

# Bab I

## Pendahuluan

### 1.1 Latar Belakang

*Software Defined Network* (SDN) merupakan arsitektur terbaru dalam jaringan yang melakukan pemisahan pada *control plane* dan *data plane* [1][2]. *Control plane* berfungsi untuk mengelola jaringan secara terpusat dan *real-time* dengan melakukan pemrograman pada sisi *controller*, sehingga dalam *forwarding* paket yang dilakukan oleh *switch* akan mengikuti aturan *flow table* pada *control plane* [1][2]. *Data plane* melakukan teknik *forwarding* yang diinterpretasikan sebagai *switch*, dan terdapat sebuah protokol *OpenFlow* sebagai *interface* yang menghubungkan antara *controller* dan *data plane* untuk berkomunikasi [1].

Pada topologi Telkom University, digunakan sebuah *router* sebagai fungsi *routing* dan *switch* sebagai fungsi aliran data, arsitektur seperti ini disebut inter-VLAN. Berdasarkan *interview* yang dilakukan, kelebihan menggunakan inter-VLAN pada topologi Telkom University yaitu konfigurasi yang mudah, namun terdapat kelemahan yaitu tidak adanya *backup link* jika suatu saat terjadi *link failure* dalam pengiriman paket data. Berdasarkan kelemahan pada topologi Telkom University, maka dibangun sebuah jalur *backup* ketika terjadi *link failure* untuk menjamin paket data tetap terkirim dari *source* menuju *destination*.

Pada saat terjadi *link failure* dibutuhkan sebuah algoritma *shortest path* untuk menentukan jalur terpendek sebagai jalur baru dalam pengiriman paket data [3]. Dalam SDN, terdapat suatu mekanisme yang dapat mengatasi *link failure* dengan mengkonfigurasi *controller* agar dapat berpindah ke *link* alternatif yang telah disediakan dengan bantuan *OpenFlow*, sehingga *controller* akan dengan otomatis menentukan jalur *backup* [4].

*Open Shortest Path First* (OSPF) merupakan teknik *routing* yang digunakan pada topologi Telkom University, menerapkan algoritma Dijkstra sebagai algoritma *shortest path* [5]. Selain algoritma Dijkstra terdapat juga algoritma Bellman-Ford yang merupakan algoritma *shortest path*. Algoritma Dijkstra dan algoritma Bellman-Ford merupakan *single-source path problem* [5][6], dimana dapat menemukan jalur terpendek dari satu *node* menuju ke semua *node* yang ada. Perbedaan pada kedua algoritma ini yaitu dalam penentuan jalur terpendek, algoritma Dijkstra menentukan jalur terpendek berdasarkan perhi-

tungan *cost* atau jarak antara dua buah *node*, sedangkan algoritma Bellman-Ford menentukan jalur terpendek berdasarkan perhitungan jumlah *hop* [7]. Namun apabila terdapat *metric routing* berupa *bandwidth*, dalam pencarian jalur akan memilih berdasarkan *bandwidth* tertinggi diantara kedua *node* [3].

Oleh sebab itu, dilakukan emulasi ketika terjadi *link failure* pada topologi Telkom University dengan memperhatikan performansi dari dua algoritma *shortest path* berbasis SDN. Emulasi adalah kegiatan replikasi yang dilakukan pada sebuah sistem dengan menggunakan lingkungan yang berbeda dari sistem sebenarnya dengan memperhatikan fungsi dan hubungannya [8]. *Metric routing* yang digunakan pada jalur dalam topologi adalah *bandwidth*. Hal ini dimaksudkan untuk melihat *controller* dapat menentukan jalur optimal yang dilalui berdasarkan *bandwidth* tertinggi [9].

Dengan melakukan analisa terhadap implementasi performansi kedua algoritma, diperoleh algoritma yang optimal dari segi parameter *delay* [10] untuk mengukur waktu tunda yang dibutuhkan suatu paket dalam pengiriman data dari sumber ke tujuan, *convergence time* [11] untuk menghitung berapa lama waktu yang dibutuhkan dalam pembentukan suatu jalur, dan *packet loss* [12] untuk mengukur jumlah paket yang hilang selama pengiriman paket dari sumber ke tujuan. Sehingga dari hasil uji parameter diatas dapat ditentukan algoritma *shortest path* yang optimal yaitu algoritma yang dapat menentukan jalur baru yang terpendek guna meminimalkan *cost* sebelum dan sesudah terjadi *link failure*.

## 1.2 Perumusan Masalah

Pemasalahan yang diangkat dalam Tugas Akhir ini antara lain:

1. Bagaimana cara mengatasi *link failure* pada topologi Telkom University?
2. Bagaimana cara menentukan jalur yang optimal sebelum dan sesudah terjadi *link failure* pada topologi Telkom University?
3. Bagaimana mengukur perbandingan performansi implementasi algoritma *shortest path* sebagai penentuan jalur yang optimal pada saat sebelum dan sesudah terjadi *link failure*?

## 1.3 Tujuan

Berdasarkan masalah tersebut, tujuan pengerjaan Tugas Akhir ini antara lain:

1. Mengemulasikan *backup link* pada topologi Telkom University.
2. Melakukan perbandingan performansi pada implementasi algoritma *shortest path* yaitu algoritma Dijkstra dan algoritma Bellman-Ford untuk menentukan jalur yang optimal pada topologi Telkom University.

3. Melakukan analisa terhadap performansi implementasi algoritma *shortest path* yaitu algoritma Dijkstra dan algoritma Bellman-Ford yang diterapkan berdasarkan parameter *delay*, *convergence time*, dan *packet loss*.

## 1.4 Batasan Masalah

Batasan masalah dalam pengerjaan Tugas Akhir ini antara lain:

1. Menggunakan Mininet sebagai *emulator* pada *data plane* untuk memuliskan topologi Telkom University.
2. Menggunakan Ryu sebagai *controller* pada *control plane*.
3. Menggunakan D-ITG untuk membangkitkan trafik dan mengontrol laju QoS
4. Topologi yang digunakan adalah topologi Telkom University.
5. Trafik data yang digunakan dalam pembangkitan trafik untuk mengukur *delay* dan *packet loss* adalah TCP.
6. *Metric routing* yang digunakan adalah *bandwidth*.
7. Parameter yang dipakai dalam melakukan analisis adalah *delay*, *convergence time*, dan *packet loss*.
8. Keseluruhan *host* pada topologi yang digunakan berada dalam *subnet* 10.0.0.0/8.
9. Pengambilan data untuk analisis performansi *delay* menggunakan *average delay*.
10. Trafik *background* yang dibangkitkan adalah *average* trafik dari keseluruhan trafik yang ada pada tiap jalur dengan nilai sebesar 1707060 byte/s.

## 1.5 Hipotesis

Berdasarkan paper [3] dijelaskan bahwa algoritma Dijkstra dari segi performansi *delay* dan dalam membentuk jalur baru ketika terjadi *link failure* memiliki nilai yang lebih kecil dibandingkan algoritma Bellman-Ford. Hal ini dikarenakan dalam pencarian jalur yang dimiliki oleh algoritma Dijkstra berdasarkan pada *cost* terkecil diantara dua buah *node*, dan juga waktu respon yang dimiliki algoritma Dijkstra lebih cepat daripada algoritma Bellman-Ford. *Cost* yang dimaksud merupakan nilai yang ditentukan oleh suatu *administrator* jaringan.

## 1.6 Rencana Kegiatan

Rencana kegiatan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

- Studi literatur

Pada tahap ini dilakukan pencarian literatur terkait dengan arsitektur jaringan SDN, topologi Telkom University, routing protokol, algoritma routing yang akan dilakukan penelitian terhadap Algoritma Bellman-Ford dan Algoritma Dijkstra.

- Pengumpulan Data

Pada tahap ini dilakukan untuk melakukan pengumpulan data terhadap hal-hal yang berkaitan dengan penelitian seperti pengumpulan data mengenai topologi Telkom University.

- Analisa dan perancangan sistem

Pada tahap ini akan dilakukan analisa secara mendalam terhadap performansi kedua algoritma shortest path ketika diterapkan pada arsitektur SDN dalam topologi Telkom University.

- Implementasi

Pada tahap ini akan dilakukan implementasi terhadap perancangan sistem dan analisa pada poin sebelumnya. Sistem berupa pembuatan program emulasi untuk melakukan analisa terhadap penerapan Algoritma Bellman-Ford dan Dijkstra dan melihat hasil performansi QoS yang dihasilkan pada arsitektur SDN dalam topologi Telkom University

- Analisa hasil implementasi

Pada tahapan ini akan dilakukan analisa hasil implementasi atau bahasa lainnya adalah pengujian. Pengujian akan dilakukan sesuai dengan skenario yang dibuat dengan memperhatikan beberapa parameter yang akan diterapkan.

- Penulisan laporan

Tahap ini merupakan tahap terakhir, yang akan dilakukan yaitu berupa dokumentasi dari hasil kerja yang dilakukan dan sebagai penarik kesimpulan dari langkah-langkah sebelumnya.