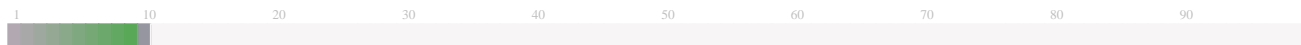


Submission Information

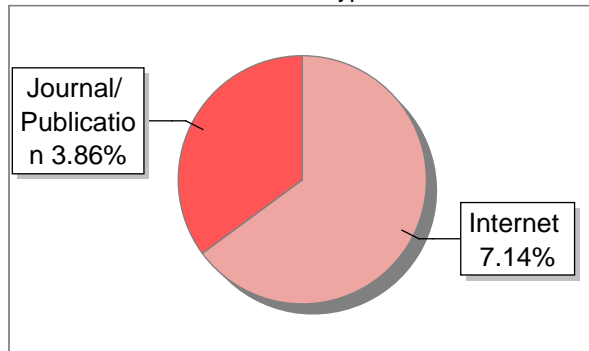
Author Name	Jeffri Alfa Razaq
Title	Monitoring Load Balancing Server E-Learning Unisbank Menggunakan Metode Least
Paper/Submission ID	2247571
Submitted by	atikrahma@edu.unisbank.ac.id
Submission Date	2024-08-20 22:41:21
Total Pages, Total Words	9, 2930
Document type	Article

Result Information

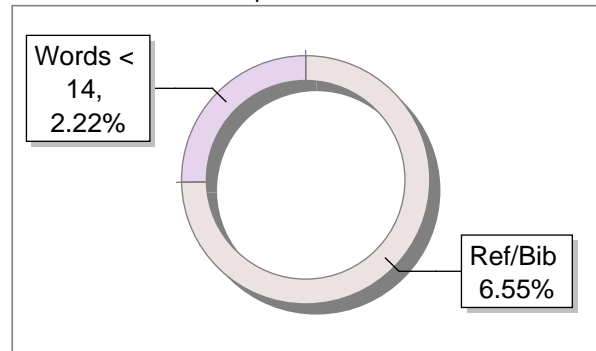
Similarity **11 %**



Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

Database Selection

Language	Non-English
Student Papers	No
Journals & publishers	No
Internet or Web	Yes
Institution Repository	No

A Unique QR Code use to View/Download/Share Pdf File



DrillBit Similarity Report

11

SIMILARITY %

11

MATCHED SOURCES

B

GRADE

A-Satisfactory (0-10%)**B-Upgrade (11-40%)****C-Poor (41-60%)****D-Unacceptable (61-100%)**

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	journal.stekom.ac.id	4	Publication
2	adoc.pub	2	Internet Data
3	adoc.pub	1	Internet Data
4	adoc.pub	1	Internet Data
5	docplayer.info	1	Internet Data
6	en.nsd.pku.edu.cn	1	Internet Data
7	journal.stekom.ac.id	<1	Internet Data
8	docplayer.info	<1	Internet Data
9	docplayer.info	<1	Internet Data
10	e-journal.unmas.ac.id	<1	Internet Data
11	repository.unika.ac.id	<1	Internet Data

MONITORING LOAD BALANCING SERVER E-LEARNING UNISBANK MENGGUNAKAN METODE LEAST

Jeffri Alfa Gazaq¹, Dwi Budi Santoso², Fatkhul Amin³

^{1,3}Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank
Jl. Trilomba Juang No.1 Semarang, 0248311668, e-mail: mrjf@edu.unisbank.ac.id,
fatkhulam8@edu.unisbank.ac.id

²Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank
Jl. Trilomba Juang No.1 Semarang, 0248311668, e-mail: dbs@edu.unisbank.ac.id

ARTICLE INFO

Article history:

Received 18 November 2022
Received in revised form 24 November 2022
Accepted 26 November 2022
Available online 12 July 2023

ABSTRACT

Implementation of Distance Learning (PJJ) is effective in the teaching and learning process during the pandemic. This is supported by Unisbank students in mastering information technology. PJJ technology, also known as e-learning, is a bridge between lecturers and students in the process of meeting in class into a meeting process in cyberspace. E-learning becomes ineffective when the demand for e-learning usage is high, causing server overload or crashes. When the e-learning server is overloaded, all parts that are usually connected and interrelated, such as lecturers, students, administration and others, become disconnected. The importance of monitoring the e-learning server to keep the e-learning load under control, and the server that has a lot of load will be divided so that e-learning will run smoothly and not be overloaded. Monitoring the Elearning server with the Least Connection method is applied to the Unisbank e-learning server and makes e-learning not overloaded. The results of the load balancer workload test with 1000 connections, the resulting response time is 0.2 ms, the workload is 5000 connections, the response time is 0.2 ms and the workload is 10,000 connections, the resulting reply time is 0.2 ms.

Keywords : Least Connection, elearning unisbank, load balancing

1. Introduction

Teknologi Informasi di era Pandemi Corona saat ini sangat mendukung untuk dilaksanakannya pembelajaran jarak jauh (pjj) di semua jenjang pendidikan. Pentingnya dukungan Hardware, software dan brainware untuk keberhasilan pelaksanaan PJJ. Dukungan Server sebagai penopang utama dalam pelaksanaan PJJ untuk sebuah kampus menjadi keharusan bagi institusi pendidikan. Server Elearning Unisbank adalah salah satu yang menjadi perhatian penting dalam era PJJ saat ini. Kebutuhan internet yang semakin meningkat membuat pengguna internet membutuhkan bandwidth yang besar. Sehingga tidak jarang di perusahaan, intitusi pendidikan, wara bahkan di perumahan menerapkan multihoming untuk memenuhi kebutuhannya. Jaringan multihoming merupakan suatu sistem jaringan yang memiliki jalur keluar lebih dari satu.

3 Berawal dari sebuah single server yang selalu mendapatkan request dari banyak user, perlahan tapi pasti akan terjadi overload dan crash sehingga berdampak pada request yang tidak dapat dilayani oleh single server. Desain arsitektur cluster dapat di sederhana untuk proses dalam sistem operasi. Round Robin ditugaskan untuk membagi waktu setiap proses pada porsi yang sama dan dalam 2 utan melingkar, menjalankan semua proses tanpa prioritas dikenal juga sebagai eksekutif siklik. Load Balancing adalah teknik untuk mendistribusikan beban trafik koneksi pada dua atau lebih 2 jalur koneksi secara seimbang agar trafik dapat berjalan optimal, memaksimalkan troughput, 4 memperkecil waktu tanggap dan menghindari overload pada salah satu jalur koneksi[2]. Jaringan Komputer adalah kumpulan dua atau lebih komputer yang saling berhubungan untuk melakukan komunikasi data. Komunikasi data yang bisa dilakukan melalui Jaringan Komputer dapat berupa data teks, gambar, video dan suara. Untuk membangun sebuah jaringan komputer harus diperhatikan tentang situasi dan kondisi organisasi yang akan membangun jaringan 1 tersebut. Adapun solusi untuk mengatasi masalah overload dan crash ini adalah dengan membuat Monitoring Load Balancing Server E-learning Unisbank Menggunakan Metode Least Connection.

Optimalisasi Jaringan Komputer Menggunakan Algoritma Load Balancing di Virtual Server Citrix ADC, Meningkatnya jumlah pengguna internet harus diimbangi dengan kesiapan kapasitas infrastruktur jaringan dan server. Di era digital dan merebaknya pandemi COVID-19 memaksa semua aktifitas dilakukan secara online[5]. Apabila jumlah pengguna yang mengakses server melebihi kapasitas infrastruktur jaringan dan server maka yang terjadi adalah server down. Diperlukan perangkat load balancer untuk membagi beban traffic request dari pengguna menuju server. Penelitian ini membandingkan tiga algoritma pada load balancer Citrix ADC VPX: least connection, least response time, dan round-robin. Menggunakan empat skenario yang berbeda pada GNS3, pertama adalah topologi yang hanya menggunakan single web server, kedua menggunakan load balancer, empat web server dan algoritma round-robin, ketiga algoritma diganti menjadi least connection, dan terakhir algoritma least response time. Hasil pengujian parameter response time, throughput dan cpu utilization menunjukkan algoritma least connection lebih unggul. Terdapat penurunan response time sebesar 33%, cpu utilization 75% dan kenaikan throughput 53%. Pada parameter service hits, algoritma round-robin memiliki distribusi traffic yang paling merata. Maka algoritma dengan response time, throughput dan cpu utilization terbaik adalah least connection. Algoritma dengan service hits terbaik adalah round-robin. Implementasi skala besar direkomendasikan menggunakan algoritma least connection mengacu pada parameter response time, throughput dan cpu utilization. Apabila menekankan pada distribusi yang paling merata gunakan algoritma round-robin.

Penelitian terdahulu dilakukan oleh Ari Wibowo[3] Load Balancing Pada Cloud Computing Menggunakan Metode Least Connection, Load balancing adalah suatu teknik didalam lingkungan cloud computing yang memiliki tujuan utama untuk menyeimbangkan load server dalam menangani request. Load balancing juga merupakan salah satu teknik untuk meningkatkan performasi server secara keseluruhan dengan menyeimbangkan request yang diproses oleh server sehingga server mampu menyelesaikan request secara efektif dan efisien. Terdapat banyak library maupun tools yang dapat digunakan untuk menganalisis sistem cloud salah satunya adalah cloudsims. Pada tugas akhir disimulasikan dan dianalisis mengenai metode Least Connection dalam menangani load balancing pada sistem cloud yang disimulasikan pada cloudsims. Hasil simulasi menghasilkan hasil yang cukup baik yakni pemerataan CPU load sekitar 28% dari seluruh VM yang digunakan, kemudian tingkat pemerataan beban jaringan juga yang cukup baik dimana kecepatan rata rata proses yang mampu diselesaikan oleh masing masing server mampu diterima client dalam waktu yang relatif singkat (sekitar 40ms). Kecepatan proses akhir secara keseluruhan terlihat bahwa server mampu memberi waktu penyelesaian yang hampir sama bagi proses yang memiliki bobot yang sama (hanya terdapat selisih maksimal 40 mili second). Dengan demikian dapat disimpulkan bahwa algoritma Least Connection adalah algoritma yang layak digunakan oleh cloud server yang memiliki skalabilitas dan workload yang tinggi.

Penelitian tentang Least Connection juga dilakukan oleh Agung Nugroho[1] Analisis Perbandingan Performa Algoritma Round Robin dan Least Connection untuk Load Balancing pada Software Defined Network, Perkembangan teknologi internet dimana para penggunanya yang menginginkan layanan yang selalu tersedia setiap saat semakin meningkat mengikuti perkembangan zaman yang biasanya belum tentu didukung dengan bertambahnya layanan server yang memadai. Request pengguna yang terlalu banyak menjadikan beban kerja pada server meningkat dengan cepat, mengakibatkan server down dalam waktu yang singkat serta dibutuhkan teknologi yang menangani penggunaan jaringan yang kompleks. Sebagai teknologi baru, software defined network menawarkan scalability dan programmability untuk penggunaan jaringan yang semakin kompleks seperti load balancing. Beberapa penelitian tentang load balancing pada software defined network (SDN) dengan berbagai algoritma yang diterapkan akan menghasilkan performa yang berbeda-beda, diantaranya yaitu algoritma round robin dan least connection. Pengujian dilakukan dengan 3 kategori rate yaitu low, medium, dan high. Parameter pengujian yang digunakan adalah throughput, response time, dan CPU Usage menggunakan Httperf dan psutil. Algoritma round robin lebih unggul dibandingkan algoritma least connection pada koneksi kecil. Sedangkan untuk koneksi besar algoritma least connection lebih unggul. Nilai rata-rata response time menunjukkan keunggulan algoritma round robin dibandingkan dengan algoritma least connection. CPU Usage server dengan algoritma round robin lebih stabil pada setiap kategori rate. Sedangkan algoritma least connection lebih ringan beban dari algoritma round robin meskipun nilainya mengalami kenaikan pada setiap kategori rate. Penelitian tentang load balancing juga dilakukan oleh Hardiyan dkk[4] Optimalisasi Jaringan Komputer Menggunakan Algoritma Load Balancing di Virtual Server Citrix ADC, Meningkatnya jumlah pengguna internet harus diimbangi dengan kesiapan kapasitas infrastruktur jaringan dan server. Di era digital dan merebaknya pandemi COVID-19 memaksa semua aktifitas dilakukan secara online. Apabila jumlah pengguna yang mengakses server melebihi kapasitas infrastruktur jaringan dan server maka yang terjadi adalah server down. Diperlukan perangkat load balancer untuk membagi beban traffic request dari pengguna menuju server. Penelitian ini membandingkan tiga algoritma pada load balancer Citrix ADC VPX: least connection, least response time, dan round-robin. Menggunakan empat skenario yang berbeda pada GNS3, pertama adalah topologi yang hanya menggunakan single web server, kedua menggunakan load balancer, empat web server dan algoritma round-robin, ketiga algoritma diganti menjadi least connection, dan terakhir algoritma least response time. Hasil pengujian parameter response time, throughput dan cpu utilization menunjukkan algoritma least connection lebih unggul. Terdapat penurunan response time sebesar 33%, cpu utilization 75% dan kenaikan throughput 53%. Pada parameter service hits, algoritma round-robin memiliki distribusi traffic yang paling merata. Maka algoritma dengan response time, throughput dan cpu utilization terbaik adalah least connection. Algoritma dengan service hits terbaik adalah round-robin. Implementasi skala besar direkomendasikan menggunakan algoritma least connection mengacu pada parameter response time, throughput dan cpu utilization. Apabila menekankan pada distribusi yang paling merata gunakan algoritma round-robin.

Penelitian terdahulu dilakukan oleh Ari Wibowo[3] Load Balancing Pada Cloud Computing Menggunakan Metode Least Connection, Load balancing adalah suatu teknik didalam lingkungan cloud computing yang memiliki tujuan utama untuk menyeimbangkan load server dalam menangani request. Load balancing juga merupakan salah satu teknik untuk meningkatkan performa server secara keseluruhan dengan menyeimbangkan request yang diproses oleh server sehingga server mampu menyelesaikan request secara efektif dan efisien. Terdapat banyak library maupun tools yang dapat digunakan untuk menganalisis sistem cloud salah satunya adalah cloudsims. Pada tugas akhir disimulasikan dan dianalisis mengenai metode Least Connection dalam menangani load balancing pada sistem cloud yang disimulasikan pada cloudsims. Hasil simulasi menghasilkan hasil yang cukup baik yakni pemerataan CPU load sekitar 28% dari seluruh VM

yang digunakan, kemudian tingkat pemerataan beban jaringan juga yang cukup baik dimana kecepatan rata rata proses yang mampu diselesaikan oleh masing masing server mampu diterima client dalam waktu yang relatif singkat (sekitar 40ms). Kecepatan proses akhir secara keseluruhan terlihat bahwa server mampu memberi waktu penyelesaian yang hampir sama bagi proses yang memiliki bobot yang sama (hanya terdapat selisih maksimal 40 mili second). Dengan demikian dapat disimpulkan bahwa algoritma Least Connection adalah algoritma yang layak digunakan oleh cloud server yang memiliki skalabilitas dan workload yang tinggi.

Penelitian tentang Least Connection juga dilakukan oleh Agung Nugroho[1] Analisis Perbandingan Performa Algoritma Round Robin dan Least Connection untuk Load Balancing pada Software Defined Network, Perkembangan teknologi internet dimana para penggunanya yang menginginkan layanan yang selalu tersedia setiap saat semakin meningkat mengikuti perkembangan zaman yang biasanya belum tentu didukung dengan bertambahnya layanan server yang memadai. Request pengguna yang terlalu banyak menjadikan beban kerja pada server meningkat dengan cepat, mengakibatkan server down dalam waktu yang singkat serta dibutuhkan teknologi yang menangani penggunaan jaringan yang kompleks. Sebagai teknologi baru, software defined network menawarkan scalability dan programmability untuk penggunaan jaringan yang semakin kompleks seperti load balancing. Beberapa penelitian tentang load balancing pada software defined network (SDN) dengan berbagai algoritma yang diterapkan akan menghasilkan performa yang berbeda-beda, diantaranya yaitu algoritma round robin dan least connection. Pengujian dilakukan dengan 3 kategori rate yaitu low, medium, dan high. Parameter pengujian yang digunakan adalah throughput, response time, dan CPU Usage menggunakan Httperf dan psutil. Algoritma round robin lebih unggul dibandingkan algoritma least connection pada koneksi kecil. Sedangkan untuk koneksi besar algoritma least connection lebih unggul. Nilai rata-rata response time menunjukan keunggulan algoritma round robin dibandingkan dengan algoritma least connection. CPU Usage server dengan algoritma round robin lebih stabil pada setiap kategori rate. Sedangkan algoritma least connection lebih ringan beban dari algoritma round robin meskipun nilainya mengalami kenaikan pada setiap kategori *rate*.

Komponen Hardware

(1) Server Load Balancer :

- Processor Intel Xeon E5-2609 v4 1.7Ghz 4 Core
- RAM 4GB
- Gigabit Ethernet

(2) Web Server E-Learning

- Processor Intel Xeon E5-2609 v4 1.7Ghz 4 Core
- RAM 8GB
- Gigabit Ethernet

(3) Client PC

- Intel Core i5
- RAM 4GB
- Gigabit Ethernet

b. NGINX

NGINX adalah web server yang juga berfungsi sebagai email proxy, reverse proxy, dan load balancer. Struktur software ini bersifat asinkron dan event-driven; yang memungkinkan banyak request atau permintaan diproses pada waktu bersamaan. Selain itu, NGINX juga dapat diskalakan. Hal ini menunjukkan bahwa layanan ini tumbuh dan berkembang seiring dengan bertambahnya jumlah trafik yang datang ke website.

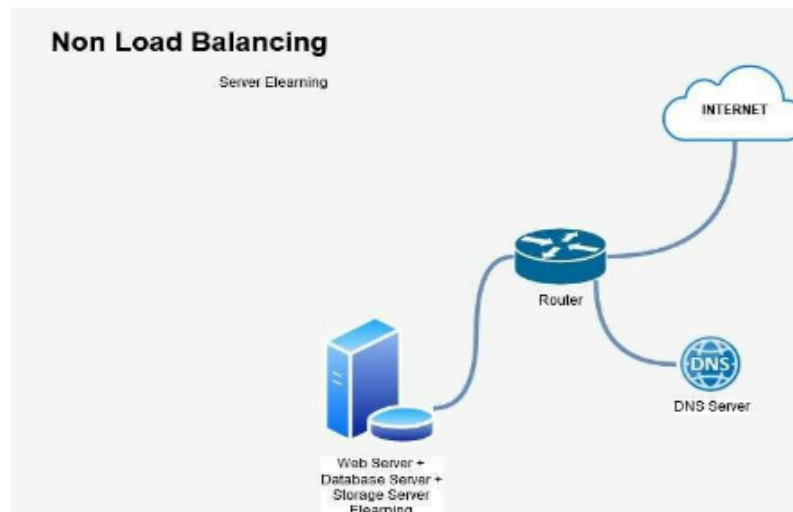
Fungsi utama Server atau Web server adalah untuk melakukan atau akan mentransfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa.

halaman web yang diminta terdiri dari berkas teks, video, gambar, file dan banyak lagi. pemanfaatan web server berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web termasuk yang di dalam berupa teks, video, gambar dan banyak lagi. Salah satu contoh dari Web Server adalah Apache. Apache (Apache Web Server – The HTTP Web Server) merupakan web server yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX.

2. Research Method

2.1. Arsitektur Informasi

Jaringan elearning yang dihubungkan dengan internet akan diteruskan kepada router yang akan disalurkan kembali kepada DNS Server dan Web Server, Database Server, Storage Server Elearning. Pada model semua permintaan pengguna akan didistribusikan langsung ke 1 server akan mengakibatkan server mengalami overload dan crash[6]. Hal ini diakibatkan karena banyaknya pengguna yang akses elearning dan server elearning hanya 1 sehingga terjadi penumpukan beban di server. Jika 1 server digunakan untuk banyak data seperti Web Server, Database Server, Storage Server Elearning maka kemungkinan terjadinya overload menjadi besar dan akan mengganggu jalannya elearning yang menggunakan server tersebut (gambar 1).

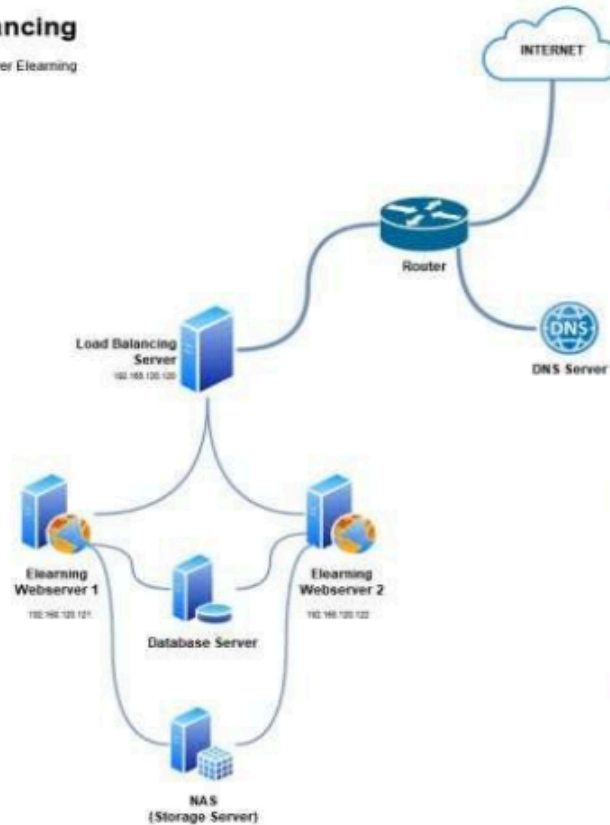


Gambar 1. Non Load Balancing

Pada penelitian load balancing ini konfigurasi sistem dari server Elearning Unisbank bisa dilihat pada gambar 2. Jaringan internet disalurkan melalui Router dan diteruskan kepada server load balancing. Kemudian oleh server load balancing dibuatlah pembagian atau didesai agar jaringan yang masuk dibagi kepada 2 server yang ada yaitu Elearning Webserver 1 dan Elearning Webserver 2. Semua data yang masuk di elearning selanjutnya akan dimasukkan tersistem pada database server dan akan disimpan juga pada storage server.

Load Balancing

Server Elearning



Gambar 2. Load Balancing

2.2. Metode Load Balancing Least Connection

Algoritma load balancing paling sederhana yaitu membagi beban secara bergiliran dan berurutan dari satu server ke server lain.. Teknik load balancing di beberapa instanse aplikasi adalah teknik yang umum digunakan untuk mengoptimalkan pemanfaatan resource, memaksimalkan throughput, mengurangi latensi, dan memastikan konfigurasi yang toleran terhadap kesalahan. Pada Penelitian ini, menggunakan NGINX yang difungsikan sebagai pembagi beban (load balancer). Konfigurasi bisadilihatpada gambar 3.

2. Least Connections – A request is sent to the server with the least number of active connections, again with *server weights* taken into consideration:

```
upstream backend {
    least_conn;
    server backend1.example.com;
    server backend2.example.com;
}
```

Gambar 3. Konfigurasi Load Balancing

3. Result and Analysis

3.1. Konfigurasi Load Balancing di NGINX

Berikut ini gambar 4 formulasi round robin pada server elearning Unisbank yang digunakan dalam penelitian.

```
http {
    upstream moodle {
        least_conn;
        server 192.168.120.238;
        server 192.168.120.204;
    }

    server {
        listen 80;
        location / {
            proxy_pass
            http://moodle;
        }
    }
}
```

Gambar 4. SourceCode Round Robin.

3.2. Uji Load Balancing

Pengujian dilakukan dengan menggunakan aplikasi httperf (baca : h-t-t-perf) untuk membangkitkan request ke server load balancing secara simultan dengan sejumlah koneksi pada port tujuan 80 (webserver) dan request url home page/Halaman depan. Uji Load Balancing dilakukan pada; 1000 koneksi, 5000 koneksi, 10000 koneksi.

```
httplib --hog --server 192.168.120.200 --num-conn 1000 --ra 1000 timeout 5
```

- a. Uji load balancing dengan workload dengan kali model yaitu dengan 1000 connection dengan hasil reply time 0.2 ms.

```

root@tik:~# httpperf --hog --server 192.168.120.200 --max-conn 1000 --qs 1000 timeout 5
httpperf --hog --client=0/1 --server=192.168.120.200 --port=80 --uri=/ --rate=1000 --send-buffer=4096 --recv-buffer=16384 --max-conn=1000 --max-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 2

Total: connections 1000 requests 1000 replies 1000 test-duration 1.000 s

Connection rate: 1000.1 conn/s (1.0 ms/conn, <=2 concurrent connections)
Connection time [ms]: min 0.1 avg 0.2 max 1.2 median 0.5 stddev 0.1
Connection time [ms]: connect 0.0
Connection length [replies/conn]: 1.000

Request rate: 1000.1 req/s (1.0 ms/req)
Request size [B]: 68.0

Reply rate [replies/s]: min 0.0 avg 0.0 max 0.0 stddev 0.0 (0 samples)
Reply time [ms]: response 0.2 transfer 0.0
Reply size [B]: header 157.0 content 157.0 footer 0.0 (total 314.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=0 5xx=1000

CPU time [s]: user 0.30 system 0.63 (user 29.9% system 63.3% total 93.1%)
Net I/O: 373.1 KB/s (3.1*10^6 bps)

Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 address-unavail 0 flush-full 0 other 0
root@tik:~#

```

Gambar 5. workload httpperf dengan 1000 connection

- b. Uji load balancing seperti terlihat pada gambar 6 dengan workload dengan kali model yaitu dengan 5000 connection, dengan hasil reply time 0.2 ms.

```

root@tik:~# httpperf --hog --server 192.168.120.200 --max-conn 5000 --qs 1000 timeout 5
httpperf --hog --client=0/1 --server=192.168.120.200 --port=80 --uri=/ --rate=1000 --send-buffer=4096 --recv-buffer=16384 --max-conn=5000 --max-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 10

Total: connections 5000 requests 5000 replies 5000 test-duration 5.005 s

Connection rate: 999.0 conn/s (1.0 ms/conn, <=10 concurrent connections)
Connection time [ms]: min 0.1 avg 0.2 max 3.4 median 0.5 stddev 0.1
Connection time [ms]: connect 0.0
Connection length [replies/conn]: 1.000

Request rate: 999.0 req/s (1.0 ms/req)
Request size [B]: 68.0

Reply rate [replies/s]: min 997.3 avg 997.3 max 997.3 stddev 0.0 (1 samples)
Reply time [ms]: response 0.2 transfer 0.0
Reply size [B]: header 157.0 content 157.0 footer 0.0 (total 314.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=0 5xx=5000

CPU time [s]: user 1.20 system 3.72 (user 24.0% system 74.4% total 98.4%)
Net I/O: 373.7 KB/s (3.1*10^6 bps)

Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 address-unavail 0 flush-full 0 other 0
root@tik:~#

```

Gambar 6. workload httpperf dengan 5000 connection

- c. Uji load balancing seperti terlihat pada gambar 7 dengan workload dengan kali model yaitu dengan 10000 connection, dengan hasil reply time 0.2 ms.

```

root@tik:~# httpperf --hog --server 192.168.120.200 --max-conn 10000 --qs 1000 timeout 5
httpperf --hog --client=0/1 --server=192.168.120.200 --port=80 --uri=/ --rate=1000 --send-buffer=4096 --recv-buffer=16384 --max-conn=10000 --max-calls=1
httpperf: warning: open file limit > FD_SETSIZE; limiting max. # of open files to FD_SETSIZE
Maximum connect burst length: 10

Total: connections 10000 requests 10000 replies 10000 test-duration 9.999 s

Connection rate: 1000.1 conn/s (1.0 ms/conn, <=15 concurrent connections)
Connection time [ms]: min 0.1 avg 0.2 max 13.9 median 0.5 stddev 0.2
Connection time [ms]: connect 0.0
Connection length [replies/conn]: 1.000

Request rate: 1000.1 req/s (1.0 ms/req)
Request size [B]: 68.0

Reply rate [replies/s]: min 999.9 avg 999.9 max 999.9 stddev 0.0 (1 samples)
Reply time [ms]: response 0.2 transfer 0.0
Reply size [B]: header 157.0 content 157.0 footer 0.0 (total 314.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=0 5xx=10000

CPU time [s]: user 2.62 system 7.27 (user 26.2% system 72.7% total 98.9%)
Net I/O: 373.1 KB/s (3.1*10^6 bps)

Errors: total 0 client-time 0 socket-time 0 connrefused 0 connreset 0
Errors: fd-unavail 0 address-unavail 0 flush-full 0 other 0
root@tik:~#

```

Gambar 7. workload httpperf dengan 10000 connection

3.3. Hasil Uji Troughput dan Response Time Server Elearning

Berikut ini hasil uji **load balancing** pada elearning Unisbank dengan workload 1000 connection, 5000 connection dan 10000 connection.

Tabel. 1. hasil uji Troughput dan Response Time Server Elearning

Jumlah Koneksi	Troughput (KBPS)	Response Time (MS)	Request Lost
1000	373.1 KB/s	0.2	0
5000	372.7 KB/s	0.2	0
10000	373.1 KB/s	0.2	0

4. Conclusion

- Throughput dengan pengujian menggunakan algoritma load balancing round robin bernilai 373.1 KB/s
- Respon time pada pengujian menggunakan algoritma load balancing round bernilai 0.2 ms.
- Penggunaan metode load balancing **metode Round Robin** mampu membagi permintaan user di beberapa web server elearning sehingga kebutuhan pengguna elearning akan lebih optimal (tidak overload atau crash).

References

- [1] Agung Nugroho dkk. 2017. Analisis Perbandingan Performa Algoritma Round Robin dan Least Connection untuk **Load Balancing** pada Software Defined Network. **Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer**. Vol. 1, No. 12, Desember 2017, hlm. 1568-1577. e-ISSN: 2548-964X
- [2] Anwar, M. K., & Nurhaida, I. (2019). **Implementasi Load Balancing Menggunakan Metode Equal Cost Multi Path (ECMP) Pada Interkoneksi Jaringan**. InComTech: Jurnal Telekomunikasi dan Komputer, 9(1), 39-48.
- [3] Ari Wibowo, Dkk. 2018. **Load Balancing Pada Cloud Computing Menggunakan Metode Least Connection**. e-Proceeding of Engineering : Vol.5, No.3 Desember 2018
- [4] Hardiyan K. Ramadhan, Sukma Wardhana. 2021. **Optimalisasi Jaringan Komputer Menggunakan Algoritma Load Balancing di Virtual Server Citrix ADC**. Seminar Nasional Pengaplikasian Telematika (SINAPTIKA) Vol 1 No 1 2021 ISSN: 2086-8251
- [5] Putra, A. S., & Radita, F. R. (2020). **Paradigma Belajar Mengaji Secara Online Pada Masa Pandemic Coronavirus Disease 2019 (Covid-19)**. MATAAZIR: Jurnal Administrasi dan Manajemen Pendidikan, 1(1), 49-61.
- [6] Solehudin, A., Mayasari, R., Garno, G., & Irawan, A. S. Y. (2020). **Perbandingan Algoritma Round Robin dan Algoritma Least Connection pada Haproxy untuk Load Balancing Web Server**. SYSTEMATICS, 2(1), 21-27.