

1. Pendahuluan

1.1 Latar Belakang

Pada masa perkembangan teknologi seperti sekarang, banyak aspek mengalami perkembangan yang pesat, termasuk teknologi dalam bidang arsitektur jaringan. Banyaknya masalah serta rasa ketidakpuasan menjadi salah satu alasan mengapa bidang arsitektur jaringan juga harus berkembang. Dengan arsitektur jaringan konvensional, masih ada beberapa hal yang tidak memuaskan seperti *high operational costs*, *difficult to manage*, selalu ada masalah dengan skalabilitas jaringan, *decentralization*, dan lain sebagainya.

Software-Defined Network (SDN) hadir sebagai solusi dari ketidakpuasan terhadap arsitektur jaringan konvensional. SDN memiliki prinsip utama yaitu sentralisasi, dimana semua konfigurasi dilakukan pada *control plane*. Hal ini terjadi karena konsep SDN memisahkan *control plane* dengan *data plane* [8].

SDN memiliki sebuah protokol yang dapat mengontrol *data plane* melalui *control plane* dengan memanfaatkan *controller*, protokol tersebut dinamakan OpenFlow. Ada beberapa *controller* pada OpenFlow seperti POX, Floodlight, OpenDayLight, Ryu dan lain-lain. Dengan semakin berkembangnya penelitian dan implementasi SDN, maka aspek penting seperti *controller* merupakan hal dasar yang sangat penting untuk dipertimbangkan dalam implementasi SDN. Dengan banyaknya *controller* yang hadir, maka penting untuk mengetahui jenis *controller* mana yang akan dipilih, tentunya dengan performansi terbaik. Floodlight merupakan sebuah *controller* yang bersifat *open source* dan dikembangkan oleh BigSwitch, dirancang agar mempunyai performa tinggi dan dirancang dengan Bahasa java. Sementara OpenDayLight merupakan *controller* yang dirancang dan dikembangkan setelah floodlight. OpenDayLight menyediakan set layanan yang lengkap baik untuk jaringan dasar maupun *extended network* untuk mengoptimisasi *network resource*. Sama seperti floodlight, ODL juga dibangun dengan bahasa java. POX merupakan sebuah *controller* pada SDN yang bersifat *open source*, sama seperti floodlight, ODL dan Ryu. POX dibangun dengan menggunakan Bahasa python. POX dibangun dan dikembangkan berdasarkan pada *controller* NOX, POX dirancang agar memudahkan *developer* untuk membangun serta merancang jaringan lebih

cepat dan efisien daripada NOX. Sama halnya dengan POX, Ryu merupakan sebuah *controller* yang dibangun dengan bahasa python.

Dalam penelitian ini, keempat *controller openflow* dengan tingkat popularitas tertinggi akan diuji performansinya dalam jaringan berbasis SDN. Pemilihan keempat *controller* tersebut berdasarkan pada bahasa pembangunnya yang berbeda. Floodlight dan ODL dibangun berdasarkan Bahasa java, sementara Ryu dan POX dibangun berdasarkan bahasa python. Dengan hasil penelitian ini, diharapkan dapat menjadi acuan bagi pengembang untuk menentukan *controller Openflow* mana yang lebih baik digunakan. Selain itu, alasan mengapa memilih keempat *controller* tersebut adalah berdasarkan hasil survey yang dilakukan oleh [19] yang hasilnya menyatakan bahwa keempat *controller* tersebut adalah yang paling banyak digunakan. Ada banyak penelitian yang secara garis besar melakukan penelitian mengenai perbandingan dari beberapa *controller*, namun penelitian tersebut lebih berfokus terhadap tingkat kompleksitas suatu topologi jaringan, namun aspek penting seperti pengaruh tingkat keberhasilan transaksi data paket belum diteliti. Oleh karena itu, penelitian ini akan melakukan perbandingan dan analisis SDN *controller OpenFlow* dengan metode paket.

1.2 Rumusan Masalah

Rumusan masalah yang diajukan pada tugas akhir ini adalah sebagai berikut:

1. Bagaimana performansi keempat *controller Openflow* yang disimulasikan?

1.3 Tujuan

Adapun tujuan yang ingin dicapai pada tugas akhir ini adalah sebagai berikut:

1. Untuk mengetahui performansi keempat *controller Openflow* yang disimulasikan.

1.4 Batasan Masalah

Berikut batasan masalah pada tugas akhir ini:

1. Menggunakan rancangan topologi yang bersumber dari [2] dengan IP masih satu subnet.
2. Menggunakan Mininet sebagai emulator simulasi.
3. Parameter performansi yang digunakan adalah *latency, throughput, jitter* dan *packet loss*.
4. *Controller* yang digunakan berbasis Openflow

1.5 Hipotesa Awal

Dari beberapa penelitian yang sudah dilakukan dengan metode keberagaman kompleksitas topologi jaringan, maka diperoleh *controller* dengan performansi terbaik dari keempat *controller* yang diujikan adalah Ryu, dengan ODL merupakan *controller* yang memiliki performansi paling buruk. Pada penelitian dengan metode paket, Ryu juga diprediksi lebih unggul karena memiliki fitur yang sangat lengkap, sementara ODL diprediksi memiliki performansi buruk karena masih memiliki masalah pada *memory leakages*.

1.6 Metode Penelitian

Adapun metodologi yang akan dilakukan untuk penyelesaian tugas akhir ini adalah sebagai berikut:

1. Studi literature

Studi literature merupakan sebuah kegiatan yang dilakukan untuk mempelajari literature yang berkaitan dengan topik yang diajukan. Melakukan pencarian data dan sumber terkait secara mandiri, baik dari buku, paper, tugas akhir, jurnal, dan lain sebagainya.

2. Perancangan system

Perancangan system merupakan suatu kegiatan yang dilakukan sebagai persiapan implementasi berdasarkan hasil yang didapat dari studi literature. Pada tahap ini, akan dibahas mengenai topologi jaringan, spesifikasi perangkat, scenario yang akan digunakan, dan lain-lain.

3. Implementasi system

Implementasi system merupakan suatu bentuk pengaplikasian berdasarkan hasil perancangan system.

4. Pengujian

Pengujian dilakukan terhadap system yang telah di implementasikan untuk menguji masing-masing performansi *controller* yang diteliti.

5. Analisis

Melakukan analisis terhadap hasil pengujian berdasarkan rumusan masalah dan tujuan yang ada.

6. Penarikan kesimpulan

Menarik kesimpulan berdasarkan data dari tahap pengujian dan analisis .

7. Pembuatan laporan akhir

Membuat laporan akhir berdasarkan data yang diperoleh dari pengujian dan analisis yang dilakukan selama pembuatan tugas akhir sesuai dengan ketentuan yang berlaku.

1.7 Jadwal Pelaksanaan

Berikut rencana jadwal kegiatan tugas akhir:

Kegiatan	Bulan ke-1	Bulan ke-2	Bulan ke-3	Bulan ke-4	Bulan ke-5	Bulan ke-6	Bulan ke-7
Studi Literatur							
Persiapan Data							
Analisis Perancangan							
Implementasi Prototipe							
Pengujian dan Analisa Hasil							
Penyusunan Laporan dan Dokumentasi							