

BAB I PENDAHULUAN

1.1 Latar Belakang

Pemanfaatan teknologi jaringan komputer sebagai media komunikasi data hingga saat ini semakin meningkat. Namun, infrastruktur jaringan saat ini masih dianggap belum mampu menangani kebutuhan tersebut sehingga muncul konsep jaringan baru yang disebut *Software Defined Network* (SDN). SDN merupakan sebuah metode dalam jaringan komputer yang sistem kontrolnya memisahkan jalur kontrol (*control plane*) dan jalur data (*data plane*). Metode ini dikembangkan sekitar tahun 2008 di *University of California, Berkley* dan *Stanford University*. [1]. SDN adalah pendekatan manajemen jaringan yang memungkinkan konfigurasi jaringan yang dinamis dan efisien secara terprogram untuk meningkatkan kinerja. SDN diciptakan untuk mengatasi arsitektur jaringan tradisional yang statis, terdesentralisasi dan kompleks, karena jaringan saat ini membutuhkan lebih banyak fleksibilitas dan pemecahan masalah yang mudah [2].

SDN *controller* merupakan aplikasi SDN yang mengelola *flow control* untuk mengaktifkan *intelligence networking*. SDN *controller* bekerja berdasarkan protokol seperti *OpenFlow* yang memungkinkan server memberitahu kemana paket dikirimkan. Kemudian perangkat meneruskan paket data yang diterima berdasarkan aturan yang ditetapkan dari *controller* [3]. Fungsi utama dari kontroler SDN adalah untuk membentuk flow. *Controller* menentukan apa yang harus dilakukan paket dan jika perlu mengirimkan aturan baru untuk perangkat sehingga dapat menangani paket dimasa depan dengan cara yang sama [14].

NOX merupakan *controller* pertama saat SDN dirilis pada Tahun 2008. *Controller* ini menggunakan bahasa pemrograman C++ dan sejak saat itu *NOX* telah menjadi dasar banyak penelitian dalam eksplorasi awal jaringan SDN. [21]

Pyrectic merupakan keluarga dari *frenetic*, *controller pyrectic* memungkinkan memprogram jaringan modular yang ringkas dengan menyediakan abstraksi yang kuat. *Pyrectic* menggunakan bahasa pemrograman *python*. [22]

RUNOS merupakan *controller* SDN terdistribusi sumber terbuka. Dikembangkan di Russia oleh *Applied Research Center for Computer Networks* (ARCCN) dengan bahasa pemrograman C++. [23]

OpenDaylight (ODL) merupakan sebuah infrastruktur *controller* yang memiliki ketersediaan tinggi, *modular*, *extensible*, *scalable* dan *multi-protocol*,

dibangun untuk penyebaran SDN di jaringan modern. *OpenDaylight* memiliki kelebihan dibandingkan dengan controller lain diantaranya mampu melakukan manajemen perangkat yang lebih baik, dapat meminimalisir kesalahan konfigurasi, dan mendukung protokol SDN *OpenFlow*. *OpenDayLight* adalah sebuah proyek *Open Source software* dalam naungan *Linux Foundation*. Tujuannya melanjutkan penerapan dan inovasi *Software Defined Network* (SDN) melalui penciptaan *framework* yang umum dikalangan industri [15].

Pada penelitian sebelumnya, membahas mengenai manajemen *Quality of Service* (QoS) pada SDN menggunakan *controller NOX, Pyrectic, dan RUNOS*. Hasil yang didapat dari ketiga *controller* tersebut belum memenuhi standar QoS ITU-T G. 1010. Pengujian berkaitan dengan QoS menggunakan parameter *throughput, packet loss, dan jitter* [13]. Dapat dilihat pada tabel 1.1

Tabel 1. 1 Nilai QoS *NOX, Pyrectic, dan RUNOS*

No	Controller	Parameter QoS		
		<i>throughput</i>	<i>packet loss</i>	<i>jitter</i>
1	<i>NOX</i>	540 kbps	11 %	0,0012 ms
2	<i>Pyrectic</i>	600 kbps	0 %	1,00543 ms
3	<i>RUNOS</i>	1 Mbps	0 %	1,0122 ms

Berdasarkan latar belakang tersebut, dimana hasil pengukuran dari beberapa *controller* pada SDN dalam penelitian terdahulu dinyatakan belum memenuhi standar QoS ITU-T G. 1010. Sehingga dibutuhkan *controller* lain untuk di uji sebagai solusi dari permasalahan tersebut. Untuk itu, dalam penelitian ini akan dilakukan pengukuran terhadap *opendaylight controller* dengan tujuan untuk mengetahui performa *opendaylight* sebagai *controller* pada SDN.

1.2 Rumusan Masalah

1.2.1 Permasalahan

Beberapa *controller* seperti *NOX, Pyrectic, dan RUNOS* yang tersebar pada jaringan SDN belum memenuhi standar QoS ITU-T G. 1010. Sehingga pengukuran performa *controller* lain dibutuhkan untuk menangani permasalahan tersebut.

1.2.2 Usulan Solusi

Penelitian ini mengusulkan pengukuran performa *opendaylight controller* pada SDN dengan parameter QoS sebagai solusi pemilihan *controller* untuk SDN

1.2.3 Pertanyaan Penelitian

Berapa nilai *throughput*, *packet loss*, dan *jitter* yang dihasilkan pada uji coba menggunakan *opendaylight controller* ?

1.3 Tujuan dan Manfaat

1.3.1 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah untuk mengetahui performa dari *opendaylight controller* sebagai pengontrol pada SDN.

1.3.2 Manfaat Penelitian

Manfaat dari penelitian ini sebagai solusi untuk menerapkan konfigurasi jaringan berbasis SDN menggunakan *opendaylight controller* .

1.4 Batasan-batasan

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Topologi yang digunakan adalah topologi *mesh*
2. Menggunakan *virtual box* untuk melakukan simulasi jaringan
3. Menggunakan sistem operasi *Linux Ubuntu*
4. Menggunakan *mininet* untuk merancang jaringan
5. Menggunakan *Opendaylight controller* sebagai *controller*
6. Pengukuran QoS (*Quality of Service*) layanan data berupa *throughput*, *packet loss*, dan *jitter*
7. Menggunakan *Wireshark* dan *Iperf* untuk menghitung nilai *throughput*, *packet loss*, dan *jitter*

1.5 Sistematika Penulisan

Sistematika penulisan Proposal ini terdiri dari beberapa bab, yaitu :

BAB I PENDAHULUAN

Pada bab ini membahas tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika penulisan laporan.

BAB II TINJAUAN PUSTAKA

Pada bab ini menjelaskan teori – teori yang mendukung penelitian, serta menjadi dasar dalam pemecahan masalah.

BAB III METODE USULAN

Pada bab ini membahas tentang metode yang diusulkan, arsitektur data penelitian, tahapan penelitian, serta rencana pengujian.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang pelaksanaan pengujian yang telah disimulasikan dan diperoleh hasil pengujian kemudian di analisis.

BAB V PENUTUP

Pada bab ini membahas tentang kesimpulan dan saran dari keseluruhan tahapan pelaksanaan penelitian ini.