dan *packet loss*. Serta melakukan perbandingan prosentase *throughput* yang diperoleh pada *single path* dengan *multipath routing*.

1.2.3 Pertanyaan Peneliti

Berdasarkan latar belakang diatas ada beberapa pertanyaan peneliti sebagai berikut :

1. Berapa prosentase perbandingan *throughput*, *jitter*, dan *packet loss* pada *single-path* dengan *multipath routing* ?
2. Apakah Algoritma Dijkstra sudah mampu melakukan penentuan jalur pada topologi ?

1.3 Tujuan dan Manfaat

1.3.1 Tujuan

Adapun tujuan yang ingin dicapai melalui penelitian ini adalah untuk memberikan perbandingan prosentase *throughput*, *jitter*, dan *packet loss* yang di peroleh dari *Single-path* terhadap *Multipath Routing*.

1.3.2 Manfaat

Dari hasil penelitian ini diharapkan dapat digunakan sebagai salah satu algoritma *alternative* untuk meningkatkan kinerja jaringan pada SDN dan membantu pengembang jaringan agar lebih efektif lagi di kemudian hari.

1.4 Batasan Masalah

Dengan adanya batasan masalah maka dapat lebih disederhanakan lagi dan terarah agar penelitian tidak menyimpang dari apa yang diteliti. Adapun batasan masalahnya sebagai berikut:

1. Penerapan jaringan SDN berbasis protokol OpenFlow secara emulasi.
2. Penelitian lebih difokuskan kepada perancangan dan pengujian sebuah jaringan *Single-path* terhadap *Multipath Routing*.
3. Sistem dirancang menggunakan Mininet *emulator*.
4. Bahasa pemrograman menggunakan Python dengan Ryu *controller*.
5. Menggunakan *ipref tools* mengukur performa QoS jaringan SDN.
6. Pengukuran pada *Protocol* UDP (*User Datagram Protocol*).
7. Skenario yang digunakan algoritma Dijkstra pada topologi kampus Universitas Trunojoyo Madura. Ryu *controller* akan terhubung dengan 25 switch dan 100 *host*.

1.5 Sistematika Penulisan

Sistematika penulisan pada skripsi ini terdiri dari beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang mengenai ruang lingkup masalah yang dibahas, solusi yang diusulkan untuk dapat menyelesaikan permasalahan, tujuan dan manfaat dengan diterapkannya solusi yang diusulkan, batasan solusi yang usulkan, serta sistemapenulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisikan teori yang mendukung dalam perancangan jaringan SDN (*Software Defined Network*), algoritma Dijkstra, Ryu *Controller*, dan penelitian sebelumnya.

BAB III METODOLOGI USULAN

Bab ini berisi metode usulan dan perancangan sistem yang dilakukan dalam simulasi *multipath routing* pada SDN.

BAB IV IMPLEMENTASI PROGRAM

Bab ini berisi pelaksanaan pengujian *multipath routing* pada SDN yang telah disimulasikan dan diperoleh hasil pengujian kemudian dilakukan analisis.

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran pengembangan dai keseluruhan tahapan pelaksanaan penelitian ini.