

BAB 1

PENDAHULUAN

1.1 Latar belakang

Software-Defined Networking (SDN) merupakan sebuah teknologi dalam jaringan dimana *Control Plane* terpisah dari *Data Plane*. *Control Plane* berfungsi untuk mengatur bagaimana suatu paket dikirimkan pada suatu jaringan, sedangkan *Data Plane* merupakan perangkat yang menjalankan kegiatan penerusan atau pengiriman paket berdasarkan informasi yang diberikan *control plane* (Kreutz, et al., 2015). Komunikasi antara *control plane* dan *data plane* melalui protokol *openflow*. *Protocol openflow* merupakan *protocol* yang paling banyak digunakan untuk SDN dan *protocol openflow* dijadikan sebagai suatu standar *protocol* komunikasi antara perangkat *data plane* dan *control plane* (McKeown, et al., 2008). Hal ini membuat SDN menjadi solusi untuk teknologi jaringan masa depan karena dapat digunakan untuk komunikasi jaringan yang berbeda. Konsep SDN juga telah menyederhanakan konsep jaringan yang ada sekarang ini, dimana kontrol jaringan pada sebuah *controller* dapat membuat suatu jaringan mudah diatur dan lebih fleksibel, hal tersebut dikarenakan pada SDN sebuah *controller* bersifat *programmable* (Kreutz, et al., 2015). SDN dapat melakukan berbagai hal di dalam *controller* mulai dari pengaturan ataupun *monitoring* jaringan. SDN digunakan untuk mengatasi masalah pada jaringan komputer yaitu konfigurasi *real-time*. Seperti yang diutarakan oleh MPLS (*Multi Protocol Label Switching*) secara teoritis membahas masalah *routing* dengan kemampuan *traffic engineering* (TE). Tujuan utama dari SDN sendiri adalah untuk mempermudah penerapan berbagai jenis macam aplikasi jaringan yang dapat diprogram pada *controller*-nya, karena sifat *controller* SDN yang *programmable* tersebut. Macam-macam contoh aplikasi jaringan yang dapat diterapkan pada SDN seperti *Load balancing*, *Network Virtualization*, *Intrusion Detection*, *Routing*, dan lain-lain sebagainya (McKeown, et al., 2008).

Algoritme *routing single-path* atau satu jalur konvensional, seperti algoritme *Dijkstra* pada OSPF (*Open Shortest Path First*), tidak dapat sepenuhnya memanfaatkan keseluruhan jalur yang ada pada topologi jaringan, atau yang bisa disebut kapabilitas *multipath*. Pada algoritme-algoritme *spanning tree* seperti *Dijkstra*, topologi jaringan akan selalu dikurangi dan dipotong sehingga berbentuk *tree*. Hal tersebut akan menghapus kemampuan *multipath* pada topologi yang memiliki banyak cabang sehingga menyebabkan kepadatan jaringan (Lei, Wang, & Hsu, 2015).

Dengan *multipath routing*, *traffic* yang akan menuju suatu tujuan akan dibagi ke beberapa jalur untuk tujuan tersebut. Oleh karena itu, *multipath routing* diusulkan sebagai alternatif untuk *single-path routing* agar dapat sepenuhnya

memanfaatkan kemampuan *multipath* pada topologi jaringan. Pemilihan jalur yang hanya menggunakan satu jalur pengiriman saja hanya akan mengakibatkan *bottleneck* pada jalur pengiriman yaitu pengurangan performa dari jalur tersebut, oleh karena itu mencari jalur K menggunakan *Yen algorithm multipath routing* pada *software defined network* akan memberikan alternatif jalur lain karena disini bisa menentukan Jumlah K yang akan digunakan untuk sebagai media pengiriman sebanyak K- jalur yang diinginkan Yen, Jin Y. (Jul 1971), metode ini akan mengatasi permasalahan kemacetan pada jaringan. Selain itu, *routing* biasanya dilakukan untuk memilih jalur terpendek.

Penelitian sebelumnya tentang *Multipath routing dengan load balancing* pada *openflow software defined network* yang pernah dilakukan oleh Syahidillah, W., M. dari Universitas Brawijaya pada 2017 lalu. Pada penelitiannya dilakukan implementasi dengan menggunakan algoritme DFS(*Depth first search*) tersebut yang penerapannya pada jaringan *OpenFlow*. Dimana parameter yang diuji meliputi Bandwidth, throughput dan pencarian rute. Dari penelitian yang sudah ada, penulis berminat untuk mengimplementasikan dan kemudian melakukan perbandingan dari dua macam Algoritme *Shortest Path* DFS dan *Yen Algorithm* serta melakukan perbandingan dengan algoritme Dijkstra (*Single Path*).

Berdasarkan permasalahan *routing* yang telah dijelaskan diatas, maka penulis akan melakukan implementasi *Yen algorithm mulipath routing* pada *Software Defined Network* untuk mengatasi kekurangan dari OSPF. Penelitian ini akan menggunakan *Yen algorithm* untuk mencari beberapa jalur terpendek dan menggunakan parameter *cost* untuk memilih jalur yang akan digunakan sebagai jalur pengiriman data. *Controller* yang digunakan pada penelitian ini adalah Ryu. *Ryu* merupakan *controller* dengan bahasa pemograman *Phyton*. Dengan itu diharapkan dapat mengembangkan protokol *routing* yang dapat memanfaatkan kemampuan *multipath* dari jaringan dengan jalur yang ada pada topologi.

1.2 Rumusan masalah

Berdasarkan latar belakang dapat dirumuskan permasalahan sebagai berikut.

1. Bagaimana menemukan keseluruhan jalur menggunakan *Yen algorithm* pada jaringan *OpenFlow*?
2. Bagaimana analisis kinerja dari *Yen algorithm* pada jaringan *Openflow*?

1.3 Tujuan

Penelitian ini bertujuan untuk menghasilkan protokol *multipath routing* yang dapat menemukan jalur terpendek untuk menemukan jalan yang terbaik

yang dapat dilalui pada jaringan dan kemudian, melanjutkan untuk menemukan K-1 penyimpangan dari jalur terbaik berdasarkan jalur yang telah ditemukan pada jaringan dengan menggunakan protokol komunikasi SDN *OpenFlow*.

1.4 Manfaat

Adapun manfaat penelitian yang diharapkan adalah sebagai berikut.

1. Manfaat bagi penulis

Dapat meningkatkan pengetahuan teori maupun aplikatif mengenai SDN *OpenFlow* terhadap mengaplikasikan apapun yang menjadi bidang keilmuan maupun bakat kompetensi baik secara *hard skill* maupun *soft skill* sampai memiliki *result* yang jelas.

2. Manfaat bagi peneliti lain

Dapat menjadi acuan untuk penelitian tentang *multipath routing* dengan *Yen algorithm* pada *OpenFlow* SDN.

3. Manfaat bagi jaringan

1. Dapat mengoptimalkan utilitas dari suatu jaringan.
2. Dapat meningkatkan keamanan dalam pengiriman data pada jaringan.
3. Dapat mengetahui jalur yang terbaik.

1.5 Batasan masalah

Agar pembahasan masalah dapat terarah dengan baik dan tidak menyimpang dari pokok permasalahan, maka penulis membatasi permasalahan yang akan dibahas, yakni:

1. Implementasi SDN yang digunakan adalah *OpenFlow*.
2. Menggunakan *Ryu Controller* sebagai *controller* SDN sehingga bahasa pemrograman yang digunakan adalah *Python*.
3. Menggunakan emulasi jaringan *Mininet*.
4. Menggunakan Topologi *Abilene*.
5. Parameter *cost* digunakan sebagai *K-path* pengiriman data.

1.6 Sistematika pembahasan/laporan

Struktur penulisan pada masing-masing bab adalah sebagai berikut.

BAB 1 PENDAHULUAN

Pada bab ini mencakup latar belakang, rumusan masalah, tujuan, manfaat yang akan diperoleh, batasan masalah dan sistematika penulisan.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi kajian tentang penelitian yang dilakukan sebelumnya dan teori serta konsep yang didapatkan dari sumber yang relevan untuk dijadikan sebagai panduan dalam penyusunan laporan skripsi.

BAB 3 METODOLOGI

Pada bab ini membahas tentang metodologi yang digunakan pada penelitian ini. Di sini berisi langkah-langkah kerja yang terdiri dari studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian dan analisis dan kesimpulan

BAB 4 REKAYASA KEBUTUHAN

Pada bab ini berisi penjelasan tentang gambaran umum sistem dan kebutuhan sistem dalam penelitian ini dengan lebih detail.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini berisi tentang tahapan perancangan dan implementasi yang dilakukan untuk membangun sistem *multipath routing* dengan *Yen algorithm* pada *OpenFlow* SDN.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini berisi pengujian dan analisis sistem *multipath routing* dengan *Yen Algorithm* dan DFS pada *OpenFlow* SDN yang telah diimplementasikan.

BAB 7 PENUTUP

Pada bab ini berisi kesimpulan yang telah diambil dari penelitian dan perancangan sistem, serta saran untuk pengembangan selanjut agar bisa dilakukan perbaikan di masa yang akan datang.