

**IMPLEMENTASI LOAD BALANCING PADA SERVER DENGAN
ALGORITMA ROUND ROBIN PADA SOFTWARE DEFINED
NETWORK**

SKRIPSI



disusun oleh

Agil Angga Saputra

15.11.9200

**PROGRAM SARJANA
PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2019**

**IMPLEMENTASI LOAD BALANCING PADA SERVER DENGAN
ALGORITMA ROUND ROBIN PADA SOFTWARE DEFINED
NETWORK**

SKRIPSI

untuk memenuhi sebagian persyaratan
mencapai gelar Sarjana
pada Program Studi Informatika



disusun oleh

Agil Angga Saputra

15.11.9200

**PROGRAM SARJANA
PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2019**

PERSETUJUAN

SKRIPSI

IMPLEMENTASI LOAD BALANCING PADA SERVER DENGAN ALGORITMA ROUND ROBIN PADA SOFTWARE DEFINED NETWORK

yang dipersiapkan dan disusun oleh

Agil Angga Saputra

15.11.9200

telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 19 Februari 2019

Dosen Pembimbing,

Ahlihi Masruro, M.Kom.

NIK. 190302148

PENGESAHAN
SKRIPSI
IMPLEMENTASI LOAD BALANCING PADA SERVER DENGAN
ALGORITMA ROUND ROBIN PADA SOFTWARE DEFINED
NETWORK

yang dipersiapkan dan disusun oleh

Agil Angga Saputra

15.11.9200

telah dipertahankan di depan Dewan Penguji
pada tanggal 22 Juli 2019

Susunan Dewan Penguji

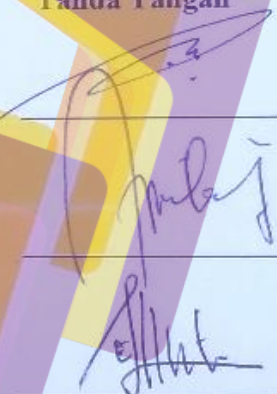
Nama Penguji

Ahlihi Masruro, M.Kom.
NIK. 190302148

Nila Feby Puspitasari, S.Kom, M.Cs.
NIK. 190302161

Erni Seniwati, S.Kom, M.Cs.
NIK. 190302231


Tanda Tangan



Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer
Tanggal 12 September 2019

DEKAN FAKULTAS ILMU KOMPUTER




Krisnawati, S.Si, M.T.
NIK. 190302038

PERNYATAAN

Saya yang bertandatangan dibawah ini menyatakan bahwa, skripsi ini merupakan karya saya sendiri (ASLI), dan isi dalam skripsi ini tidak terdapat karya yang pernah diajukan oleh orang lain untuk memperoleh gelar akademis di suatu institusi pendidikan tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis dan/atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Segala sesuatu yang terkait dengan naskah dan karya yang telah dibuat adalah menjadi tanggungjawab saya pribadi.

Yogyakarta, 6 September 2019

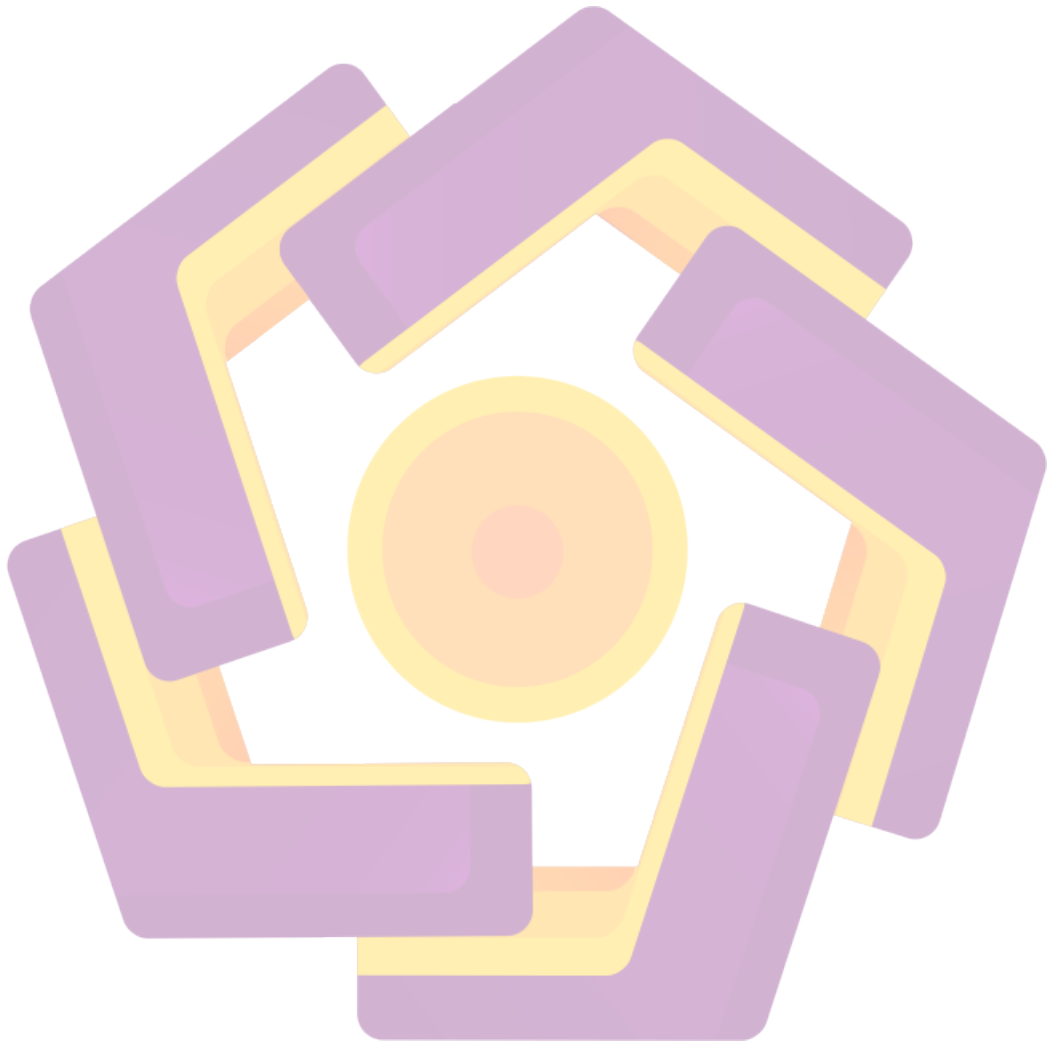


Agil Angga Saputra

NIM. 15.11.9200

MOTTO

DESTROY WHAT DESTROYS YOU



PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji Syukur kepada Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya. Serta Shalawat kepada Nabi Muhammad SAW., yang telah memberikan contoh Akhlakul Kharimah bagi seluruh umat muslim di penjuru dunia. Didalam pengerjaan skripsi ini telah melibatkan banyak pihak yang sangat membantu. Oleh sebab itu, disini penulis sampaikan rasa terima kasih sedalam-dalamnya kepada :

1. Kedua orang tua tercinta yang tak pernah lelah memberi doa, dukungan, perjuangan, motivasi, dan pengorbanan selama ini.
2. Dosen Pembimbing Bapak Ahlihi Masruro, M.Kom, terimakasih untuk bimbingannya selama ini.
3. Mbak tercinta Mbak Ika Ratna Furi dan Adisti Rani yang tak perna lelah memberi uang bulanan dan uang lainnya.
4. Rizqika Anissa Insani jele yang tak pernah henti memberikan cinta kasih tulus murni kepada ku.
5. Temen-temen seperjuangan 15-IF-11 yang tidak bisa disebutkan satu-persatu, kalian luar biasa dan terimakasih buat temen-temen semua.
6. Sobat Dolano Dewe Dewe (DDD) Akbar Muzakki, Muhammad Khairil Hasnan Habib, Wilda Mauli, Ade Rufaidah Mutmainah, Muhammad Zainal Arifin, Heri Cahyono, I Pande Made Apriliana, Restu Fajri Irawan, Dessi Putri Hartaningsih, Ulfa Nuaifah, Muhammad Ferditya Nugraha, Muhammad Aldi Prabowo,
7. Terimakasih juga kepada Raden Sugeng, Bebep Tresno, Appa, Raden Limbot, Fluffy, Moko, Karamel, Cokosip, yang telah merepotkan saya.

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis bisa menyelesaikan skripsi dengan tepat waktu, yang diberi Judul “Implementasi Load Balancing Pada Server Dengan Algoritma Round Robin Pada Software Defined Network”

Skripsi ini ditujukan untuk memenuhi salah satu persyaratan ujian guna memperoleh gelar Sarjana Komputer (S.Kom) pada Jurusan Informatika Fakultas Ilmu Komputer Universitas Amikom Yogyakarta.

Dalam penulisan skripsi ini tidak lepas dari hambatan dan kesulitan, namun berkat bimbingan, bantuan, nasihat, dan saran serta kerjasama dari berbagai pihak, khususnya pembimbing, segala hambatan tersebut akhirnya dapat diatasi dengan baik.

Selanjutnya, dalam penulisan skripsi ini tentunya tidak lepas dari kekurangan baik materi yang disajikan, tentunya semua ini didasari dari keterbatasan yang dimiliki penulis.

Terakhir semoga segala bantuan yang telah diberikan senantiasa mendapat Ridho Allah SWT. Sehingga pada akhirnya skripsi ini dapat bermanfaat bagi penelitian selanjutnya.

Yogyakarta, 6 September 2019

Penulis,


Agil Angga Saputra

15.11.9200

DAFTAR ISI

JUDUL.....	I
PERSETUJUAN.....	III
PENGESAHAN.....	IV
PERNYATAAN.....	V
MOTTO.....	VI
PERSEMBAHAN.....	VII
KATA PENGANTAR.....	VIII
DAFTAR ISI.....	IX
DAFTAR TABEL.....	XI
DAFTAR GAMBAR.....	XII
INTISARI.....	XIV
ABSTRACT.....	XV
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Maksud dan Tujuan Penelitian.....	4
1.5 Metodologi Penelitian.....	4
1.6 Sistematika Penulisan.....	6
BAB II LANDASAN TEORI.....	8
2.1 Tinjauan Pustaka.....	8
2.2 Dasar Teori.....	10
2.2.1 Konsep Dasar <i>Software-Defined Network</i>	10
2.2.2 Arsitektur <i>Software-Defined Network</i>	11
2.2.3 Openflow.....	12
2.3 Controller OpenDaylight.....	14
2.4 <i>Load balancing</i>	16
2.5 Algoritma Round Robin.....	17
2.5.1 Source Code Algoritma Round Robin.....	19
2.5.2 Flowchart Algoritma Round Robin.....	20
2.6 Mininet.....	21
2.7 Iperf.....	23
BAB III METODE PENELITIAN.....	25

3.1	Gambaran Umum Penelitian	25
3.2	Metode Penelitian.....	26
3.2.1	Desain Penelitian.....	26
3.2.2	Alur Penelitian	27
3.2.3	Pengumpulan Data	30
3.2.3.1	Jenis Data	30
3.2.3.2	Teknik Data.....	30
3.2.3.3	Instrumen Penelitian.....	30
3.2.4	Pengelolaan Data	32
3.2.5	Penyajian Data	33
3.2.5.1	Penyajian Data Bentuk Tabel	33
3.2.5.2	Penyajian Data Bentuk Diagram.....	33
3.2.6	Variabel Penelitian.....	34
3.3	Instalasi.....	35
3.3.1	OpenDaylight	35
3.3.2	Mininet.....	38
3.4	Pengujian Penelitian.....	40
3.4.1	Uji Konektivitas Mininet dengan OpenDaylight	40
3.4.2	Pengujian <i>Throughput</i>	41
3.4.2.1	Pengujian <i>Throughput</i> Tidak Menggunakan <i>Load balancing</i>	41
3.4.2.2	Pengujian <i>Throughput</i> Menggunakan <i>Load balancing</i>	46
3.4.3	Pengujian <i>Latency</i>	51
3.4.3.1	Pengujian <i>Latency</i> Tidak Menggunakan <i>Load balancing</i>	51
3.4.3.2	Pengujian <i>Latency</i> Menggunakan <i>Load balancing</i>	55
BAB IV HASIL DAN PEMBAHASAN		61
4.1	Uji dan Tabulasi Data.....	61
4.1.1	Tabulasi Data <i>Throughput</i>	62
4.1.2	Tabulasi Data <i>Latency</i>	62
4.2	Analisis dan Pembahasan	63
4.2.1	<i>Throughput</i>	63
4.2.2	<i>Latency</i>	66
BAB V PENUTUP		70
5.1	Kesimpulan.....	70
5.2	Saran.....	71
DAFTAR PUSTAKA		72

DAFTAR TABEL

Tabel 2.1 Perbedaan Referensi dan Penelitian yang Dilakukan	9
Tabel 2.2 Daftar opsi perintah Iperf.....	24
Tabel 3.1 Perangkat Lunak	31
Tabel 3.2 Spesifikasi Perangkat Keras.....	31
Tabel 3.3 Subjek Penelitian	32
Tabel 3.4 Format Tabulasi Data.....	32
Tabel 3.5 Parameter Pengujian <i>Throughput</i> Tidak Menggunakan <i>Load Balancing</i>	42
Tabel 3.6 Parameter Pengujian <i>Throughput</i> Menggunakan <i>Load balancing Round Robin</i>	47
Tabel 3.7 Parameter Pengujian <i>Latency</i> Tidak Menggunakan <i>Load balancing</i>	51
Tabel 3.8 Parameter Pengujian <i>Latency</i> Menggunakan <i>Load balancing Round Robin</i>	56
Tabel 4.1 Parameter dalam pengujian <i>Throughput</i>	61
Tabel 4.2 Parameter dalam pengujian <i>Latency</i>	61
Tabel 4.3 Data uji <i>Throughput</i> server	62
Tabel 4.4 Data uji <i>Latency</i> server	63

DAFTAR GAMBAR

Gambar 2.1 Arsitektur <i>Software-Defined Network</i>	11
Gambar 2.2 OpenFlow.....	13
Gambar 2.3 Arsitektur Controller OpenDaylight	14
Gambar 2.4 Komponen <i>Load balancing</i>	17
Gambar 2.5 Antrian Algoritma Round Robin.....	18
Gambar 2.6 Source Code Algoritma Round Robin.....	19
Gambar 2.7 Flowchart Algoritma Round Robin.....	20
Gambar 2.8 Mininet High Level Layer Diagram.....	22
Gambar 2.9 Topologi bawaan pada Mininet.....	23
Gambar 3.1 Perancangan Topologi.....	27
Gambar 3.2 Alur Penelitian	29
Gambar 3.3 Menjalankan Opendaylight.....	36
Gambar 3.4 Tampilan Login Opendaylight.....	37
Gambar 3.5 Tampilan Menu Opendaylight	37
Gambar 3.6 Menjalankan Mininet	38
Gambar 3.7 Uji Konektivitas Mininet	39
Gambar 3.8 Tampilan Miniedit.....	39
Gambar 3.9 Menjalankan mn-topo single, 3.....	40
Gambar 3.10 Uji Konektivitas Mininet dengan Opendaylight.....	41
Gambar 3.11 Pengujian <i>Throughput</i> Tidak Menggunakan <i>Load balancing</i> (1 client).....	43
Gambar 3.12 Pengujian <i>Throughput</i> Tidak Menggunakan <i>Load balancing</i> (2 client).....	44
Gambar 3.13 Pengujian <i>Throughput</i> Tidak Menggunakan <i>Load balancing</i> (3 client).....	45
Gambar 3.14 Menjalankan <i>Load balancing</i> Round Robin	47
Gambar 3.15 Pengujian <i>Throughput</i> Menggunakan <i>Load balancing</i> (1 client)....	48
Gambar 3.16 Pengujian <i>Throughput</i> Menggunakan <i>Load balancing</i> (2 client)....	49
Gambar 3.17 Pengujian <i>Throughput</i> Menggunakan <i>Load balancing</i> (3 client)	50
Gambar 3.18 Pengujian <i>Latency</i> Tidak Menggunakan <i>Load balancing</i> (1 client).....	52
Gambar 3.19 Pengujian <i>Latency</i> Tidak Menggunakan <i>Load balancing</i>	

(2 client).....	53
Gambar 3.20 Pengujian <i>Latency</i> Tidak Menggunakan <i>Load balancing</i> (3 client).....	54
Gambar 3.21 Menjalankan <i>Load balancing</i> Round Robin	56
Gambar 3.22 Pengujian <i>Latency</i> Menggunakan <i>Load balancing</i> (1 client)	57
Gambar 3.23 Pengujian <i>Latency</i> Menggunakan <i>Load balancing</i> (2 client)	58
Gambar 3.24 Pengujian <i>Latency</i> Menggunakan <i>Load balancing</i> (3 client)	59
Gambar 4.1 Grafik data hasil pengujian <i>Throughput</i> Menggunakan <i>Load</i> <i>Balancing</i> Round Robin.....	64
Gambar 4.2 Grafik data hasil pengujian <i>Throughput</i> Tidak Menggunakan <i>Load</i> <i>Balancing</i>	65
Gambar 4.3 Grafik perbandingan data hasil pengujian <i>Throughput</i>	66
Gambar 4.4 Grafik data hasil pengujian <i>Latency</i> Menggunakan <i>Load</i> <i>Balancing</i> Round Robin.....	67
Gambar 4.5 Grafik data hasil pengujian <i>Latency</i> Tidak Menggunakan <i>Load balancing</i>	68
Gambar 4.6 Grafik perbandingan data hasil pengujian <i>Latency</i>	69

INTISARI

Peningkatan jumlah pengguna pada internet, akan berpengaruh terhadap kinerja pada server. Semakin banyak yang mengakses akan membuat beban kerja yang lebih pada server dan menjadi kurang optimal. Banyaknya jumlah request pada suatu server akan mengakibatkan terjadinya overloading bahkan kemungkinan server akan mengalami down. Oleh karena itu dibutuhkan penyeimbang beban server yang akan mengatur dan membagi beban kerja server.

Pengimplementasian *load balancing* pada server dapat dilakukan untuk menangani masalah penggunaan jaringan yang semakin kompleks tersebut. Algoritma *load balancing* yang digunakan pada penelitian ini adalah Algoritma round robin merupakan algoritma yang bekerja dengan cara membagi beban secara bergiliran dan berurutan dari satu server ke server lainnya. Konsep dasar dari algoritma Round Robin ini adalah dengan menggunakan time sharing, pada intinya algoritma ini memproses antrian secara bergiliran.

Pengujian telah dilakukan, hasilnya Algoritma Round Robin dalam menghandle request dari *client* lebih bagus dibandingkan tanpa menggunakan *Load balancing* karena *Load balancing* Round Robin akan secara otomatis membagi beban traffic ke setiap server secara berurutan.

Kata Kunci: Internet, *load balancing*, sdn, round robin.

ABSTRACT

Increasing the number of users on the internet, will affect the performance on the server. More and more accesses will make more workload on the server and be less than optimal. The large number of requests on a server will result in overloading and even the possibility that the server will go down. Therefore we need a server load balancer that will manage and share server workloads.

Implementation of load balancing on the server can be done to handle the problem of the increasingly complex network usage. Load balancing algorithm used in this study is the round robin algorithm is an algorithm that works by dividing the load in turns and sequentially from one server to another server. The basic concept of the Round Robin algorithm is to use time sharing, in essence this algorithm processes the queue in turns.

Testing has been done, the results of Round Robin Algorithm in handling requests from clients is better than without using Load balancing because Load Robin Load balancing will automatically divide the traffic load to each server in sequence.

Keyword: *Internet, load balancing, sdn, round robin.*