BAB 1

PENDAHULUAN

1.1 Latar Belakang

Menurut data APJII pengguna internet pada tahun 2018 sebesar 54.7% dari seluruh rakyat indonesia atau setara dengan 143.26 jiwa (Asosiasi Penyelenggara Jasa Internet Indonesia, 2018). *Software Defined Network* (SDN) adalah sebuah konsep pendekatan jaringan komputer dimana sistem pengkontrol dari arus data dipisahkan dari perangkat kerasnya. Dengan kata lain SDN juga dapat disebut sebagai sebuah paradigma baru di dunia *networking*, sebuah pendekatan dalam membangun, mendesain serta mengelola jaringan komputer. Dalam SDN seorang administrator jaringan dapat mengelola jaringan melalui konsol pengontrolan terpusat tanpa harus menyentuh *switch* secara langsung. (Anggara, 2015)

SDN erat kaitannya dengan *OpenFlow* sehingga banyak yang beranggapan bahwa SDN adalah *OpenFlow* itu sendiri. *OpenFlow* adalah sebuah elemen dalam arsitektur SDN. *OpenFlow* merupakan *open standar* komunikasi protokol yang mampu melakukan pemisahan antara *control plane* dan *data plane* dari sebuah perangkat jaringan, serta mampu menciptakan komunikasi yang sangat baik antara *control plane* dan *data plane*. Namun perlu diingat bahwa *OpenFlow* bukanlah satu-satunya protokol yang dapat digunakan untuk pengembangan SDN. Ada BGP, XMPP, OVSDB, dan MPLS-TP.(Kaur *et al*, 2014)

Controller SDN adalah aplikasi SDN yang mengelola *flow control* untuk mengaktifkan *intelligent networking. Controller* SDN bekerja berdasarkan protokol seperti *OpenFlow* yang memungkinkan *server* memberitahu kemana paket dikirim. *Controller* adalah inti dari jaringan SDN. Terletak antara *network device* dan aplikasi. Setiap komunikasi perangkat yang dilakukan dengan aplikasi harus melalui *controller*. *Controller* juga menggunakan protokol *OpenFlow* untuk mengkonfigurasi perangkat jaringan dan memilih jalur yang optimal untuk aplikasi trafik. (Kaur *et al*, 2014)

Beberapa tahun terakhir teknologi SDN menjadi salah satu topik menarik bagi peneliti. Teknologi SDN merupakan teknologi jaringan dimana bagian infrastruktur perangkat, yakni *control plane* dan *data plane* dilakukan pemisahan, sehingga

kebijakan *routing* dapat dilakukan terpusat melalui *controller*. Pengkontrolan jaringan terpusat SDN membuat pengaturan jaringan lebih mudah dan fleksibel. Pengembangan SDN yang dilakukan saat ini telah meliputi bagian hal seperti *load balancing*, VLAN, dan salah satunya mengenai protokol *routing* melalui *RouteFlow*.

SDN mempunyai banyak *controller* seperti; Floodlight, Open Daylight, Maestro, NOX, POX dan lain-lain. *Controller* tersebut dibangun dalam Bahasa pemrograman yang berbeda-beda. Beberapa kontroller dibangun dari *controller* yang lain dan merupakan pengembangan dari *controller* sebelumnya.Alasan peneliti memilih 2 (dua) *controller* ini adalah karna kedua *controller* ini memakai bahasa pemrograman yang berbeda.

POX adalah *controller* yang sebelumnya dikembangkan dengan nama NOX yang dibangun berdasarkan bahasa pemrograman Python. NOX awalnya dikembangkan oleh Nicira *Networks* bersamaan dengan *OpenFlow*. NOX adalah *controller OpenFlow* pertama. Nicira menyumbangkan NOX untuk komunitas riset pada tahun 2008. Sedangkan OpenDayLight dibangun berdasarkan bahasa pemrograman Java.(Putra *et al*, 2018).

OpenDayLight adalah *open source* dan pengontrol SDN berbasis Java yang dikelola oleh Linux Foundation dan didukung oleh lebih dari 40 perusahaan seperti IBM, Cisco, Juniper, VMWare dan sejumlah *vendor* jaringan besar lainnya. Ini dikenal sebagai pengontrol yang paling terdokumentasi.

Tujuan peneliti dalam memilih judul ini adalah untuk menentukan *controller* dengan performa yang lebih baik antara POX dan OpenDayLight, dan untuk menemukan *controller* yang paling optimal dalam membuat suatu jaringan.

1.2 Identifikasi Masalah

Dari latar belakang yang tertulis di atas, peneliti mengidentifikasi masalah yang akan dijadikan bahan penelitian sebagai berikut:

1. Performa kerja dari OpenDayLight dan POX SDN *controller*
2. Perbandingan OpenDayLight dan POX berdasarkan *throughput* dan *Latency*

1.3 Rumusan Masalah

Berdasarkan latar belakang dan identifikasi yang tertulis di atas berikut adalah rumusan masalah:

1. Bagaimana performa kerja dari POX SDN *controller* berdasarkan *Latency*?
2. Bagaimana performa kerja dari OpenDayLight SDN *controller* berdasarkan *Latency*?
3. Bagaimana performa kerja dari POX SDN *controller* berdasarkan *throughput*?
4. Bagaimana performa kerja dari OpenDayLight SDN *controller* berdasarkan *throughput*?
5. Bagaimana perbandingan kecepatan dari kedua SDN *controller* menggunakan topologi dalam skala sedang-besar?

1.4 Batasan Masalah

Mengingat luasnya jangkauan dari permasalahan ini maka perlu adanya batasan-batasan masalah yang jelas mengenai apa yang dibuat dan diselesaikan dalam tugas akhir ini. Adapun batasan-batasan masalah pada penelitian ini sebagai berikut:

1. Pembuatan simulasi topologi menggunakan mininet.
2. *Controller* yang dibandingkan adalah OpenDayLight dan POX *controller*.
3. Bahasa pemrograman yang digunakan *controller* OpenDayLight adalah Bahasa pemrograman Java.
4. Bahasa pemrograman yang digunakan *controller* POX adalah Bahasa pemrograman Python

1.5 Tujuan penelitian

Berdasarkan latar belakang yang tertulis diatas dapat diketahui tujuan sebagai berikut;

1. Untuk mengetahui performa kerja *controller* POX
2. Untuk mengetahui performa kerja *controller* OpenDayLight
3. Untuk mengetahui perbandingan dalam *controller* yang lebih optimal antara OpenDayLight dan POX

1.6 Manfaat Penelitian

Berdasarkan latar belakang yang sudah ditulis diatas dapat diketahui manfaat sebagai berikut;

1. Mengetahui hasil perbandingan OpenDayLight dan POX melalui *throughput* dan *Latency*
2. Memberikan informasi tentang performa *controller* mana yang lebih optimal untuk jaringan kecil atau jaringan yang menengah-besar.