

BAB I PENDAHULUAN

1.1 Latar Belakang

Layanan online populer yang sering digunakan oleh pengguna internet adalah media sosial dan situs *web*. Layanan tersebut harus mampu menangani banyak trafik, jika layanan bergantung pada satu *server web*, maka *single point of failure* dapat terjadi kapan saja. *Single point of failure* (SPOF) adalah kesalahan dalam sistem yang menghentikan sistem tersebut untuk berfungsi. Salah satu penyebab terjadinya SPOF adalah *overload*, dimana terjadinya kelebihan muatan dalam kapasitas *server* dalam menangani permintaan klien. Permasalahan tersebut dapat diatasi dengan menerapkan *load balancing* pada *server*, yaitu membagi permintaan dari klien ke beberapa *server* yang tersedia. Penyeimbangan beban adalah teknologi jaringan komputer yang mendistribusikan beban kerja di berbagai komputer atau kluster komputer. *Cluster* komputer sering digunakan untuk meningkatkan kinerja dan ketersediaan satu computer [1]. *User* yang mengirim permintaan ke *server load balancing* dapat melihat operasi *load balancing* secara *real time*. Adapun beberapa algoritma yang diterapkan untuk *load balancing* antara lain yaitu, *Round Robin*, *Least Bandwidth*, *IP Hash*, *Least connection*, *Least Respon time* [2].

Teori jaringan baru yang dikenal sebagai *Architecture Software Defined Network* (SDN) dapat digunakan untuk merancang metode *load balancing* [3]. Kemampuan SDN memecahkan sejumlah masalah antar perangkat jaringan yang berbeda pada *web server* sehingga mencegah *server down* karena terlalu banyak permintaan dalam waktu singkat, *load balancing* yang akan mendistribusikan beban *trafik* pada dua atau lebih jalur koneksi secara seimbang dan menangani kesulitan *server* sehingga trafik dapat berjalan [4].

Penelitian razzak, dkk. Penerapan algoritma *fuzzy* untuk memuat *server* seimbang pada SDN dengan memperhitungkan penggunaan sumber daya saat ini dari setiap *server* adalah salah satu cara untuk mengatasi kelebihan beban. Penerapan logika *fuzzy* pada *load balancing system* di arsitektur SDN dapat

mendistribusikan beban *server* berdasarkan kemampuan *server*. Hasil penelitian tersebut algoritma *fuzzy* dapat mendistribusikan pekerjaan *server* berdasarkan bobot *server* terendah dan tidak mengalami kelebihan beban [5]. Solusi lain pada mekanisme *load balancing* yaitu dengan menggunakan algoritma *naïve bayes*, pada penelitian Utomo, dkk. Penelitian ini akan menerapkan *Load balancing* dengan menggunakan penerapan algoritma *fuzzy* untuk memuat *server* seimbang pada SDN dengan memperhitungkan penggunaan sumber daya saat ini dari setiap *server* adalah salah satu cara untuk mengatasi kelebihan beban [6].

Penelitian ini mengusulkan penggunaan algoritma klasifikasi *round robin* dan *least connection*. *Round robin* pada mekanisme *load balancing* merupakan algoritma sederhana dan paling banyak digunakan. Algoritma *round robin* bekerja secara bergilir dan berurutan menggunakan *time sharing* dengan membagi beban dari satu *server* ke *server* lainnya dan tidak ada prioritas [7]. Adapun kekurangan dari Algoritma *round robin* adalah yang memiliki *Quantum Time* yang dapat memengaruhi hasil *Throughput* [8]. sedangkan algoritma koneksi paling sedikit membagi beban di seluruh koneksi yang dilayani *server*. Baik *server* dengan koneksi paling sedikit dan *server* dengan jumlah koneksi terbesar akan ditransfer bebannya ke *server* dengan beban lebih rendah., kekurangan dari algoritma *least connection* tidak mempertimbangkan kapabilitas pemrosesan masing – masing *server* [9].

Berdasarkan permasalahan dan penjelasan diatas, penelitian ini menggunakan algoritma *round robin* dan *least connection* sebagai alternatif untuk mekanisme *load balancing* pada SDN. Menggunakan *controller floodlight* untuk mencari nilai dari *Troughput*, *Respon time*, *CPU Usage*, dan *Packet loss* yang dihasilkan dengan menerapkan algoritma *load balancing* pada arsitektur *software defined network*. Dari penelitian ini, diharapkan dapat digunakan untuk menentukan algoritma mana yang lebih efektif digunakan dalam situasi tertentu untuk meningkatkan kinerja server web terutama pada *architecture Software Defined Network*.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah:

1. Bagaimana kinerja dari algoritma *round robin* pada arsitektur SDN?
2. Bagaimana kinerja dari algoritma *least connection* pada arsitektur SDN?
3. Bagaimana perbandingan dari algoritma *round robin* dan *least connection* pada arsitektur SDN?

1.3 Batasan Masalah

Batasan masalah dari penelitian ini adalah:

1. Penelitian ini menggunakan topologi dengan tiga buah *http server*, satu *client*, satu *Openflow Switch*, dan satu *SDN controller*.
2. Penelitian ini menggunakan *hypertext transfer protocol* (HTTP) sebagai protokol komunikasi pada jaringan.
3. Penelitian ini menggunakan algoritma *round robin* dan *least connection* yang diterapkan pada arsitektur SDN.
4. Parameter sebagai bahan analisis adalah *Troughput*, *Respon time*, *CPU Usage*, dan *Packet loss*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Menganalisis bagaimana algoritma *round robin* bekerja saat beban jaringan bervariasi.
2. Menganalisis bagaimana algoritma *least connection* berkerja pada saat beban yang dikirimkan.
3. Menganalisis perbandingan kinerja dari algoritma *round robin* dan *least connection* berdasarkan parameter *Troughput*, *Respon time*, *CPU Usage*, dan *Packet loss*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan gambaran mengenai kinerja dari *load balancing* menggunakan algoritma *round robin* dan *least connection*, serta dapat mengetahui hasil uji dari perbandingan antara algoritma *round robin* dan *least connection* dengan menggunakan parameter pengujian *CPU utilization*, *Respon time*, *Troughput*, dan *error* dan dapat dijadikan sebagai acuan dalam

penggunaan metode algoritma yang efisien pada *load balancing* pada arsitektur *software defined network*.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ini dibagi menjadi 3 bagian:

1. BAB 1 : PENDAHULUAN

Bagian pendahuluan berisi mengenai latar belakang, rumusan masalah yang diangkat, batasan masalah, manfaat dan tujuan, serta berisi tentang sistematika penelitian.

2. BAB 2 : DASAR TEORI

Bagian ini membahas tentang dasar teori dan kajian Pustaka yang bersangkutan dengan penelitian.

3. BAB 3 : METODE PENELITIAN

Bagian ini membahas mengenai perangkat yang digunakan, alur penelitian, topologi jaringan, dan skenario pengujian yang akan dilakukan.

4. BAB 4 : HASIL DAN PEMBAHASAN

Bagian ini membahas mengenai hasil dan analisa dari pengujian yang telah dilakukan.

5. BAB 5 : PENUTUP

Bagian ini membahas mengenai kesimpulan dan saran dari hasil dari pengujian yang telah dilakukan.

6. LAMPIRAN