

1. Pendahuluan

Latar Belakang

Arsitektur SDN menyederhanakan manajemen jaringan dan memungkinkan jaringan untuk diprogram[1]. Penyederhanaan manajemen jaringan dilakukan dengan memisahkan antara *control plane* dan *data plane*. *Control plane* berfungsi sebagai kendali jaringan terpusat dan menghasilkan *routing table* sementara *data plane* berfungsi sebagai perangkat *forwarding* sederhana yang dapat diprogram melalui antarmuka[2].

Dalam arsitektur SDN peran *controller* menjadi sangat penting karena seluruh jaringan dikelola oleh *controller*. Penggunaan *single controller* pada *traditional centralized architecture* memiliki tantangan yang besar ketika intensitas paket yang masuk tinggi karena menyebabkan *controller* rentan terhadap *overload*. Untuk menyelesaikan masalah tersebut paper [2] telah mengimplementasikan *multiple controller* berarsitektur hirarki, dimana dalam arsitektur tersebut terdapat sebuah *super controller* dan beberapa *local controller*. Implementasi *multiple controller* dalam satu jaringan mencegah terjadinya *overload* pada sebuah *controller* karena beban akan didistribusikan secara merata pada seluruh *controller* oleh *super controller*. Namun metode yang diajukan oleh paper tersebut tidak mempertimbangkan pemilihan *switch* dalam proses *switch migration* dan kegagalan pada *super controller*. Kegagalan pada *super controller* dapat menyebabkan seluruh jaringan *down* sedangkan pemilihan *switch* yang salah akan mengakibatkan *load oscillation*.

Untuk mengatasi masalah di atas diusulkan metode *load balancing* pada SDN berbasis *distributed decision* dan *switch group*. Pada metode ini seluruh *controller* pada jaringan dapat menjadi *super controller* dan pemilihan *switch* dalam proses *switch migration* mempertimbangkan beban pada setiap *switch* agar jumlah beban sesuai dengan beban yang dapat ditampung oleh *controller* tujuan.

Topik dan Batasannya

Penggunaan metode *load balancing* berbasis *distributed decision* dan *switch group* untuk meningkatkan *availability* dan mencegah *load oscillation* pada arsitektur SDN dengan intensitas paket masuk yang tinggi.

Batasan masalah yang digunakan dalam penelitian ini yaitu simulasi menggunakan Mininet, *controller* yang digunakan adalah Pox *controller*, menggunakan protokol Openflow, protokol yang digunakan untuk komunikasi antar *controller* adalah RabbitMQ, dan keterangan *switch* disimpan dalam *database* MySQL.

Tujuan

Berdasarkan pada hal-hal yang telah dijelaskan sebelumnya, tujuan dari tugas akhir ini adalah untuk meningkatkan *availability* dan mencegah *load oscillation* pada arsitektur SDN dengan mensimulasikan *load balancing* berbasis *distributed decision* dan *switches group*.

Organisasi Tulisan

Bagian selanjutnya terdiri atas studi terkait, sistem yang di bangun, evaluasi, dan kesimpulan. Pada bagian studi terkait akan dijelaskan studi literatur dan teori yang menjadi dasar pengembangan dan pelaksanaan penelitian ini. Penjelasan mengenai rancangan dan implementasi sistem akan dijelaskan lebih lanjut pada bagian sistem yang akan dibangun. Dari implementasi tersebut akan dilaksanakan beberapa skenario untuk melakukan pengujian terhadap sistem. Hasil dan analisis berdasarkan uraian pada bagian evaluasi akan dituliskan pada bagian kesimpulan.