

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada saat teknologi jaringan komputer berkembang dengan sangat pesat, dan dampak dari perkembangan yang pesat ini membuat kebutuhan manusia dalam hal kebutuhan internet meningkat dengan drastis, ini membuat server seringkali mengalami *overload*[1]. Cara yang banyak dilakukan untuk mengatasi ini yaitu dengan menambahkan server baru atau menambahkan *harddisk* tambahan untuk *database* tetapi hal ini membutuhkan biaya yang besar, di sisi lain terdapat teknik yang dapat mengatasi masalah ini yaitu teknik *load balancing*. *Load balancing* adalah suatu mekanisme untuk membagi beban komputasi ke beberapa server, tujuan dari *load balancing* ini untuk mengoptimalkan sumber daya, meningkatkan *throughput*, agar server tidak mengalami *overload*.

Software-Defined Network (SDN) adalah konsep pendekatan jaringan komputer dimana sistem pengontrol (*control plane*) dari arus data dipisahkan dari perangkat kerasnya (*data plane*)[2]. Pada SDN terdapat protokol *openflow* ini digunakan untuk mengontrol/mengatur *traffic flows* tetapi *openflow* ini sudah ditentukan dan tidak dapat ubah atau dimodifikasi. Programming Protocol-independent Packet Processors (P4) adalah sebuah bahasa pemrograman untuk *top-down programming* yang dapat menentukan bagaimana *pipelines* pada switch bekerja dan bagaimana paket-paket ini dapat diproses[3]. P4 ini dapat mengatasi kelemahan *openflow* yang kurang fleksibel dalam mengontrol/mengatur *traffic-flow*. IP hash merupakan algoritma *load balancing* yang dalam mendistribusikan bebannya dengan cara mencocokkan hash key dan ini membuat beban yang diterima oleh server tidak merata, untuk mengatasi kelemahan IP hash dilakukan optimasi dengan menggunakan algoritma Round Robin, untuk menerapkan *load balancing* dibutuhkan aplikasi tambahan seperti HAProxy, F5, Nginx dan lain-lain dengan P4 proses *load balancing* dapat langsung dilakukan pada *data plane* switch.

Pada penelitian sebelumnya yang berjudul “*Load Balancing Implementation Strategy for Various Services in Software Defined Network using ONOS Controller*”[4] pada penelitian ini sudah diterapkan pada jaringan sdn namun untuk protokol yang digunakan adalah openflow sehingga tidak bisa menerapkan *programmable* pada *pipeline* openflow akibatnya untuk melakukan proses load balancing memerlukan aplikasi tambahan yaitu F5. Pada penelitian yang berjudul “Analisis Dan Implementasi *Load Balancing* Pada Web Server Dengan Algoritme *Shortest Delay* Pada *Software Defined Network*”[5] namun pada penelitian ini masih menerapkan protokol openflow sehingga tidak bisa memodifikasi *pipeline* openflow dan proses load balancing terjadi pada controller ryu. Pada penelitian berjudul “Analisis Kinerja Load Balancing pada Server Web Menggunakan Algoritma Weighted Round Robin pada Proxmox VE”[6] namun pada penelitian ini untuk melakukan load balancing masih menggunakan aplikasi tambahan yaitu HAProxy dan masih di implementasikan pada jaringan konvensional sehingga pada switch tidak bisa diterapkan *programmable* di sisi *data plane*. Pada penelitian berjudul “Implementasi Load Balancing dengan Algoritma Weighted Round Robin menggunakan NGINX”[7] namun pada penelitian ini load balancing masih menggunakan aplikasi Nginx dan tidak diterapkan *programmable* pada switch. Pada penelitian berjudul “Comparative Analysis of Load Balancing Dynamic Ratio and Server Ratio Algorithms”[8] namun pada penelitian ini laod balancing masih menggunakan F5 BIG IP yang memerlukan *resource* yang besar dan tidak di implementasikan pada jaringan sdn sehingga switch yang dipakai tidak bisa untuk diprogram untuk melakukan load balancing. Pada penelitian berjudul “Implementasi High-Availability Web Server Menggunakan Load Balancing As a Service Pada Openstack Cloud”[9] namun pada penelitian ini untuk menjalankan proses load balancing menggunakan aplikasi Openstck. Pada penelitian berjudul “Load Balancing In Software Defined Networking (SDN)”[10] namun pada penelitian ini load balancing terdapat pada controller pox dan masih menggunakan protokol openflow sehingga tidak bisa melakukan *programmable* pada sisi *dataplane* switch. Pada penelitian yang berjudul “Implementasi Load Balancer Berdasarkan Server Status pada Arsitektur Software Defined Network (SDN)”[11]

namun pada penelitian ini load balancing terjadi pada controller sdn dan masih menggunakan protokol openflow.

Oleh karena itu pada Tugas Akhir ini telah dilakukan simulasi dan analisis load balancing pada *programmable network infrastructure* yang berbasis P4 *language* untuk sisi *data plane*. Untuk algoritma yang digunakan adalah IP hash dan round robin. Pada algoritma IP hash dilakukan Optimasi menggunakan algoritma round robin agar beban yang diterima oleh server sama atau didistribusikan secara merata. Penelitian ini akan menggunakan Mininet dan P4 BMv2 switch agar dapat melakukan simulasi infrastruktur jaringan.

1.2 Rumusan Masalah

Adapun rumusan masalah dari tugas akhir ini, sebagai berikut.

1. Bagaimana pengaruh algoritma round robin dan IP Hash berbasis P4 *language* pada jaringan load balancing?
2. Bagaimanakah pengaruh algoritma round robin dan IP Hash terhadap *fairness index* pada server?
3. Bagaimanakah performansi algoritma round robin dan IP Hash terhadap *throughput*?
4. Bagaimanakah performansi algoritma round robin dan IP Hash terhadap *respons time*?
5. Bagaimanakah performansi algoritma round robin dan IP hash terhadap *request loss*?

1.3 Tujuan dan Manfaat

Adapun tujuan dan manfaat dari penulisan tugas akhir ini, sebagai berikut.

1. Mengetahui dampak algoritma round robin dan IP Hash berbasis P4 *language* pada jaringan load balancing.
2. Mengetahui dampak algoritma round robin dan IP Hash terhadap *fairness index* pada server.
3. Menganalisis *throughput* pada algoritma round robin dan IP Hash.
4. Menganalisis *respons time* pada algoritma round robin dan IP Hash.
5. Menganalisis *request loss* pada algoritma round robin dan IP Hash.

1.4 Batasan Masalah

Adapun batasan masalah dari tugas akhir ini, sebagai berikut.

1. Sistem kerja dari *load balancing* menggunakan teknologi P4.
2. Tugas Akhir ini dilakukan dalam skala Laboratorium virtual.
3. Pengukuran kinerja *load balancing* menggunakan algoritma *Round Robin* dan *IP Hash*.
4. Parameter keluaran yang di gunakan adalah *Throughput*, *response time*, *request loss* dan *fairness index*.
5. Pengujian dilakukan hanya menggunakan jaringan local IPv4.
6. Tidak membahas keamanan jaringan pada server.

1.5 Metode Penelitian

Adapun metodologi pada penelitian tugas akhir ini, sebagai berikut.

1. Studi Literatur
Mencari sumber referensi yang berkaitan dengan load balancing, software defined network, server, P4 berupa jurnal, buku, dan penelitian yang dilakukan sebelumnya.
2. Perancangan dan Implementasi
Melakukan perancangan sistem SDN dan *load balancer*, topologi jaringan, algoritma, *controller* yang telah ditentukan.
3. Pengujian dan analisis
Melakukan pengujian terhadap sistem yang telah di rancang dengan skenario dan parameter uji yang telah ditentukan serta menganalisis hasil dari pengujian yang telah dilakukan sesuai dengan skenario yang telah dibuat.
4. Penarikan kesimpulan
Merupakan tahap terakhir dari keseluruhan pengerjaan proyek akhir, maka dapat diambil kesimpulan tentang analisis yang telah dilakukan.

1.6 Jadwal Pelaksanaan

Adapun jadwal pelaksanaan pengerjaan Tugas Akhir sebagai berikut.

Tabel 1 1. jadwal pelaksanaan pengerjaan Tugas Akhir

No.	Deskripsi Tahapan	Durasi	Tanggal Selesai	<i>Milestone</i>
1	Desain Sistem	2 minggu	22 Jan 2022	Diagram Blok dan spesifikasi <i>Input-Output</i>
2	Instalasi perangkat lunak	1 bulan	5 Maret 2022	Install ubuntu, p4- utils, mininet, BMv2
3	Implementasi program P4	3 bulan	28 Mei 2022	Mengintegrasikan semua komponen
4	Penyusunan laporan/buku TA	2.5 bulan	8 Agustus 2022	Buku TA selesai