



Peran Load Balancing Dalam Meningkatkan Kinerja Web Server di Lingkungan Cloud

Sampurna Dadi Riskiono¹, Dedi Darwis²

¹Teknik Elektro, Universitas Teknokrat Indonesia, Indonesia

²Sistem Informasi Akuntansi, Universitas Teknokrat Indonesia, Indonesia

E-mail koresponden: sampurna.go@teknokrat.ac.id

Abstrak

Semakin pesatnya pertumbuhan internet saat ini, berdampak pada meningkatnya akses pengguna yang terhubung di dalamnya. Hal tersebut berpengaruh pada kebutuhan terhadap mesin penyedia layanan, seperti halnya server web server. Hadirnya teknologi cloud saat ini, sangat membantu para pengelola web server dalam melakukan manajerial Web server khususnya terhadap mesin server yang digunakan. Namun dalam upaya untuk meningkatkan kinerja sebuah web server yang mengalami peningkatan akses oleh penggunanya, maka diperlukan evaluasi terhadap penerapan metode load balancing dalam mendukung peningkatan kinerja di lingkungan cloud. Sehingga beban koneksi yang masuk untuk meminta layanan web server tersebut dapat dilayani oleh beberapa server yang menyediakan konten web server yang sama. Sehingga dengan hal tersebut, maka kinerja dari sebuah web server dapat terus dipertahankan bahkan mungkin ditingkatkan. Oleh karena itu diperlukan penelitian untuk merancang bagaimana membangun model arsitektur server yang mampu mengatasi peningkatan beban server yang disebabkan meningkatnya permintaan layanan yang diterima. Hal ini bertujuan untuk meningkatkan layanan yang dapat diberikan oleh server penyedia kepada penggunanya. Salah satu yang dapat diupayakan untuk mengatasi permasalahan tersebut adalah penggunaan banyak server. Sehingga dengan metode load balancing maka distribusi beban dapat seimbang dan merata ke masing-masing server. Dengan melihat hasil pengujian yang telah didapat, maka perbandingan nilai rata-rata menunjukkan sistem server yang telah didukung dengan penerapan load balancing memiliki nilai response time 61.19 ms, nilai ini lebih kecil 1.62 ms jika dibandingkan tanpa implementasi dari load balancing yang memiliki waktu respon 62.81 ms. Kondisi tersebut menunjukkan waktu respon yang diberikan web server lebih cepat dengan adanya load balancing yang memungkinkan beban dapat didistribusikan merata pada masing-masing web server yang tersedia.

Kata kunci: Cloud, Kinerja Web Server, Load Balancing.

Abstract

The more rapid growth of the internet today has an impact on increasing access to users connected to it. This affects the need for service provider machines, such as web server servers. The presence of cloud technology nowadays really helps web server managers in carrying out web server managerial especially for the server machines used. However, to improve the performance of a web server that has increased access by its users, it is necessary

to evaluate the application of load balancing methods to support increased performance in a cloud environment. So that the incoming connection load to request the Web server service can be served by several servers that provide the same web server content. So with this, the performance of a web server can be maintained or even improved. Therefore, research is needed to design how to build a server system that can handle the number of incoming service requests so that the load from the server can be overcome. This aims to improve the services that the provider's server can provide to its users. One thing that can be done to solve this problem is the use of multiple servers. So that with the load balancing method, the load distribution can be balanced and evenly distributed to each server. Finally, with the implementation of load balancing, the performance of web server services in a cloud environment can be continuously improved. From the test results that have been obtained, the comparison of the average value shows that a server system that has been supported by the application of a load balancing has a response time of 61.19 ms, this value is smaller 1.62ms than the implementation without load balancing which has a response time of 62.81 ms. This condition shows that the response time given by the web server is faster with load balancing which allows the load to be distributed evenly on each available web server.

Keywords: *Load Balancing, Web Server Performance, Cloud*

PENDAHULUAN

Semakin pesatnya pertumbuhan internet saat ini, berdampak pada meningkatnya akses pengguna yang terhubung di dalamnya. Hal tersebut menjadikan beragamnya layanan seperti aplikasi *web* dalam berbagai bentuk, yang menjadikan akses terhadap servis oleh *user* semakin melonjak. Beberapa model layanan aplikasi web yang ada diantaranya adalah *e-business*, *e-learning*, *e-news* dan sebagainya [1][2]. Perkembangan tersebut memicu munculnya teknologi baru yang bernama *cloud computing*. *Cloud computing* dapat didefinisikan sebagai kumpulan dari sumber daya komputasi, jaringan dan solusi manajemen penyimpanan serta dukungan terhadap aplikasi berbasis virtual. Dimana ketersediaannya dapat diatur sesuai kebutuhan, dengan mempertimbangkan berbagai aspek seperti aspek ekonomi. Peranan utama dari *cloud computing* adalah karena kemampuannya untuk membangun infrastruktur IT yang dinamis, terdapatnya jaminan dari kualitas layanan, dan kemudahan dalam hal konfigurasi layanan aplikasi [3].

Dengan adanya *cloud computing* maka sangat membantu dalam hal penyediaan infrastruktur. Dalam hal ini penelitian dilakukan dengan pemanfaatan *cloud computing* sebagai salah satu sumberdaya penyedia infrastruktur untuk *web server*. Sehingga dalam upaya meningkatkan kinerja sistem layanan *web server*, sistem *server* yang baik sangat dibutuhkan. Hal ini bertujuan agar dapat mengatasi permintaan ketika terjadi akses yang tinggi pada *web server* tersebut. Kondisi tersebut berpengaruh pada ketersediaan layanan, dimana model arsitektur dengan banyak *server* menjadikan sebuah opsi dalam penerapan *load balancing*. *Load balancing* yang diterapkan pada *server* merupakan salah satu teknik yang bisa diterapkan dalam meningkatkan ketersediaan serta kinerja dari sebuah *server*, yaitu dengan cara mendistribusikan akses layanan yang masuk ke sejumlah *server* yang ada secara bersamaan, hal ini menghindari terjadinya kelebihan beban yang diterima oleh *server* [4].

Penelitian ini bertujuan untuk memodelkan, merancang, menguji dan mengevaluasi metode *load balancing* yang bertujuan untuk meningkatkan *Quality of Service* (QoS) layanan dalam mengatasi permintaan yang tinggi dalam menjaga ketersediaan layanan pada *web server* di dalam lingkungan *cloud computing*. Sehingga parameter seperti *response time* dapat dimaksimalkan. Disamping itu untuk menangani kelebihan beban saat jumlah akses layanan

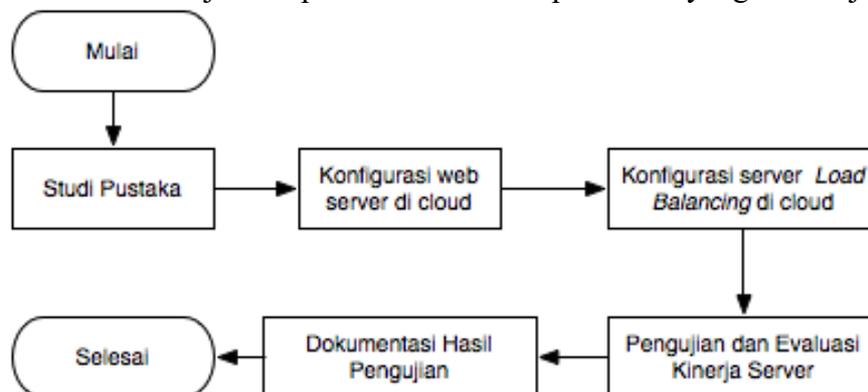
meningkat, maka penerapan sejumlah *server* bisa menjadi sebuah pilihan untuk mengatasi hal tersebut. Kumpulan dari sejumlah *server* dapat disebut juga dengan *cluster server*. Sehingga dengan implementasi *cluster server* maka ketersediaan aplikasi maupun keandalan sistem dapat ditingkatkan [5][6].

METODE PENELITIAN

Pada pengujian berikut, *load balancing* digunakan untuk mengukur peningkatan kinerja dari sebuah *web server* dengan cara mengevaluasi kondisi ketika sebelum dan sesudah menerapkan *load balancing*. Pada fase awal dilakukan kajian melalui studi pustaka. Tahap berikutnya adalah penerapan *web server* kemudian dilanjutkan penerapan dari metode *load balancing*. *Web server* akan diberikan sejumlah koneksi dalam pengujiannya. Koneksi tersebut diberikan terhadap *web server*, baik *server* tunggal maupun *server* jamak.

Tahap berikutnya dilakukan evaluasi atas penerapan metode *load balancing* yang telah diimplementasikan menggunakan indikator pengujian yang telah ditetapkan. Hasil dari evaluasi akan memperlihatkan nilai dari pengujian indikator yang telah dilaksanakan.

Pada Gambar 1 menunjukkan proses dari metode penelitian yang akan dijalankan.



Gambar 1. Metode Penelitian

Dalam melakukan penelitian ini, maka diperlukan berbagai informasi keterangan yang lengkap. Ada berbagai metode yang dilakukan penelitian ini dalam mengumpulkan data diantaranya yaitu:

1. Studi Pustaka

Pada tahap ini dilakukan pengumpulan data dengan cara mempelajari berbagai bentuk bahan tulisan seperti buku, jurnal, artikel, dan berbagai dokumen yang terkait secara langsung [7].

2. Konfigurasi *web server di cloud*

Pada tahap ini dilakukan konfigurasi *web server* di lingkungan *cloud*.

3. Konfigurasi *server load balancing di cloud*

Pada tahap ini dilakukan konfigurasi *server load balancing* di lingkungan *cloud*. Dimana nantinya ada 3 buah *server* yang akan digunakan dan dikonfigurasi.

4. Pengujian dan evaluasi kinerja *server*

Pada tahap ini dilakukan pengujian untuk mengetahui kinerja baik sebelum maupun setelah penerapan *load balancing* yang dan selanjutnya dilakukan evaluasi terhadap hasil pengujian tersebut.

5. Dokumentasi Hasil Pengujian

Pada tahap ini dilakukan dokumentasi dari hasil pengujian yang telah dilakukan, sebagai sebuah informasi yang dapat dimanfaatkan untuk kemudian hari.

Load Balancing

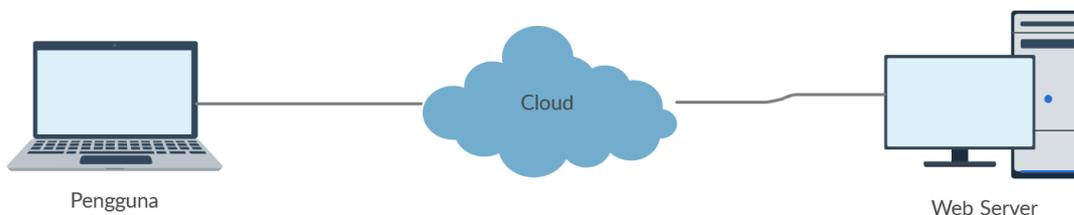
Load balancing adalah suatu teknik dalam *computer network* yang memiliki sistem kerja dengan cara membagi permintaan yang masuk ke sejumlah komputer atau *cluster* komputer untuk tujuan mencapai pemanfaatan yang optimal dari sumber daya yang ada, memperbesar *throughput*, memperkecil waktu respon, serta mengurangi terjadinya *overload*. *Cluster server* merupakan kumpulan dari sejumlah perangkat komputer yang terkoneksi satu sama lain dan saling bekerjasama dalam banyak aspek sehingga mereka dapat dimaknai sebagai satu sistem. *Cluster* komputer umumnya difungsikan agar dapat menjamin ketersediaan serta meningkatkan kinerja dari satu komputer [8]. *Load balancing* memiliki peranan yang multifungsi diantaranya dapat sebagai penyeimbang beban, pengaturan *traffic*, dan pengalihan jalur trafik. Selain sebagai penyeimbang beban, *load balancing* dapat juga difungsikan sebagai pengukur kesehatan pada sebuah *server* baik aplikasi maupun konten dalam upaya meningkatkan layanan yang tersedia serta mempermudah pengelolaan [9].

Spesifikasi Sistem Server

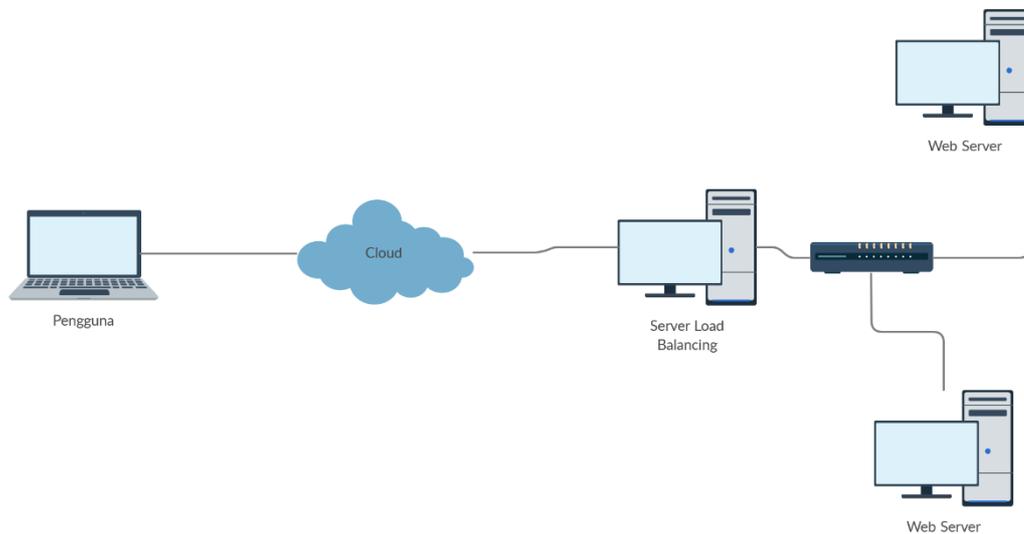
Dalam implementasi sistem *server*, penelitian ini menggunakan *Virtual Private Server* (VPS). Dimana jumlah keseluruhan VPS yang digunakan sebanyak tiga buah. Masing-masing VPS memiliki spesifikasi *server* antara lain RAM 1GB, vCPU 1 Core, Full SSD 20 GB dan satu *public IP*. Sehingga masing-masing *server* dapat diakses dan dikontrol darimana saja.

HASIL DAN PEMBAHASAN

Pengujian parameter pada *load balancing* bertujuan untuk melihat kemampuan *server* dalam melayani sejumlah permintaan yang datang dari pengguna dalam satuan waktu tertentu. Dalam pengujian tersebut parameter yang diukur adalah *response time* baik sebelum maupun setelah *load balancing* berhasil diterapkan. Untuk membangun sistem *server* yang ada, penelitian ini menggunakan VPS yang berada di *cloud* dalam membangun infrastruktur *server*-nya. Ada dua buah bentuk arsitektur *server* yang akan diujikan. Pertama adalah bentuk dari arsitektur *server* tunggal yang merupakan desain dimana *load balancing* belum diterapkan seperti yang diperlihatkan pada Gambar 2. Arsitektur tersebut dalam melayani permintaan yang datang dari pengguna, hanya menggunakan satu *server* untuk bertugas. Kemudian yang kedua adalah arsitektur dengan banyak *server* seperti yang diperlihatkan pada Gambar 3, dimana model ini telah menerapkan *server load balancing* yang akan diimplementasikan dalam penelitian. Arsitektur ini terdiri dari satu *server load balancing* dan dua buah *web server*. Sehingga permintaan dari pengguna tidak lagi dilayani oleh satu *server*, tetapi dapat dilayani oleh dua *server* atau lebih.



Gambar 2. Arsitektur Server Tunggal



Gambar 3. Arsitektur server jamak dengan load balancing

Pada Gambar 3 menunjukkan bahwa *server load balancing*, *web server 1*, *web server 2* terletak pada jaringan *cloud*. Sehingga pengguna dapat melakukan akses melalui jaringan publik dalam mendapatkan layanan dari *web server*. Untuk IP address dari antarmuka jaringan yang terpasang dapat dilihat pada Tabel 1.

Tabel 1. Penamaan Server dan Alamat IP

Penamaan Server	Alamat IP
Server Load Balancing	103.41.207.159
web server 1	103.41.206.159
web server 2	103.41.207.160
Pengguna	192.168.109.128

Pada informasi yang terdapat pada Tabel 1, memperlihatkan alamat dari IP 103.41.207.159 merupakan alamat IP yang dimiliki *server load balancing*. Dimana *server* ini memiliki fungsi dalam mendistribusikan permintaan layanan dari pengguna ke *web server*. Saat pengguna mengakses layanan yang terdapat pada *web server*, maka setiap permintaan yang masuk akan selalu diarahkan untuk melewati *server load balancing* terlebih dahulu. Selanjutnya permintaan akan diarahkan ke *web server* yang berada di belakangnya. Jadi dalam skenario ini, pengguna tidak akan dapat mengetahui posisi *web server* mana yang bertugas melayani permintaan tersebut. Selanjutnya permintaan layanan tersebut akan dibagikan oleh *server load balancing* terhadap *web server* yang masuk dalam daftar tabel distribusi *server load balancing*.

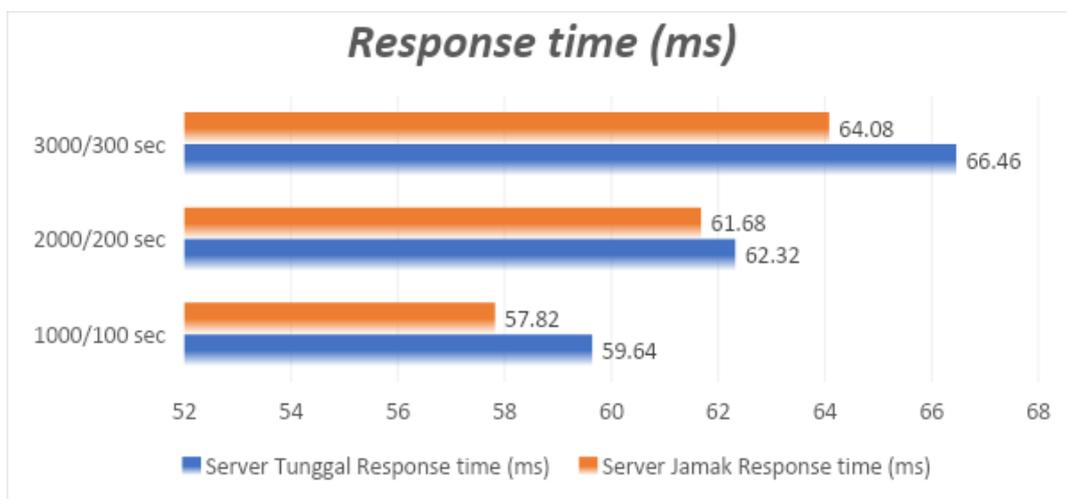
Pada penelitian ini, pengujian dilakukan dengan cara mengambil sebuah variabel nilai dari *response time*. Hal ini bertujuan untuk mengetahui kinerja dari layanan yang diberikan oleh *web server* dengan diterapkannya metode *load balancing*. Dalam proses pengujiannya, pengguna akan melakukan permintaan secara simultan melalui *tool* yang bernama *Httpperf* untuk mendapatkan nilai dari *response time*. Untuk gambaran jelasnya dapat dilihat melalui Tabel 2 yang berisikan informasi hasil pengujian dari permintaan layanan yang dilakukan oleh pengguna melalui *tool* *Httpperf* terhadap *web server*. Pelayanan permintaan dilakukan secara bertahap. Dimulai dengan membuat skenario untuk 1000 permintaan dengan 100 koneksi/*second*, 2000 permintaan dengan 200 koneksi/*second*, dan 3000 permintaan dengan

300 koneksi/*second* yang dilakukan secara bersamaan dalam satu waktu terhadap *web server*. Pada pengujian yang telah dilakukan menghasilkan nilai yang berbeda untuk setiap skenarionya. Hal ini disebabkan oleh rentang waktu pengujian yang lebih panjang untuk setiap skenario pengujian. Sehingga menyebabkan antrian yang berbeda untuk setiap permintaan yang harus diproses. Kondisi ini berpengaruh pada nilai *response time* yang terus meningkat seiring peningkatan *request* layanan. Berdasarkan dari hasil nilai *response time* yang telah diperoleh, maka penerapan kinerja dari perbandingan kedua arsitektur dapat diketahui.

Tabel 2. Hasil pengujian terhadap *server tunggal* dan *server jamak*

Koneksi /sec	Server Tunggal Response time (ms)					Hasil Rata-rata Pengujian Response time (ms)	Server Jamak Response time (ms)					Hasil Rata-rata Pengujian Response time (ms)
	U1	U2	U3	U4	U5		U1	U2	U3	U4	U5	
1000/100	58.3	58.4	61.1	59.7	60.7	59.64	62.3	60.3	59.2	53.2	54.1	57.82
2000/200	60.3	62.4	65.3	61.1	62.5	62.32	64.2	61.1	61.7	60.7	60.7	61.68
3000/300	68.4	66.1	66.1	64.4	67.3	66.46	61.3	63.2	61.5	68.5	65.9	64.08
Hasil Rata-rata Pengujian Response time (ms)						62.81	Hasil Rata-rata Pengujian Response time (ms)					61.19

Pada Gambar 4 menunjukkan perbandingan dalam bentuk grafik antara model satu *server* dan banyak *server/server jamak* dilihat dari hasil pengujian berdasarkan nilai *response time* yang didapat.

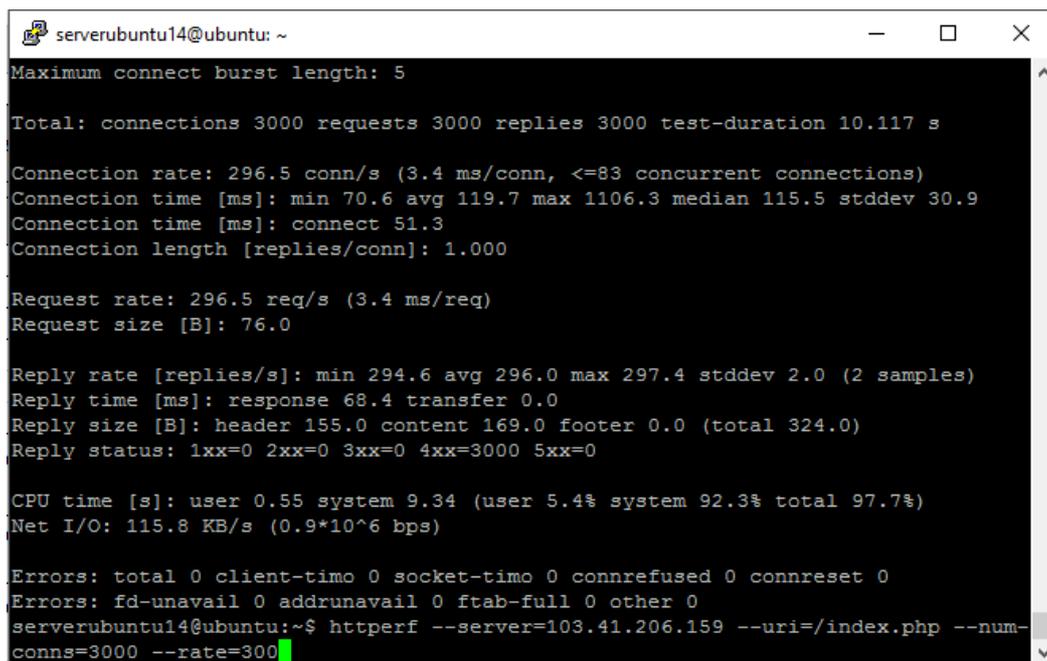


Gambar 4. Bentuk Grafik dari Nilai Response Time (ms)

Dari pengujian kedua model arsitektur baik *server tunggal* dan *server jamak* maka dapat dilihat pada Tabel 2. Dari informasi yang dihasilkan, implementasi *load balancing* memiliki

nilai *response time* lebih kecil dibandingkan sebelum diimplementasikannya *load balancing* dari rentang waktu pengujian yang dilakukan. Pada uji koneksi dengan nilai permintaan sebanyak 1000/100 *second* didapatkan nilai *response time* dari jamak yaitu 64.08 ms, nilai ini lebih kecil bila dibandingkan dengan arsitektur tanpa *load balancing* yang memiliki nilai *response time* sebesar 66.46 ms. Kondisi tersebut memperlihatkan bahwa kecepatan *response time* terhadap layanan yang didukung oleh implementasi *load balancing* lebih baik dibandingkan ketika menggunakan *server* tunggal.

Dilihat dari segi kinerjanya penerapan *load balancing* layak untuk diterapkan berdasarkan nilai QoS yang baik dari suatu layanan yang diberikan. Hal serupa juga terjadi pada uji koneksi selanjutnya seperti pada Gambar 5, dimana pada uji koneksi 2000/200 *second*, dan 3000/300 *second* penerapan *server* jamak yang menggunakan metode *load balancing* memberikan hasil yang lebih baik dibandingkan dengan implementasi *server* tunggal.



```
serverubuntu14@ubuntu: ~
Maximum connect burst length: 5

Total: connections 3000 requests 3000 replies 3000 test-duration 10.117 s

Connection rate: 296.5 conn/s (3.4 ms/conn, <=83 concurrent connections)
Connection time [ms]: min 70.6 avg 119.7 max 1106.3 median 115.5 stddev 30.9
Connection time [ms]: connect 51.3
Connection length [replies/conn]: 1.000

Request rate: 296.5 req/s (3.4 ms/req)
Request size [B]: 76.0

Reply rate [replies/s]: min 294.6 avg 296.0 max 297.4 stddev 2.0 (2 samples)
Reply time [ms]: response 68.4 transfer 0.0
Reply size [B]: header 155.0 content 169.0 footer 0.0 (total 324.0)
Reply status: 1xx=0 2xx=0 3xx=0 4xx=3000 5xx=0

CPU time [s]: user 0.55 system 9.34 (user 5.4% system 92.3% total 97.7%)
Net I/O: 115.8 KB/s (0.9*10^6 bps)

Errors: total 0 client-timo 0 socket-timo 0 connrefused 0 connreset 0
Errors: fd-unavail 0 addrunavail 0 ftab-full 0 other 0
serverubuntu14@ubuntu:~$ httpperf --server=103.41.206.159 --uri=/index.php --num-
conns=3000 --rate=300
```

Gambar 5. Hasil proses pengujian koneksi terhadap *web server* menggunakan *Httpperf*

KESIMPULAN

Penelitian ini melakukan pengujian implementasi penerapan *load balancing* dan tanpa penerapan *load balancing*. Hasil pengujian penerapan *load balancing* didapatkan waktu *response time* 1,62 ms lebih kecil dibandingkan tanpa *load balancing*. Kondisi ini menyatakan bahwa pembagian beban dapat didistribusikan secara merata pada masing-masing *web server* yang berpengaruh terhadap waktu respon yang diberikan *server*. *Server load balancing* dalam penelitian ini bersifat tunggal sehingga peningkatan jumlah *server load balancing* dan penerapan model *fault tolerance* sangat diperlukan demi ketersediaan sistem serta layanan yang lebih baik.

UCAPAN TERIMA KASIH

Terima kasih kami ucapkan kepada Universitas Teknokrat Indonesia melalui LPPM yang telah membiayai dan mendukung kegiatan penelitian ini.

DAFTAR PUSTAKA

- [1] S. D. Riskiono and D. Pasha, “Analisis Metode Load Balancing Dalam Meningkatkan Kinerja Website E-Learning,” *J. Teknoinfo*, vol. 14, no. 1, p. 22, 2020, doi: 10.33365/jti.v14i1.466.
- [2] N. Angsar, “Pengujian Distribusi Beban Web dengan Algoritma Least Connection dan Weighted Least Connection,” *Jnteti*, vol. 3, no. 1, pp. 24–28, 2014.
- [3] Y. Afrianto and A. H. Hendrawan, “Implementasi Data Center Untuk Penempatan Host Server Berbasis Private Cloud Computing,” *Krea-TIF*, vol. 7, no. 1, p. 50, 2019, doi: 10.32832/kreatif.v7i1.2031.
- [4] D. Lukitasari, F. Oklilas, F. I. Komputer, and U. Sriwijaya, “Analisis Perbandingan Load Balancing Web Server Tunggal Dengan Web server Cluster Menggunakan Linux Virtual Server,” vol. 5, no. 2, pp. 31–34, 2010.
- [5] S. D. Riskiono, S. Sulisty, and T. B. Adji, “EVALUASI METODE LOAD BALANCING MENGGUNAKAN HAPROXY SERVER CHAT SOCIAL NETWORK,” pp. 635–639, 2016.
- [6] S. D. Riskiono, S. Sulisty, and T. B. Adji, “Kinerja Metode Load Balancing dan Fault Tolerance Pada Server Aplikasi Chat,” *Pros. Semin. Nas. ReTII*, 2017.
- [7] A. Amarudin and S. D. Riskiono, “Analisis Dan Desain Jalur Transmisi Jaringan Alternatif Menggunakan Virtual Private Network (Vpn),” *J. Teknoinfo*, vol. 13, no. 2, p. 100, 2019, doi: 10.33365/jti.v13i2.309.
- [8] U. Haluoleo, K. Bumi, and T. Anduonohu, “Peningkatan Kinerja Siakad Menggunakan Metode Load Balancing dan Fault Tolerance Di Jaringan Kampus Universitas Halu Oleo,” vol. 10, no. 1, pp. 11–22, 2016.
- [9] Shahzad Malik, “Dynamic Load Balancing in a Network of Workstations,” no. 219762, 2000.