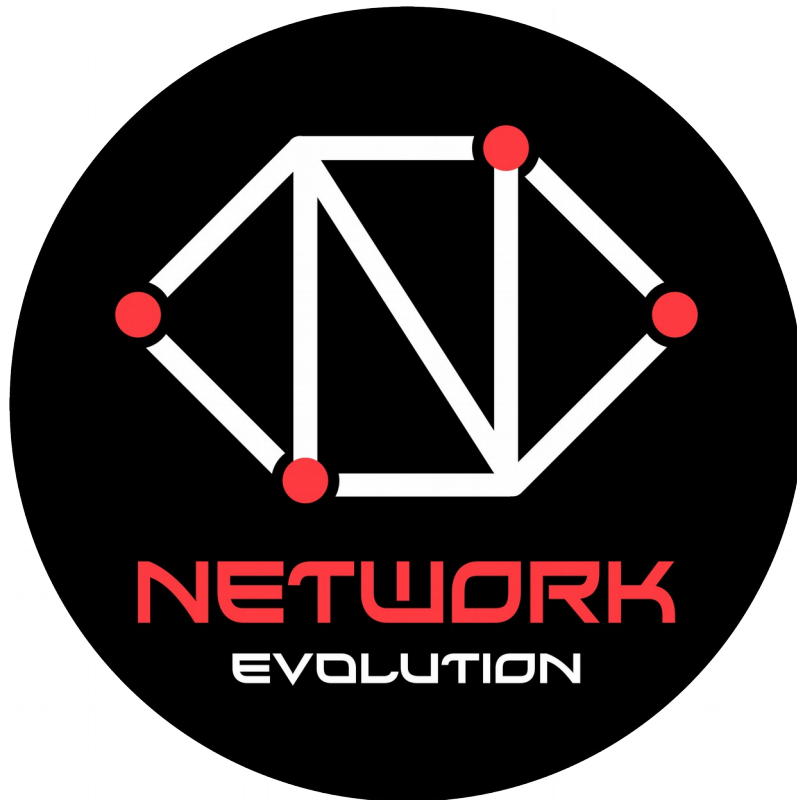


**ANALISIS QOS LOAD BALANCER DENGAN ALGORITMA
ROUND ROBIN PADA WEB SERVER APACHE**



Atwatan Malik Mahardi
S1TT05A – 17101008 – Divisi NFV

PROGRAM STUDI S1 TEKNIK TELEKOMUNIKASI
FAKULTAS TEKNIK TELEKOMUNIKASI DAN ELEKTRO
INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2019

BAB I

DASAR TEORI

1. Load Balancing

Load balancing merupakan suatu cara untuk mendistribusikan beban kerja secara merata ke beberapa node yang bekerja pada back-end node^[1]. Load balancing mengoptimalkan koneksi jaringan agar tidak kelebihan beban pada salah satu jalur koneksi serta berfungsi sebagai penggabung 2 buah jaringan atau lebih yang akan disambungkan ke router yang terhubung ke sebuah server dan client^[2].

2. Algoritma Round Robin

Algoritma round robin merupakan algoritma sederhana yang membagi beban secara bergiliran dan berurutan dari satu *server* ke *server* lainnya sehingga membentuk suatu siklus^[3].

3. Quality of Service

QoS merupakan kemampuan dalam menjamin pengiriman arus data penting atau bisa diartikan sebagai kumpulan dari berbagai kriteria performansi yang menentukan tingkat kepuasan penggunaan suatu layanan.

Parameter QoS :

➤ Throughput

Throughput merupakan kecepatan (*rate*) transfer data efektif, yang diukur dalam *bit per second*. Throughput merupakan jumlah total kedatangan paket yang sukses yang diamati pada destination selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

➤ Packet Loss

Packet Loss merupakan parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena *collision* dan *congestion* pada jaringan.

➤ Delay

Delay merupakan waktu yang dibutuhkan data untuk menempuh jarak dari titik asal ke tujuan.

➤ Jitter

Jitter menunjukkan banyaknya variasi delay pada taransmisi data di jaringan.^[4]

BAB II

MANFAAT DAN TUJUAN

A. MANFAAT

- 1) Dapat menerapkan *load balancer* dalam membagi beban kerja *web server* apache.
- 2) Dapat menggunakan algoritma round robin dalam membagi beban kerja *web server* apache.
- 3) Dapat menentukan performa terbaik *load balancer* algoritma round robin dalam membagi beban kerja *web server* apache.

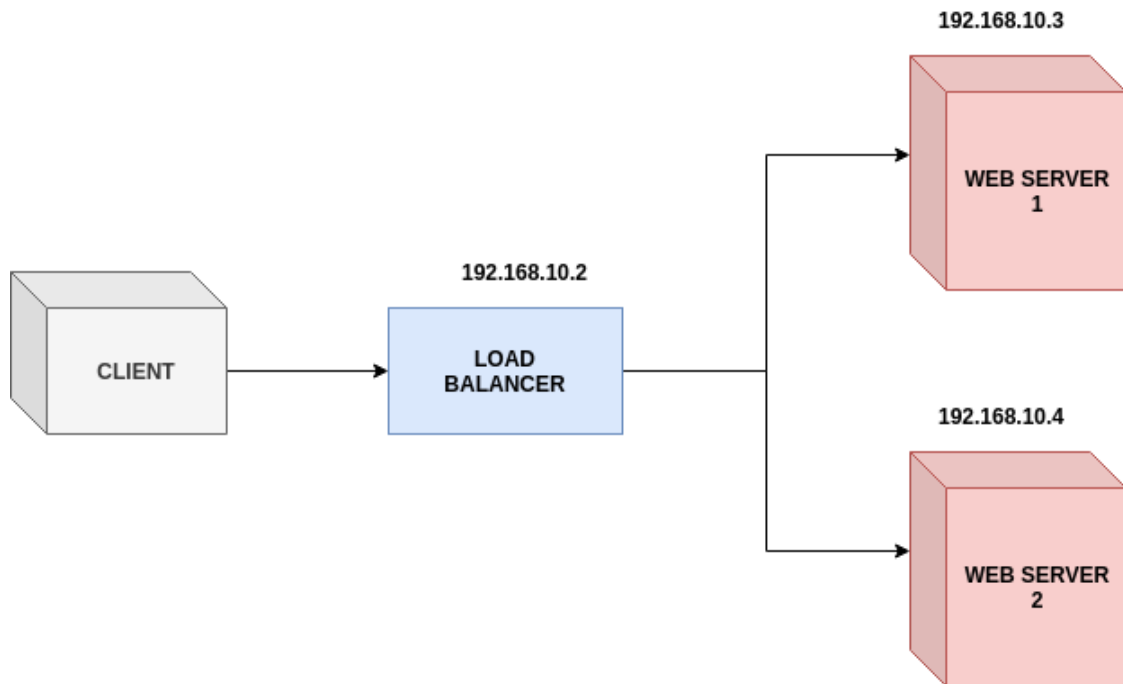
B. TUJUAN

- 1) Mengetahui peran *load balancer* dalam membagi beban kerja *web server* apache.
- 2) Mengetahui cara kerja algoritma round robin dalam membagi beban kerja *web server* apache.
- 3) Mengetahui performansi terbaik *load balancer* algoritma round robin dalam membagi beban kerja *web server* apache

BAB III

KONFIGURASI PERANGKAT

Penelitian ini menggunakan 3 buah *Virtual Machine* yang masing-masing berfungsi sebagai 1 *Load Balancer* dan 2 sebagai *Web Server* seperti pada topologi di bawah.



Gambar 3.1. Topologi *Load Balancing Web Server*.

1) Persiapan

- *Virtual Machine 1 : Load Balancer (OS : Ubuntu Server 18+HAProxy)*

Hostname : haproxy

IP Address : 192.168.10.2

- *Virtual Machine 2 : Web Server 1 (OS : Ubuntu Server 18+Apache)*

Hostname : web1

IP Address : 192.168.10.3

- *Virtual Machine 3 : Web Server 2 (OS : Ubuntu Server 18+Apache)*

Hostname : web2

IP Address : 192.168.10.4

- Wireshark

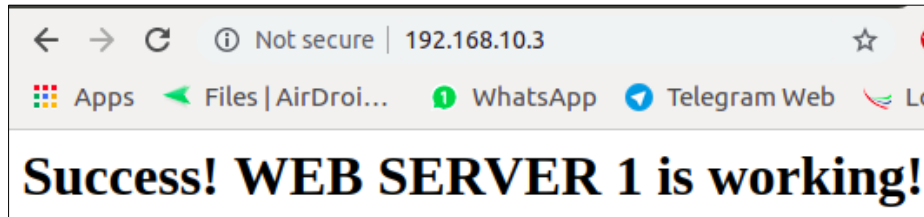
*Asumsi semua bahan telah terkonfigurasi dan tersedia

BAB IV

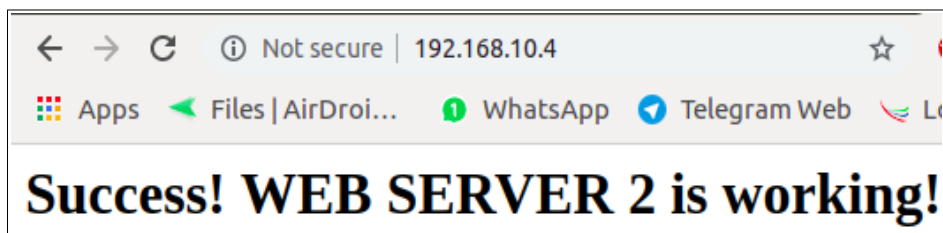
HASIL DATA DAN ANALISIS

1. Uji Coba Load Balancing.

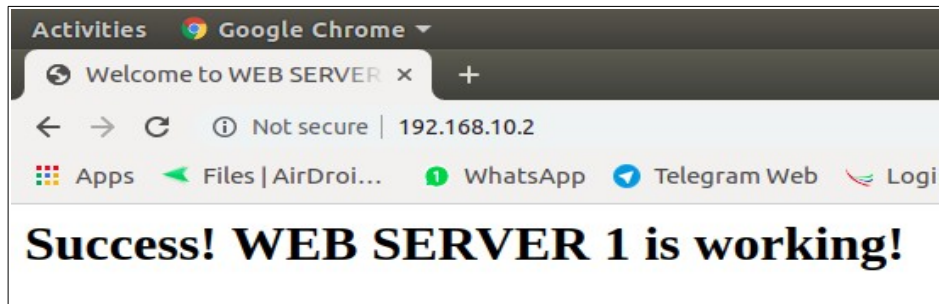
- 1) Uji Coba *Load Balancing* dengan mengakses kedua *Web Server* menggunakan 1 alamat IP *Load Balance*.



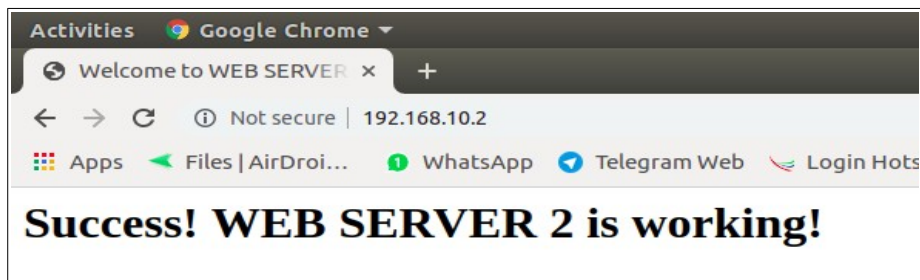
Gambar 4.1. Tes Alamat IP Web Server 1.



Gambar 4.2. Tes Alamat IP *Web Server* 2.

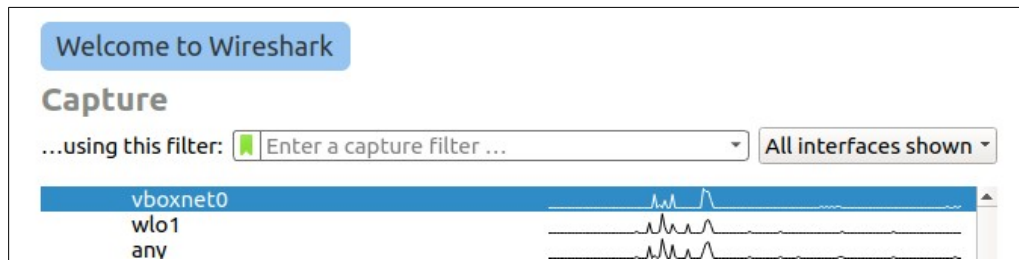


Gambar 4.3 Tes Alamat IP Load Balancer 1.



Gambar 4.4 Tes Alamat IP Load Balancer 2.

- 2) Buka Wireshark dan Start Capturing Packets pada interface vboxnet0.



2. Analisis QoS.

- 1) Analisis QoS pada salah satu web server tanpa Load Balancer.

Throughput :

- Pilih IP salah satu *web server* pada *Source*

No.	Time	Source	Destination	Protocol	Length
32	48.280714779	192.168.10.1	192.168.10.3	HTTP	59
33	48.280773899	192.168.10.3	192.168.10.1	TCP	60
34	48.280900306	192.168.10.3	192.168.10.1	TCP	60
35	48.280914506	192.168.10.1	192.168.10.3	TCP	60
36	48.281480518	192.168.10.3	192.168.10.1	HTTP	52
37	48.281497855	192.168.10.1	192.168.10.3	TCP	60

- Pilih *Capture File Properties* pada menu *Statistic*.

Wireshark - Capture File Properties - vboxnet0				
Details				
Application: Dumpcap (Wireshark) 2.6.8 (Git v2.6.8 packaged as 2.6.8-1~ubuntu18.04.0)				
Interfaces				
Interface	Dropped packets	Capture filter	Link type	Packet size limit
vboxnet0	0 (0 %)	none	Ethernet	262144 bytes
Statistics				
Measurement	Captured	Displayed	Marked	
Packets	37	37 (100.0%)	—	
Time span, s	48.281	48.281	—	
Average pps	0.8	0.8	—	
Average packet size, B	173	173	—	
Bytes	6418	6418 (100.0%)	0	
Average bytes/s	132	132	—	
Average bits/s	1,063	1,063	—	

- Menghitung *Throughput*.

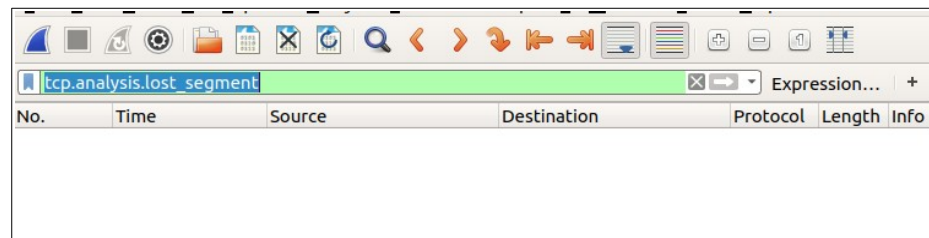
$$\begin{aligned}
 \text{Throughput} &= \text{Bytes} / \text{Time span} \\
 &= 6418 / 48,281 \\
 &= 132,9301 \text{ bytes/s} = 132,9301 \times 8 = 1063,4408 \text{ bit/s}
 \end{aligned}$$

Packet Loss :

- Filter *tcp.analysis.lost_segment* untuk mendapatkan nilai *packet loss*.
- Jika tidak ditemukan nilainya, maka *Packet Loss* = 0.

Delay :

- Filter tcp, kemudian *Export Packet Dissection as CSV* pada Menu File.



- Buka File ekstensi CSV dengan Libre Calc / MS. Excel.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.043056457	192.168.10.1	192.168.10.3	TCP	66	80 > 35170 [FIN, ACK] Seq=1 Ack=1 Win=235 Len=0 TSval=1330554508 TSecr=1925557355
2	0.043056457	192.168.10.1	192.168.10.3	TCP	66	35170 > 80 [ACK] Seq=1 Ack=2 Win=237 Len=0 TSval=1925562403 TSecr=1330554508
3	2.566229456	192.168.10.1	192.168.10.3	TCP	66	35170 > 80 [FIN, ACK] Seq=1 Ack=2 Win=237 Len=0 TSval=1925564927 TSecr=1330554508
4	2.566419733	192.168.10.1	192.168.10.3	TCP	74	35174 > 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1925564927 TSecr=0 WS=128
5	2.566460276	192.168.10.1	192.168.10.3	TCP	66	80 > 35170 [ACK] Seq=2 Ack=2 Win=235 Len=0 TSval=1330557074 TSecr=1925564927
6	2.566577756	192.168.10.3	192.168.10.1	TCP	74	80 > 35174 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1330557074 TSecr=1925564927 WS=128
7	2.566608228	192.168.10.1	192.168.10.3	TCP	66	35174 > 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1925564927 TSecr=1330557074
8	2.567777665	192.168.10.1	192.168.10.3	HTTP	598	GET / HTTP/1.1
9	2.567957966	192.168.10.3	192.168.10.1	TCP	66	80 > 35172 [ACK] Seq=1 Ack=533 Win=235 Len=0 TSval=1330557076 TSecr=1925564928
10	2.568555674	192.168.10.3	192.168.10.1	HTTP	522	HTTP/1.1 200 OK (text/html)
11	2.568571579	192.168.10.1	192.168.10.3	TCP	66	35172 > 80 [ACK] Seq=533 Ack=457 Win=237 Len=0 TSval=1925564929 TSecr=1330557076
12	3.231609446	192.168.10.1	192.168.10.3	HTTP	598	GET / HTTP/1.1
13	3.232199631	192.168.10.3	192.168.10.1	HTTP	521	HTTP/1.1 200 OK (text/html)
14	3.232229609	192.168.10.1	192.168.10.3	TCP	66	35172 > 80 [ACK] Seq=1065 Ack=912 Win=245 Len=0 TSval=1925565593 TSecr=1330557740
15	3.732317668	192.168.10.1	192.168.10.3	HTTP	598	GET / HTTP/1.1

- Menghitung *Delay*.

$Delay = Rata-Rata (Time\ 2 - Time\ 1) \rightarrow$ Perhitungan menggunakan Excel

$$= 1,42004405455882\ s$$

Jitter :

- *Jitter* = $Rata-Rata (Delay\ 1 - Delay\ 2) \rightarrow$ Perhitungan menggunakan Excel

$$= -0,001304215757576\ s$$

- 2) Analisis QoS pada salah satu web server tanpa Load Balancer dengan 1000 request.

- 1) Menggunakan *apache benchmark* pada client.
- 2) Melakukan 1000 request dengan 100 pengakses yang request bersamaan.

```
atmlkmh@ibnmahar:~$ ab -n 1000 -c 100 http://192.168.10.3/
This is ApacheBench, Version 2.3 <Revision: 1807734 >
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.10.3 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests
```

3) Seperti cara analisis sebelumnya, didapatkan hasil :

- Throughput : 97122,677783624 bit/s
- Packet Loss : 1
- Delay : 0,018357248537627 s
- Jitter : 0.00000003851 s

3) Analisis QoS pada Load Balancer.

Seperti cara analisis sebelumnya namun dengan menggunakan IP *Load Balancer*, didapatkan hasil :

- Throughput : 23602,771362587 bit/s
- Packet Loss : 0
- Delay : 0,101905449176471 s
- Jitter : -0,000076123 s

4) Analisis QoS pada Load Balancer dengan 1000 request.

Seperti cara analisis sebelumnya namun dengan menggunakan IP *Load Balancer*, didapatkan hasil :

- Throughput : 666226,836807482 bit/s
- Packet Loss : 0
- Delay : 0.00160077675019 s
- Jitter : -0.00000000508540 s

Tabel 4.1. Standarisasi Throughput^[5].

Kategori	Throughput(bit/s)	Index
Terbaik	>2100000	4
Lebih Baik	1200000 - 2100000	3
Baik	700000 – 1200000	2
Cukup Baik	338000 - 700000	1
Buruk	0 - 338000	0

Tabel 4.2. Standarisasi Packet Loss^[5].

Kategori	Packet Loss (%)	Index
Sangat Bagus	0	4
Bagus	3	3

Sedang	15	2
Jelek	25	1

Tabel 4.3. Standarisasi Delay^[5].

Kategori	Delay (s)	Index
Sangat Bagus	<0,15	4
Bagus	0,15 - 0,3	3
Sedang	0,3 – 0,45	2
Jelek	>0,45	1

Tabel 4.4. Standarisasi Jitter^[5].

Kategori	Jitter (s)	Index
Sangat Bagus	<0	4
Bagus	0 - 0,075	3
Sedang	0,075 – 0,125	2
Jelek	>0,125	1

3. Hasil data dan penjelasan.

1. Throughput

Saat mendapat 1000 request, web server tanpa load balancer memiliki nilai throughput 97122,677783624 yang masuk ke dalam kategori Buruk, sedangkan load balancer memiliki nilai throughput 666226,836807482 yang masuk ke dalam kategori Cukup Baik.

2. Packet loss

Saat mendapat 1000 request, web server tanpa load balancer memiliki 1 paket loss, sedangkan load balancer tidak memiliki packet loss.

3. Delay

Saat mendapat 1000 request, web server tanpa load balancer memiliki nilai delay sebesar 0,018357248537627 yang masuk ke dalam kategori Sangat Bagus, sedangkan load balancer memiliki nilai delay 0.00160077675019 yang masuk ke dalam kategori Sangat Bagus juga tetapi dengan nilai delay yang lebih kecil.

4. Jitter

Saat mendapat 1000 request, web server tanpa load balancer memiliki nilai jitter 0.00000003851 yang masuk ke dalam kategori Sangat Bagus, sedangkan load balancer memiliki nilai jitter -0.0000000508540 yang masuk ke dalam kategori Sangat Bagus juga tetapi dengan nilai yang lebih kecil.

BAB IV

KESIMPULAN

Web Server menggunakan Load Balancer unggul dalam parameter Throughput, Packet Loss, Delay, dan Jitter dibandingkan Web Server tanpa Load Balancer. Walaupun terdapat kesamaan kategori di parameter Delay dan Jitter, tetapi Load Balancer tetap unggul jika diukur menggunakan nilai terkecil. Berdasarkan penelitian ini dapat disimpulkan bahwa Load Balancer mampu menghasilkan nilai throughput yang tinggi tanpa packet loss, serta latency(delay) dan jitter yang rendah.

DAFTAR PUSTAKA

- [1] M. R. Pratama, Hafidudin, and S. Aulia, "ANALISIS PERFORMANSI LOAD BALANCING DENGAN ALGORITMA ROUND ROBIN DAN LEAST CONNECTION PADA SEBUAH WEB SERVER," *e-Proceeding Appl. Sci.*, vol. 1, p. 1577, 2015.
- [2] I. M. A. S. Darma and I. G. O. G. Atitama, "Implementasi Load Balancing Pada Openstack dengan Metode Round Robin," *Pros. Semin. Nas. Pendidik. Tek. Inform. Ke-8*, 2017.
- [3] K. Ansharullah, "IMPLEMENTASI SISTEM LOAD BALANCING DENGAN ALGORITMA ROUND ROBIN UNTUK MENGATASI BEBAN SERVER DI SMK NEGERI 2 KUDUS," UNIVERSITAS NEGERI SEMARANG, 2016.
- [4] M. Anjani, A. Wijaya, and F. Panjaitan, "IMPLEMENTASI LOAD BALANCING DENGAN METODE ECMP (EQUAL COST MULTI PATH) STUDI KASUS : SMK TEKNOLOGI BISTEK PALEMBANG."
- [5] M. S. Adnan, S. Ikhwan, and Yosy Rahmawati, "Implementasi Load Balancing Metode ECMP, NTH dan PCC dengan Empat Link Internet Menggunakan Mikrotik," *CENTIVE 2018*, 2018.