

BAB I PENDAHULUAN

I.1 Latar Belakang

Arsitektur jaringan yang saat ini lebih banyak digunakan adalah arsitektur jaringan tradisional. Arsitektur jaringan komputer tradisional menggunakan media interkoneksi seperti *switch* atau *hub* untuk menghubungkan piranti yang terhubung ke dalam suatu jaringan. Model arsitektur ini memiliki beberapa kekurangan terutama dalam penanganan skalabilitas jaringan yang besar, serta dalam penanganan kepadatan lalu lintas data. Selain itu, jaringan konvensional kurang efektif jika digunakan dalam pengembangan aplikasi pengontrol jaringan (Abdillah & Tanuwidjaja, 2017).

Oleh karena itu, terciptalah arsitektur jaringan baru yang disebut *Software Defined Networking* (SDN). Konsep *Software Defined Networking* (SDN) merupakan sebuah wujud dari pemenuhan kebutuhan pengguna dalam menentukan sendiri performa dari *intermediary device* yang digunakan. Konsep *open source* yang diterapkan juga memungkinkan sebuah produk perangkat jaringan yang diproduksi oleh sebuah vendor dapat berkomunikasi dengan perangkat dari vendor lainnya yang memiliki standar yang berbeda (*proprietary*) dengan menggunakan protokol yang disebut OpenFlow (Abdillah & Tanuwidjaja, 2017). *Software Defined Networking* (SDN) memberikan kemudahan kepada pengguna dalam mengembangkan aplikasi pengontrol jaringan dengan memisahkan fungsi *data plane* dari *control plane*. Pemisahan ini juga memudahkan administrator dalam mengontrol secara langsung paket yang berjalan melalui jaringan (Sudiyatmoko dkk., 2016).

Dalam *Software Defined Networking* (SDN) terdapat *firewall* yang disebut *multi-tenancy*. *Multi-tenancy* adalah pola arsitektural yang mana satu *instance* perangkat lunak dijalankan pada infrastruktur penyedia layanan, dan beberapa penyewa mengakses *instance* yang sama. Manfaat dari *multi-tenancy* yaitu memudahkan penyedia layanan dalam menerapkan aplikasi, biaya yang dibutuhkan lebih rendah, serta menarik bagi pelanggan di segmen pasar usaha kecil dan menengah (Bezemer & Zaidman, 2010).

Metode yang digunakan dalam *multi-tenancy* adalah metode *network slicing*. Metode tersebut dibentuk agar jaringan pada sekala besar menjadi lebih fleksibel dengan memecah arsitektur jaringan menjadi elemen virtual yang dapat berjalan secara paralel. Pemisahan struktur jaringan tersebut berdasarkan definisi port yang diatur oleh slice yang diimplementasikan menggunakan FlowVisor (Muttaqin dkk., 2018).

FlowVisor sebagai salah satu perangkat yang mampu menegakan isolasi pada setiap pemisahan jaringan berada diantara *physical hardware* dan *software*. FlowVisor menggunakan protokol OpenFlow untuk mengontrol *traffic* jaringan dan mengendalikan beberapa jaringan OpenFlow dimana setiap jaringan yang dikendalikan disebut *slice*. *Slice* yang dikendalikan oleh FlowVisor memiliki satu *controller*. Setiap *controller* pada *slice* tersebut tidak bisa mengendalikan *slice* yang lain (Sherwood R dkk., 2009).

Sistem isolasi pada *network slicing* mampu membuat lapisan kuat yang mampu memisahkan setiap *slice* dengan ketentuan-ketentuan yang telah dibuat sebelumnya. *Slice* pada *network slicing* memiliki *controller* tersendiri untuk mengawasi kinerja *slice*-nya secara mandiri (Sherwood R dkk., 2010). Salah satu jenis *Controller* yang digunakan dalam penelitian ini adalah POX *controller*.

Penelitian ini bertujuan untuk melakukan penegakan isolasi *switch* CPU pada jaringan *Software Defined Networking* (SDN). Oleh karena itu, penulis mengimplementasikan metode *network slicing* dengan menggunakan pendekatan FlowVisor serta menerapkan protokol OpenFlow dan POX *controller* untuk mengatur arus *traffic* paket data pada jaringan *Software Defined Networking* (SDN) yang dilakukan pada *software* berbasis *Graphical User Interface* (GUI) yaitu Mininet.

I.2 Perumusan Masalah

Berdasarkan latar belakang tersebut, maka dapat dirumuskan permasalahan yang dibahas dalam penelitian tugas akhir ini adalah bagaimana menganalisa *network slicing* menggunakan FlowVisor pada jaringan *Software Defined Networking*

(SDN) untuk penegakan isolasi *switch* CPU dengan menerapkan OpenFlow dan POX *controller*?

I.3 Tujuan Penelitian

Berdasarkan rumusan masalah tersebut, maka tujuan dari penelitian tugas akhir yang dilakukan adalah memaparkan hasil analisa *network slicing* menggunakan FlowVisor pada jaringan *Software Defined Networking* (SDN) untuk penegakan isolasi *switch* CPU dengan menerapkan OpenFlow dan POX *controller*.

I.4 Batasan Penelitian

Batasan penelitian diperlukan agar pembahasan penelitian tidak melebar dan menyimpang dari tujuan. Oleh karena itu, batasan dari penelitian tugas akhir ini adalah sebagai berikut:

1. Penelitian hanya dilakukan dalam ruang lingkup analisa *network slicing* pada jaringan *Software Defined Networking* (SDN).
2. Penelitian hanya menerapkan protokol OpenFlow dan POX *controller*.

I.5 Manfaat Penelitian

Manfaat yang diperoleh dengan dilakukannya penelitian dalam tugas akhir ini, yaitu:

1. Bagi penulis, penelitian ini bermanfaat untuk mendapatkan pengetahuan tentang *network slicing* pada jaringan *Software Defined Networking* (SDN), serta mengembangkan isolasi *switch* CPU dengan menerapkan POX sebagai *controller* untuk mengatur arus *traffic* pada jaringan *Software Defined Networking* (SDN).
2. Bagi akademis, penelitian ini dapat menjadi media pembelajaran atau referensi untuk perkuliahan yang berhubungan dengan penelitian ini dan mendapatkan sudut pandang baru untuk perkembangan jaringan *Software Defined Networking* (SDN).

I.6 Sistematika Penulisan

Penelitian tugas akhir ini diuraikan dengan sistematika penulisan sebagai berikut:

Bab I Pendahuluan

Pada bab ini berisi uraian mengenai konteks permasalahan, latar belakang penelitian, perumusan masalah, tujuan penelitian, batasan penelitian, manfaat penelitian, dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini berisi literatur yang relevan dengan permasalahan yang diteliti dan membahas hasil penelitian terdahulu.

Bab III Metodologi Penelitian

Metodologi penelitian merupakan strategi dan langkah-langkah (*plan of attack*) yang akan dilakukan di penelitian dalam rangka menjawab rumusan masalah yang disusun sebelumnya.

Bab IV Analisis dan Perancangan

Bab ini berisi rancangan pengujian yang akan dilakukan dalam penelitian.

Bab V Pengujian dan Hasil Analisa

Bab ini berisi pengujian dan hasil analisa dari penelitian yang sudah dilakukan sesuai dengan rancangan pengujian yang telah dibuat.

Bab VI Kesimpulan dan Saran

Pada bab ini dijelaskan kesimpulan dari penelitian yang dilakukan, serta jawaban dari pertanyaan penelitian yang disajikan di pendahuluan. Saran penelitian dikemukakan pada bab ini untuk penelitian selanjutnya.