BAB I PENDAHULUAN

1.1 Latar belakang

Load balancing adalah sebuah mekanisme untuk membagikan atau mendistribusikan beban yang masuk terhadap suatu layanan ke beberapa server secara seimbang (Gustin Anggraeni, Sukiswo, & Ajub Ajulian Zahra, 2014). *Load balancing* akan memberikan beban yang seimbang ke setiap server yang ada dengan parameter yang dapat disesuaikan. Pada pengujian ini akan membandingkan algoritme *shortest delay* dan algoritme *round-robin* dalam pendistrbusian beban dalam menentukan jalur ke server. *Load balancing* juga contoh aplikasi jaringan yang dapat diterapkan pada SDN. Beberapa aplikasi jaringan yang dapat diterapkan pada SDN seperti *Network Virtualization, Intrusion Detection, Routing* dan *Load Balancing* (Kurniawati, 2016).

Software Defined Networking (SDN) merupakan sebuah jaringan komputer dimana *control plane* terpisah dari *data plane*. *Control plane* bertugas mengatur bagaimana suatu paket dikirimkan pada suatu jaringan sedangkan *Data plane* bertugas meneruskan paket yang masuk ke port menuju port tujuan berdasarkan informasi yang diberikan control plane (Al-Najjar, 2016). 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. Salah satu perwujudan dari konsep SDN yang sudah ada adalah dengan hadirnya beberapa macam protocol yang sudah digunakan untuk SDN diantaranya *NETCONF, OVSDB, MPSLS-TP, Opeflow*. Dari macam-macam *protocol* tersebut *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). Konsep SDN juga telah menyederhanakan konsep jaringan yang ada sekarang ini, dimana kontrol jaringan pada sebuah *controller* yang membuat suatu jaringan mudah diatur dan lebih fleksibel, hal tersebut dikarenakan pada SDN sebuah *controlle*r bersifat *programmable*, yang menjadikan jaringan dapat diatur sesuai dengan kebutuhan (Kreutz, et al., 2015). 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. Dengan software defined networking, suatu jaringan dapat dipandang secara tersentralisasi sehingga dapat dengan jelas mengetahui kondisi dari jaringan.

Dalam konteks jaringan bermodel client-server, terdapat beberapa jenis algoritma dalam mendistribusikan beban kepada *web server*. Salah satunya adalah *round robin*. *Load balancing round robin* mendistribusikan permintaan kepada we*b server* secara bergantian dan dapat diberikan *weight* sesuai dengan *resource* yang dimiliki. Pada penggunaan algoritma *round robin*, saat *user* mengakses halaman *web server* pertama akan diarahkan ke *server* utama, permintaan *user*

yang selanjutnya akan diberikan kepada *server* yang lainnya (Mustafa, 2017). Akan tetapi, algoritme ini memiliki kelemahan yaitu ketika salah satu server diberikan beban yang besar maka otomatis delay pun semakin meningkat sehingga pada algoritme *round-robin* akan tetap melewati delay yang besar dan tidak memperhatikan beban menuju server, hal ini disebabkan karena algoritme *round-robin* tidak membebankan traffic ke setiap server dan membagi beban ke server secara silkular.

Berdasarkan penjelasan diatas algoritme tersebut tidak memperhatikan *delay*. *Delay* merupakan waktu yang dibutuhkan data untuk menempuh jarak dari *source* ke *destination* (Kurose, 2013). Semakin besar delay pada jaringan maka akan membuat semakin tinggi resiko terjadinya kegagalan komunikasi data pada jaringan tersebut. Paket yang dikirimkan akan diberikan waktu tunggu *(time out).* Bila jeda waktu paket di kirim dan melebihi dari durasi *time out* maka paket tersebut akan di drop. Sehingga *delay* merupakan aspek penting dalam pemilihan jalur menuju server pada load balancing.

Berdasarkan permasalahan diatas yang telah dijelaskan, maka fokus dari penelitian ini adalah untuk mengimplementasikan algoritme *shortest delay* dengan mengalokasikan client menuju server yang memiliki jalur delay terendah. Oleh karena itu, penulis melakukan implementasi algoritme *shortest delay* untuk mekanisme *load balancing* pada *openflow software defined network*. Diharapkan penelitian ini dapat membantu dalam perkembangan mekanisme *load balancing* pada *openflow* kedepannya.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dikemukakan, rumusan masalah penelitian ini sebagai berikut:

1. Bagaimana hasil pendistribusian *request* client ke server dengan algoritme *shortest delay* pada *software defined network* dengan beberapa parameter uji?
2. Bagaimana analisis *performa* load balancing pada algoritme shortest delay menggunakan parameter *delay* pada *software defined network* dengan membandingkan algoritme *naive load balancing*, seperti algoritme *round-robin*?

1.3 Tujuan

Penelitian ini bertujuan untuk menghasilkan algoritme *shortest delay* berdasarkan jalur yang memiliki delay terkecil menuju server yang terjadi pada setiap jalur yang ada pada topologi jaringan.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut.

- Bagi peneliti :

Dapat menjadi referensi untuk penelitian selanjutnya tentang algoritme *shortest delay* pada *software defined network*.

- Bagi penulis :
  Dapat mengasah keterampilan dalam administrasi jaringan khususnya jaringan *Openflow.*

1.5 Batasan masalah

Agar pembahasan masalah dapat dilakukan terarah dan dapat menghasilkan yang diharapkan, maka penulis membatasi permasalahan yang akan di bahas, sebagai berikut:

1. Fokus terhadap implementasi load balancing web server dengan algoritme shortest delay pada SDN
2. Menggunakan Ryu Controller dan bahasa pemrograman python
3. Menggunakan simulator mininet sebagai media web server
4. Mengabaikan server yang overload
5. Menggunakan data *round trip time*
6. Melakukan pengujian performa sistem *load balancing web server* pada *software defined network* dengan menangani banyaknya permintaan *client* ke *web server* dan menggunakan jumlah *rate* tertentu.

1.6 Sistematika pembahasan

Uraian singkat mengenai metodologi penelitian pada masing masing-masing bab adalah sebagai berikut:

BAB I PENDAHULUAN
Bab ini mencakup latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dari "Analisis dan Implementasi Load Balancing pada Server dengan Algoritme Shortest Delay Pada Software Defined Networking".

BAB II LANDASAN KEPUSTAKAAN
Bab ini akan membahas kajian pustaka dan landasan teori tentang penelitian serta penyusunan laporan.

BAB III METODE PENELITIAN
Bab ini menguraikan tentang langkah kerja yang terdiri dari studi literatur, perancangan sistem, impementasi dan analisis serta pengambilan kesimpulan.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi *load balancing shortest delay pada software defined network* dengan menggunakan *ryu Controller* dan *mininet* sebagai simulasi penelitian.

BAB V PENGUJIAN

Bab ini memuat proses pengujian dan analisis terhadap server yang dilakukan pada sistem load balancing software defined network.

BAB VI PENUTUP

Bab ini memuat kesimpulan dan saran yang diambil dari hasil pengujian sehingga dapat dilakukan perbaikan dan pengembangan dimasa yang akan datang.