

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada masa saat ini perkembangan teknologi semakin cepat karena teknologi yang mendahului sudah tidak mampu melayani kebutuhan di masa saat ini. Salah satu bidang teknologi yang berkembang tersebut adalah teknologi jaringan komputer. Sekarang ini teknologi jaringan komputer dapat berfungsi dengan adanya sekumpulan protokol untuk menghubungkan antar *host* melalui perangkat-perangkat jaringan. Semakin banyak *host* yang akan dihubungkan maka semakin banyak pula protokol yang harus dikonfigurasi ke perangkat-perangkat jaringan. Hal tersebut menyebabkan jaringan komputer memiliki skalabilitas yang rendah [1].

Salah satu teknologi jaringan komputer yang sedang berkembang adalah *Software Defined Network (SDN)*. SDN merupakan sebuah paradigma dalam jaringan komputer yang memiliki konsep memisahkan antara *control plane* dengan *data plane*. Dengan memisahkan *control plane* dengan *data plane*, manajemen jaringan komputer menjadi terpusat dan perangkat jaringan seperti *switch* hanya menjadi *forwarding device*. Untuk memungkinkan terjadinya pemisahan *control plane* dengan *data plane* diperlukan protokol *OpenFlow* [2]. Berdasarkan paper W. L. da Costa Cordeiro [3], *OpenFlow* masih memiliki kekurangan yaitu kurang mampu untuk mendefinisikan perilaku *switch* dengan baik sehingga operator dibatasi oleh serangkaian protokol dan fitur yang didukung oleh *OpenFlow switch*.

Untuk mengatasi kekurangan *OpenFlow* adalah dengan dibuatnya sebuah konsep yang dinamakan *data plane programmability*. *Data plane programmability* adalah konsep yang memungkinkan untuk memodifikasi perilaku jaringan dan mengembangkan aplikasi yang inovatif pada SDN. Dengan adanya *data plane programmability* memungkinkan operator menulis kode untuk *header fields* dan dapat mendefinisikan paket *header* yang cocok. *Data plane programmability* menggunakan *domain-specific languages (DSL)* untuk memodifikasi perilaku jaringan secara efektif dari aplikasi *control plane* ke *forwarding device*. *Programming Protocol-Independent Packet Processors (P4)* merupakan salah satu contoh dari DSL [3]. P4 adalah bahasa pemrograman yang menggunakan abstraksi dari

forwarding model untuk mengekspresikan bagaimana *switch* bekerja dan bagaimana paket-paket diolah. P4 memungkinkan untuk tercapainya SDN yang mempunyai sifat *reconfigurable*, *protocol independence*, dan *target independence* [4]. Namun, sejauh ini belum ada evaluasi P4 dari segi performansi *Quality of Service* (QoS) pada SDN.

Oleh karena itu pada Tugas Akhir ini penulis akan melakukan simulasi dan analisis performansi antara *OpenFlow-based* dengan *P4-based* SDN. Penelitian ini akan dilakukan simulasi dengan menggunakan *emulator* Mininet untuk membuat topologi *spine-leaf*. Sedangkan analisis performansi akan dilakukan pengujian QoS seperti *throughput*, *packet loss*, *jitter* dan *delay*.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah dari tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara merancang paket menggunakan P4 pada SDN?
2. Bagaimana cara mengimplementasikan paket yang dibuat menggunakan P4 pada SDN?
3. Bagaimana perbandingan performansi QoS *OpenFlow-based* SDN dengan *P4-based* SDN berbasis *Docker container*?

1.3 Tujuan dan Manfaat

Tujuan dari penelitian Tugas Akhir ini adalah merancang dan mengukur performansi antara *OpenFlow-based* SDN dan *P4-based* SDN. Adapun manfaat yang dapat diperoleh melalui penelitian Tugas Akhir ini, diantaranya adalah:

1. Mengetahui proses pendefinisian paket pada protokol P4.
2. Mengetahui perbandingan performansi QoS *OpenFlow-based* SDN dengan *P4-based* SDN untuk pengiriman paket antar *host* berbasis *Docker container*.

1.4 Batasan Masalah

Tugas Akhir ini menerapkan beberapa batasan masalah sebagai berikut:

1. Tugas Akhir ini dilakukan dalam skala laboratorium.

2. SDN *controller* yang digunakan adalah *Open Networking Operation System* (ONOS) versi 2.2.2.
3. Simulasi menggunakan *software emulator* Mininet.
4. Menggunakan *software switch* OpenvSwitch dan *Behavioral Model Version* 2.
5. SDN *controller* dan *Mininet* yang dijalankan berbasis *Docker container*.
6. Menggunakan *cloud provider* *Google Cloud Platform* untuk infrastruktur pengerjaan Tugas Akhir.
7. Pengujian berdasarkan parameter QoS *throughput*, *jitter*, *packet loss* dan *delay*.

1.5 Metode Penelitian

Metode penelitian yang digunakan pada Tugas Akhir ini adalah :

1. Studi Literatur
Mencari dan mempelajari literatur tentang SDN, *OpenFlow*, dan P4 dari buku, jurnal, artikel maupun video.
2. Analisis Masalah
Melakukan analisis kemungkinan masalah yang akan terjadi. Selanjutnya melakukan diskusi dengan dosen pembimbing terkait permasalahan yang akan dibahas pada Tugas Akhir.
3. Perancangan Sistem
Melakukan perancangan sistem yang akan dilakukan kemudian melakukan instalasi *software* yang digunakan.
4. Pengujian dan Analisis
Melakukan pengujian terhadap sistem yang telah dibuat kemudian dilakukan analisis dari hasil pengujian parameter yang telah dipakai pada sistem.
5. Penyusunan Laporan
Mencatat dan menyusun hasil dari penelitian untuk dilakukan pembukuan.