

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Software-defined networks (SDN) digunakan sebagai solusi dalam bidang IT dan arsitektur jaringan untuk mengatasi kekurangan arsitektur jaringan konvensional berbasis perangkat keras (*hardware*) [1]. SDN bekerja dengan memisahkan *control plane* (pengatur lalu lintas data) dari *data plane* (pengirim lalu lintas data) yang dilakukan melalui standar komunikasi protokol (*Openflow*) sehingga memungkinkan untuk mengontrol jaringan secara langsung dan terpusat [1]. SDN merupakan arsitektur jaringan yang bersifat dinamis, mudah beradaptasi, mudah dikelola, dan hemat biaya sehingga cocok diterapkan pada jaringan dengan *bandwidth* tinggi [2]. Pada dasarnya, arsitektur ini menawarkan kemudahan sentralisasi pengaturan dan konfigurasi perangkat pada satu *controller* yang bertindak sebagai unit *control plane*. Dengan adanya sentralisasi kerja, *controller* memiliki tampilan jaringan lebih baik dan dapat membuat rute jaringan berdasarkan frekuensi informasi terbaru pada jaringan dan mengubahnya sesuai waktu hari itu (*real-time*) atau *request traffic*. *Controller SDN* dapat mengelola rute jaringan dengan menambah atau mengurangi sumber daya yang dibutuhkan untuk menjalankan layanan ketika sebuah perangkat digunakan guna mengakses data besar [3]. Selain memonitor keadaan jaringan, *controller SDN* juga dapat melakukan perbaikan jika terjadi kerusakan jaringan [4].

Pengimplementasian *pure SDN* memiliki keterbatasan meskipun penggunaannya sangat menguntungkan dalam meningkatkan kemampuan kontrol lalu lintas jaringan, mengurangi konsumsi sumber daya jaringan, dan menyeimbangkan tautan beban. Teknologi *SDN* membutuhkan pembaruan arsitektur sehingga dibutuhkan anggaran besar untuk menginisiasi perubahan peralatan dan pelatihan dan/atau perekrutan tenaga profesional. Selain itu, produksi *controller SDN* membutuhkan waktu yang cukup lama dan implementasi *pure SDN* masih dalam tahap pengembangan awal [5]. Pengembangan kerangka *pure SDN* masih dibutuhkan untuk pengelolaan integritas *software*, manajemen jarak jauh, deteksi dan mitigasi ancaman

keamanan jaringan, serta otentikasi dan otorisasi pengguna [1]. Sementara itu, *hybrid SDN* ditawarkan sebagai solusi untuk mengatasi permasalahan migrasi dari jaringan tradisional ke *pure SDN*. *Hybrid SDN* bekerja dengan menggabungkan perangkat konvensional dan perangkat *SDN* untuk menyeimbangkan beban *traffic*. Dengan kata lain, mode *terpusat* (di *pure SDN*) dan desentralisasi (di jaringan tradisional) ada secara berdampingan dan saling berkomunikasi untuk mengontrol, memodifikasi, mengkonfigurasi, dan mengelola perilaku jaringan sehingga kinerja jaringan dan pengalaman pengguna dapat dioptimalkan [6]. Meskipun *hybrid SDN* telah dicanangkan sebagai solusi permasalahan *pure SDN*, penelitian lebih lanjut tentang perbandingan kinerja antara *hybrid SDN* dan *pure SDN* diperlukan untuk mengetahui apakah *hybrid SDN* memang merupakan solusi efisien dari permasalahan *pure SDN*. Pengujian *Quality of Service* (QoS) pada kedua arsitektur ini dapat dilakukan menggunakan parameter seperti *jitter*, *latency*, *delay*, *packet loss*, *Mean Opinion Score (MOS)*, dan *throughput* [7].

Pengujian arsitektur *hybrid SDN* dan/atau *pure SDN* membutuhkan web server. Penggunaan web server tunggal untuk melayani banyak *request* klien secara bersamaan akan berakibat kelebihan beban, peningkatan waktu respon server, dan kemungkinan terjadi penolakan *request* setelah waktu tertentu [8]. Oleh karena itu, *load balancing* dibutuhkan untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang supaya *traffic* berjalan optimal, *throughput* maksimal, waktu tanggap server semakin kecil, dan *overload* serta *crash* pada jalur koneksi dapat dihindari [9]. Salah satu pilihan yang dapat digunakan sebagai *load balancer* adalah algoritma *round robin*. Algoritma ini dapat membagi beban secara bergiliran dan berurutan dari satu server ke server lain. Selain itu, algoritma *round robin* dapat bekerja sebagai penyeimbang beban server saat merespon setiap *request user* [10]. Cek yang lain yang belum italic untuk istilah asing

Berdasarkan pemaparan permasalahan tentang SDN di atas, penulis bermaksud melakukan penelitian dengan judul “Analisis *load balancing* Menggunakan *round robin* pada *hybrid SDN* dan *pure SDN*”. Pengujian performansi *load balancing* pada arsitektur *hybrid SDN* dan *pure SDN* akan

dilakukan dengan mengintegrasikan algoritma *round robin* dan *controller SDN* berupa *ONOS*. Parameter *QoS* yang akan digunakan yaitu *jitter*, *delay*, *throughput* dan *packet loss*. Penelitian ini bertujuan untuk mengetahui perbandingan *QoS load balancing* pada *hybrid SDN* dan *pure SDN* menggunakan algoritma *round robin* berdasarkan parameter *jitter*, *delay*, *packet loss* dan *throughput*.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah:

1. Bagaimana perbandingan *QoS* pada arsitektur *hybrid SDN* menggunakan *round robin* berdasarkan standar TIPHON?
2. Bagaimana perbandingan *QoS* pada arsitektur *pure SDN* menggunakan *round robin* berdasarkan standar TIPHON?

1.3 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Mengetahui perbandingan parameter *QoS* menggunakan algoritma *round robin* dalam arsitektur *hybrid SDN* menurut standar TIPHON.
2. Mengetahui perbandingan parameter *QoS* menggunakan algoritma *round robin* dalam arsitektur *pure SDN* menurut standar TIPHON.

1.4 Batasan Masalah

Beberapa batasan masalah yang ditetapkan dalam penelitian terdiri dari :

1. Menggunakan Sistem Operasi *Backbox* versi ke 7 yang berbasis Linux Ubuntu.
2. Simulasi yang dijalankan menggunakan *controller ONOS*, 2 *switch konvensional*, 2 *router FRRouting*, 2 unit *Openvswitch*
3. Metode yang digunakan adalah *load balancing round robin*.
4. Parameter *Quality of Services* yang diukur yaitu *jitter*, *delay*, *throughput* dan *packet loss*.
5. Standar *QoS* menggunakan standardisasi THIPON.
6. Simulasi hanya menampilkan perbedaan *QoS* pada dua arsitektur jaringan yang berbeda yaitu *hybrid SDN* dan *pure SDN*.
7. Tidak membahas keamanan jaringan, *system operation*, dan media jaringan

8. Menggunakan *Apache* pada *Web Server*
9. Menggunakan *ONOS* sebagai *controller*
10. Menggunakan *IPv4* dalam jaringan lokal.

1.5 Manfaat Penelitian

Penelitian diharapkan bermanfaat dan bisa menjadi acuan dalam mengetahui performansi *load balancing* melalui perbandingan *QoS* yang ditawarkan pada dua arsitektur jaringan berbeda dengan algoritma *round robin* menggunakan *controller ONOS*.

1.6 Sistematika Penulisan

Penulisan skripsi ini memiliki sistematika penulisan yang terdiri dari 5 bab yaitu :

BAB I : PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah dan metode penelitian yang digunakan pada tugas akhir ini.

BAB II : DASAR TEORI

Bab ini menjelaskan tentang teori-teori pendukung yang mendasari pengerjaan tugas akhir ini, yaitu *hybrid SDN*, *pure SDN*, *load balancing*, algoritma *round robin*, *ONOS controller* dan materi yang mendukung tugas akhir ini.

BAB III : PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi pembahasan tentang langkah-langkah perancangan topologi *hybrid SDN* dan *pure SDN* serta sistem pengujian performansi *load balancing*.

BAB IV : ANALISIS DAN PEMBAHASAN

Bab ini berisi analisis hasil pengujian dan pembahasan yang telah dilakukan dari desain, implementasi, dan aplikasi untuk memodelkan jaringan yang telah dibuat.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang dapat diambil dari analisis dan pembahasan yang telah dibahas sebelumnya dan saran perbaikan tugas akhir ini untuk penelitian selanjutnya.