IMPLEMENTASI REVERSE PROXY SERVER SEBAGAI LOAD BALANCING DAN HTTPS PROXY MENGGUNAKAN NGINX PADA GOOGLE CLOUD PLATFORM

TUGAS AKHIR

Iman Yuliarto
41516120094

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2022

IMPLEMENTASI REVERSE PROXY SERVER SEBAGAI LOAD BALANCING DAN HTTPS PROXY MENGGUNAKAN NGINX PADA GOOGLE CLOUD PLATFORM

Tugas Akhir

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

Oleh:
Iman Yuliarto
41516120094

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MERCU BUANA
JAKARTA
2022

LEMBAR PERNYATAAN ORISINALITAS

LEMBAR PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

NIM	: 41516120094
Nama	: Iman Yuliarto
Judul Tugas Akhir	: Implementasi Reverse Proxy Server Sebagai Load Balancing Dan HTTPS Proxy Menggunakan Nginx Pada Google Cloud Platform

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan plagiat. Apabila ternyata ditemukan didalam laporan Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 06 Juli 2022

IMAN YULIARTO

ii

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa	:	Iman Yuliarto
NIM	:	41516120094
Judul Tugas Akhir	:	Implementasi Reverse Proxy Server Sebagai Load Balancing Dan HTTPS Proxy Menggunakan Nginx Pada Google Cloud Platform

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 6 Juli 2022

IMAN YULIARTO

iii

SURAT PERNYATAAN LUARAN TUGAS AKHIR

SURAT PERNYATAAN LUARAN TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa	:	Iman Yuliarto
NIM	:	41516120094
Judul Tugas Akhir	:	Implementasi Reverse Proxy Server Sebagai Load Balancing Dan HTTPS Proxy Menggunakan Nginx Pada Google Cloud Platform

Menyatakan bahwa :

1. Luaran Tugas Akhir saya adalah sebagai berikut :

No	Luaran	Jenis		Status	
1	Publikasi Ilmiah	Jurnal Nasional Tidak Terakreditasi		Diajukan	1
		Jurnal Nasional Terakreditasi	1		
		Jurnal International Tidak Bereputasi		Diterima	
		Jurnal International Bereputasi			
	Disubmit/dipublikasikan di :	Nama Jurnal	: Jurnal Informatika		
		ISSN	: 2528-2247		
		Link Jurnal	: https://ejournal.bsi.ac.id/ejurnal/index.php/ji/index		
		Link File Jurnal Jika Sudah di Publish	:		

2. Bersedia untuk menyelesaikan seluruh proses publikasi artikel mulai dari submit, revisi artikel sampai dengan dinyatakan dapat diterbitkan pada jurnal yang dituju.
3. Diminta untuk melampirkan scan KTP dan Surat Pernyataan (Lihat Lampiran Dokumen HKI), untuk kepentingan pendaftaran HKI apabila diperlukan

Demikian pernyataan ini saya buat dengan sebenarnya.

Mengetahui
Dosen Pembimbing TA

Ir. Emil R. Kaburuan, Ph.D., IPM

Jakarta, 22 Juli 2022

Iman Yuliarto

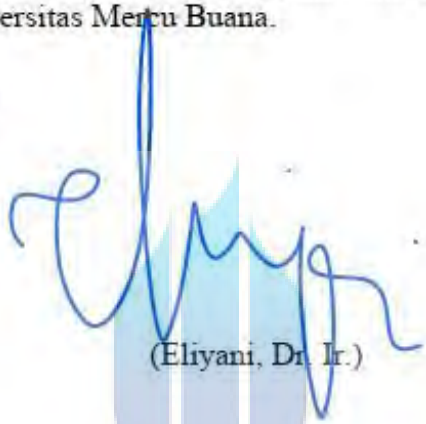
iv

LEMBAR PERSETUJUAN PENGUJI

Nama Mahasiswa	:	Iman Yuliarto
NIM	:	41516120094
Judul Tugas Akhir	:	Implementasi Reverse Proxy Server Sebagai Load Balancing Dan HTTPS Proxy Menggunakan Nginx Pada Google Cloud Platform

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 28 Juli 2022



(Eliyani, Dr. Ir.)

LEMBAR PERSETUJUAN PENGUJI

NIM	:	41516120094
Nama	:	Iman Yuliarto
Judul Tugas Akhir	:	Implementasi Reverse Proxy Server Sebagai Load Balancing dan Https Proxy Menggunakan Nginx Pada Google Cloud Platform

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 28 Juli 2022

(Achmad Kodar, Drs. MT)

LEMBAR PERSETUJUAN PENGUJI

Nama Mahasiswa	:	Iman Yuliarto
NIM	:	41516120094
Judul Tugas Akhir	:	Implementasi Reverse Proxy Server Sebagai Load Balancing Dan HTTPS Proxy Menggunakan Nginx Pada Google Cloud Platform

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 28 Juli 2022

(Sabar Rudiarto, M.Kom)

LEMBAR PENGESAHAN

NIM	:	41516120094
Nama	:	Iman Yuliarto
Judul Tugas Akhir	:	Implementasi Reverse Proxy Server Sebagai Load Balancing Dan HTTPS Proxy Menggunakan Nginx Pada Google Cloud Platform

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 28 Juli 2022

Menyetujui,

Ir. Emil R. Kaburuan, Ph.D., IPM
Dosen Pembimbing

Mengetahui,

(Wawan Gunawan, S.Kom, MT)	(Ir. Emil R. Kaburuan, Ph.D., IPM.)
Koord. Tugas Akhir Teknik Informatika	Ka. Prodi Teknik Informatika

KATA PENGANTAR

Puji syukur kita panjatkan kepada Tuhan Yang Maha Esa, berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan penyusunan tugas akhir dengan judul "Implementasi *Reverse proxy* Server Sebagai *Load balancing* Dan HTTPS Proxy Menggunakan Nginx Pada Google Cloud Platform". Penulisan tugas akhir ini diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Mercubuana.

Penulis menyadari bahwa tanpa bimbingan, arahan serta dukungan dari berbagai pihak, penulisan tugas akhir ini tidak akan berjalan lancar. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua penulis dan saudara penulis, yang selalu senantiasa memberikan doa, motivasi, dan dukungan sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Bapak Emil R. Kaburuan, Ph.D., selaku Kaprodi Teknik Informatika Fakultas Ilmu Komputer juga selaku dosen pembimbing tugas akhir dan akademik.
3. Ibu Dr. Eliyani selaku dosen penguji yang telah memberikan arahan revisi agar penulisan tugas akhir ini menjadi lebih baik.
4. Bapak Drs. Achmad Kodar, MT selaku dosen penguji tugas akhir yang senantiasa arahan dalam mengerjakan tugas akhir ini.
5. Bapak Sabar Rudiarto, M.Kom selaku dosen penguji yang telah memberikan arahan revisi agar penulisan tugas akhir ini menjadi lebih baik.
6. Teman rekan kerja tim HIINCD di PT Multipolar Technology yang selalu mensupport dan memotivasi selama bekerja maupun diluar pekerjaan.
7. Teman seperjuangan hidup digrup Detektif Rasa yang selalu mengingatkan untuk mengerjakan tugas akhir dan memberi asupan makanan.
8. Teman-teman penulis lain yang tidak dapat disebutkan satu persatu disini atas motivasi dan bantuan yang telah diberikan.

Akhir kata, penulis berharap semoga seluruh bantuan dan kebaikan mereka semua mendapat balasan dari Tuhan Yang Maha Esa. Penulis menyadari bahwa penulisan tugas akhir ini masih jauh dari kesempurnaan, oleh karena itu penulis mengharap saran dan kritik untuk penelitian selanjutnya dalam pengembangan tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi pembaca terutama teman-teman mahasiswa Fakultas Ilmu Komputer Universitas Mercubuana.

Jakarta, 6 Juli 2022

Iman Yuliarto

DAFTAR ISI

HALAMAN SAMPUL ..... **i**
HALAMAN JUDUL ..... **i**
LEMBAR PERNYATAAN ORISINALITAS ..... **ii**
SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR ... **iii**
SURAT PERNYATAAN LUARAN TUGAS AKHIR ..... **iv**
LEMBAR PERSETUJUAN PENGUJI ..... **v**
LEMBAR PENGESAHAN ..... **viii**
ABSTRAK ..... **ix**
ABSTRACT ..... **x**
KATA PENGANTAR ..... **xi**
DAFTAR ISI ..... **xiii**
NASKAH JURNAL **1**
KERTAS KERJA **10**
BAB 1. LITERATUR REVIEW **11**
BAB 2. ANALISIS DAN PERANCANGAN **16**
BAB 3. SOURCE CODE **25**
BAB 4. DATASET **27**
BAB 5. TAHAPAN EKSPERIMEN **28**
BAB 6. HASIL SEMUA EKSPERIMEN **31**
DAFTAR PUSTAKA **37**
LAMPIRAN DOKUMEN HAKI **39**
LAMPIRAN KORESPONDENSI **42**

NASKAH JURNAL

Implementasi Reverser Proxy Server Sebagai Load Balancing Dan HTTPS Proxy Menggunakan Nginx Pada Google Cloud Platform

Iman Yuliarto[1], Emil Kaburuan[2]

[1,2] Fakultas Ilmu Komputer, Universitas Mercu Buana
Jakarta, Indonesia

e-mail: [1]imansilva97@gmail.com, [2]emil.kaburuan@mercubuana.ac.id

Informasi Artikel Diterima: 00-00-2021 Direvisi: 00-00-2021 Disetujui: 00-00-2021

Abstrak

Perkembangan internet yang sudah semakin maju mempengaruhi pengunjung pada website. Kualitas layanan suatu website ditentukan oleh waktu respons server, semakin cepat responsnya maka akan lebih baik. Ketika banyak *request* dalam satu waktu, maka website akan mengalami pelambatan waktu respons akan membuat server overload dan terjadinya *downtime*. Solusi pada masalah ini adalah menggunakan sistem *load balancing* dengan membagi *traffic* pada website ke beberapa server dengan 3 metode algoritma *(Round Robin, Least connection)*. *Secure Socket Layer* merupakan tambahan lapisan keamanan pada situs web, namun adanya *SSL* akan mebebani server dengan melakukan *decrypt* proses pada setiap koneksi. Solusi agar tidak membebani setiap server, maka digunakan fitur *HTTPS Proxy* pada server *load balancing*. Untuk memenuhi kebutuhan ini *software* Nginx mempunyai fitur yang sesuai. Tujuan dari penelitian ini adalah melakukan analisis pada server yang belum dan sudah menggunakan *load balancer* dengan menggunakan beberapa metode yang tersedia untuk memperoleh performa terbaik dan beban pada setiap server yang lebih ringan. Karena keterbatasan resource, pengujian dijalankan pada server *Google Cloud Platform* dan menggunakan sertifikat gratis dari *Let's Encrypt*. Pengujian menggunakan *software apache bechmark tool* dengan setiap pengujian mengalami peningkatan yaitu 5000, 10000, 15000 *request* dengan *concurrent connection* pada setiap pengujian adalah 100 dan 500. Hasilnya algoritma *round robin* memiliki Performa lebih baik dibanding metode yang lain. Dan metode *HTTPS Proxy* yang membuat kinerja CPU dan memori pada setiap *backend* server lebih ringan adalah *Termination.*

Kata Kunci: *Load Balancing,* Nginx*, Reverse Proxy*.

Abstract

The development of the internet has been increasingly affecting visitors to the website. The quality of service of a website is determined by the server performance, the faster is better. When there are many requests at one time, the website will experience a slowdown in response time will make server overload and downtime occur. The solution to this problem is use a load balancing system by dividing traffic to several servers with Round Robin, Least Connection algorithm. The Secure Socket Layer is an additional layer of security on the website, but the existence of SSL will burden the server by decrypting the process on each connection. The solution is not to overload each server, the HTTPS Proxy feature is used on the load balancing server. To meet these needs Nginx software has the appropriate features. The purpose of this study is to perform analysis on servers that have not and are already using a load balancer using several available methods to obtain the best performance time and load on each lighter server. Due to resource limitations, tests run on Google Cloud Platform servers and use a free certificate from Let's Encrypt. Testing using apache benchmark tool with each test has increased, namely 5000, 10000, 15000 requests with concurrent connection in each tests is 100, and 500. The result is an algorithm round robin has a better performance than other methods. And the HTTPS proxy method that has a lighter amount of performance load on the server is Termination.

***Keywords**: Load Balancing,* Nginx*, Reverse Proxy.*

1. Pendahuluan

Perkembangan penggunaan internet saat ini semakin luas mengarah kesegala bidang dalam kegiatan sehari - hari. Dengan revolusi industri yang terjadi di dunia, pemeritnah menetapkan 10 prioritas nasional salah satunya yaitu dengan menerapkan insentif investasi teknologi dan membangun infrastruktur digital nasional (Dwi Anggraini & W. Finaka, 2018). Karena teknologi salah satu peran penting dalam hal ini, maka sebuah layanan *client* - server juga menjadi hal yang umum digunakan. Dengan menggunakan smartphone dan membuka aplikasi ataupun website maka terjadi proses *request* dari *client* kepada server yang dituju.

Proses ini biasanya dilakukan menggunakan protokol *HTTP* yang terjadi antara *client* dan server. Web server merupakan perangkat lunak yang berfungsi melayani permintaan *request HTTP* dari *client* dengan aplikasi seperti web browser atau aplikasi lainnya (Bustomi, Syahiruddin, Afandi, Fahmi, & Holle, 2019). *Hypertext Transfer Protocol* merupakan salah satu protokol pada layer aplikasi yang didesain untuk komunikasi antara web server dengan web browser dengan *client* mengirimkan *HTTP request* dan web server menjawab dengan *HTTP* Responsse (Friyanto, 2019). Dengan menggunakan hypertext, pengguna dapat berpindah dari satu dokumen ke dokumen yang lainnya dengan sangat mudah, hanya dengan melakan klik pada teks spesial yang disebut link dan hypertext pada situs web juga dapat menampilkan gambar, video dan lainnya dengan menggunakan *hypermedia* (Dody Firmansyah, Bachtiar, Sfenrianto, & Robert Kaburuan, 2019).

Namun karena *HTTP* tidak memiliki fitur keamanan, maka rentan adanya terjadi serangan man-in-the-middle (MitM), maka diciptakan lah *HTTPS* yang menggunakan Secure Socket Layer (*SSL*) atau Transport Layer Security (TLS) sebagai lapisan tambahan dalam protokol *HTTP* untuk melakukan enkripsi dan dekripsi antara *HTTP request*s dan *HTTP* Responsses (Wesley & Ferguson, 2021). Proses dekripsi dan enkripsi dapat mempengaruhi kinerja yang akan menambah waktu load time dari server (Radivilova, Kirichenko, Ageyev, Tawalbeh, & Bulakh, 2018).

Dengan bertambahnya pemakaian internet, maka akan bertambah juga dengan beban trafik pada web server yang akan mengakibatkan kenaikan pada waktu Respons server dan bisa mengakibatkan server menjadi down (Ibrahim et al., 2021). Oleh karena itu solusi yang bisa mengatasi permasalahan tersebut yakni dengan menggunakan sistem *load balancing*. Sistem *load balancing* merupakan metode jaringan komputer untuk mendistribusikan beban kerja pada beberapa sumber daya (Junandia, 2020). Dengan sistem *load balancing* pada web server, maka trafik yang mengarah pada web server akan didistribusikan kepada kluster server (Riskiono & Pasha, 2020).

Nginx adalah salah satu software *HTTP* server, *reverse proxy* server, mail *proxy* server, dan generic TCP/UDP *proxy* server berbasis open-source yang saat ini dimiliki oleh Nginx.inc dan telah diakusisi oleh F5.inc (Robertson, 2019). Nginx memiliki kelebihan yaitu penggunaan memori yang kecil, dapat mengatasi jumlah permintaan pada satu waktu yang tinggi, mudah dilakukan pengembangan dan juga kaya dengan modul pihak ketiga (Wen, Li, & Yang, 2018). Metode *load balancing* yang dapat digunakan pada Nginx open-source ini yaitu *Round robin*, *Least connection*, and IP-Hash(Putra, 2018).

Termination terjadi ketika Nginx sebagai *load balancer* server melakukan dekripsi terhadap data yang sudah dienkripsi oleh *SSL* dan mengirimkan data tersebut kepada web server. Hal tersebut dapat meningkatkan performa pada web server karena tidak perlu lagi melakukan dekripsi-enkripsi data (Dr. Pierre, 2020). Sertifikat yang digunakan pada penelitian ini yaitu Let's Encrypt. Sertifikat ini gratis untuk digunakan dan trusted oleh semua web browser yang disediakan oleh Internet Security Research Group (ISRG)(Tedyyana, 2020).

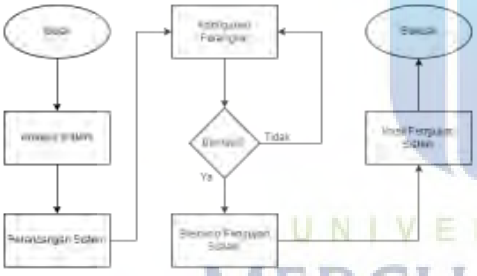
Sedangkan sistem virtual yang digunakan adalah server yang diinstall pada Google Cloud Platform (GCP) yang menyediakan layanan Infrastructure as a Service (IaaS). IaaS memiliki kelebihan dibanding sistem on-premises yaitu peneliti tidak perlu melakukan installasi network, storage, virtualization dan manajemen server fisik (Bansal, 2020). Peneliti hanya melakukan installasi Nginx sebagai *reverse proxy* atau sebagai web server. Pada GCP juga terdapat sistem monitoring untuk memantau pemakaian dari CPU, Memory,

dan Throughput pada jaringan virtualisasi server tersebut.

Pengujian pada penelitian ini menggunakan *apache benchmark tools* untuk mengetahui kinerja web server. Alat ini dapat menghitung performa dari suatu website dengan melakukan pengujian seperti mengimkan *request* yang banyak dalam waktu dan user yang bersamaan (Satwika & Semadi, 2020). Lalu hasil dari pengujian akan mempunyai data *response time*, *request* per *second*, *Transfer Rate*, dan utilisasi cpu dan memori setiap pengujian yang bertujuan untuk melihat beban kerja setiap server(Bella, Data, & Yahya, 2018).

2. Metode Penelitian

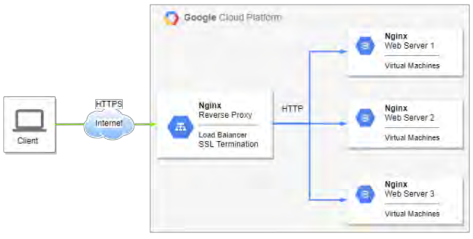
Dalam penelitian ini peneliti akan menguraikan langkah langkah yang akan dilaksanakan pada penelitian ini. terbagi menjadi beberapa tahapan penelitian yaitu: Analisis Sistem, Perancangan Sistem, Konfigurasi Perangkat, Skenario Pengujian dan Hasil Pengujian (Bustomi et al., 2019).



Gambar 1 Metodologi Penelitian

2.1. Analisis Sistem

Dalam penelitian ini akan menggunakan Nginx sebagai *load balancer* server dan juga sebagai web server. Lalu sertifikat *SSL* akan diterminasi pada server *reverse proxy*. Berikut adalah topologi dari perancangan sistem *load balancer*.



Gambar 2 Perancangan Sistem *Load Balancer*

Pada Gambar 2 merupakan diagram perancangan sistem. Berikut penjelasan dari gambar diatas:

1. Web Server
   Web server memberikan layanan kepada client yang melakukan *request* melalui protokol *HTTP* atau *HTTPS*. Dalam penelitian ini web server menggunakan software Nginx.
2. *Reverse proxy*
   Reverse proxy sebagai jembatan antara client dengan web server yang akan menjalankan sistem *load balancing* dan *Termination*. *Reverse proxy* juga akan menggunakan aplikasi Nginx.
3. Client
   Adalah pengguna yang melakukan *request HTTP* atau *HTTPS* pada web server yang akan diarahkan menuju *reverse proxy*. Dalam penelitian ini jumlah client dapat ditentukan secara dinamis.
4. *HTTP*
   Protokol yang digunakan untuk melakukan *request* dan Responsse yang terjadi dari *reverse proxy* menuju web server.
5. *HTTPS*
   Sub protokol dari *HTTP* yang memiliki sistem keamanan seperti enkripsi dan dekripsi pada lalu lintas data antara client dan reverser *proxy*.

2.2. Perancangan Sistem

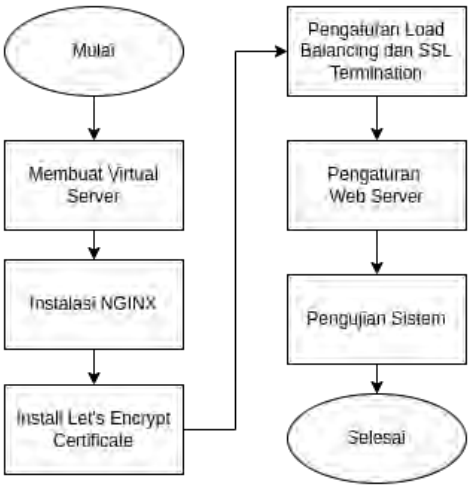
Penerapan sistem menggunakan simulasi 1 server *load balancing* dan 3 web server. Keseluruhan sistem tersebut menggunakan sistem virtual GCP yaitu *Compute Engine*. Berikut adalah tabel perbandingan dari setiap virtual server :

Tabel 1 Spesifikasi Sistem

Fungsi	CPU	Memori	Disk	Jml
Load Balancer	2	4 GB	20GB	1
Web Server	1	1.7 GB	10GB	3

Sistem operasi yang digunakan yaitu Debian 10.12 dan software yang digunakan adalah Nginx 1.20.2. Server *load balancing* menggunakan lebih banyak resource karena menjadi gateway dari setiap lalu lintas data yang menuju web server.

2.3. Konfigurasi Perangkat



Gambar 3 Alur Konfigurasi Server

Pada gambar 3 menjelaskan tentang alur konfigurasi server yang menjadi *reverse proxy* dan web server, dan tahapan ini konfigurasi server akan disesuaikan dengan perancangan.

2.3.1 Membuat Virtual Server

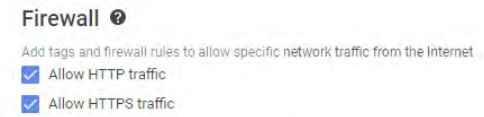
Dengan sistem virtualisasi menggunakan GCP proses pembuatan server akan dilakukan dalam satu halaman penuh yang pencakup dasar konfigurasi seperti jumlah CPU, memori, penyimpanan, IP subnet, lokasi server, dan firewall. Hal ini akan mempermudah dalam melakukan pembuatan tanpa perlu memakan waktu yang panjang sampai siap dilakukan instalasi Nginx.

Gambar 4 Memilih CPU dan Memori

Tahapan pertama yaitu memberikan hostname pada virtual machine, lalu memilih regional dimana server tersebut ditempatkan. Penulis memilih singapura untuk lokasi dan "zona b" karena alasan jarak dan juga harga sewa per jam.

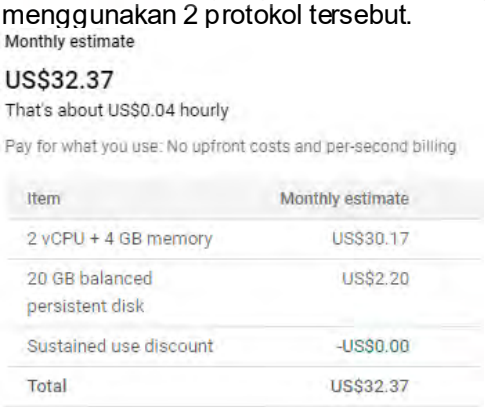
Gambar 5 Pengaturan OS

Selanjutnya memilih sistem operasi yang akan digunakan dan ukuran penyimpanan. Debian 11 yang memiliki kode *bullseye* merupakan Debian *Long Term Support* yang akan memiliki support selama 5 sejak dirilis. Dengan menggunakan debian dan Nginx, ukuran penyimpanan tidak akan membebani server.



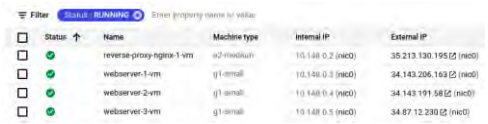
Gambar 6 Pengaturan Firewall

HTTP dan *HTTPS* perlu diizinkan untuk diakses dari internet publik karena web server yang akan dibangun menggunakan 2 protokol tersebut.



Gambar 7 Estimasi Harga

Sebelum melanjutkan untuk pembuatan server lainnya, sistem cloud google akan melakukan perhitungan untuk setiap sumber daya yang dipakai dengan hitungan USD per bulan ataupun per jam.



Gambar 8 Daftar Server

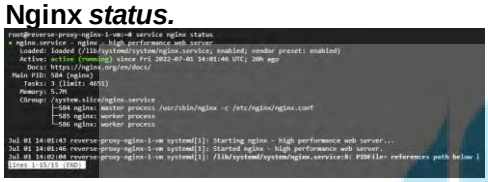
Gambar diatas adalah server yang sudah dibuat pada sistem google cloud dan siap untuk dilanjutkan dengan instalasi aplikasi Nginx.

2.3.2 Instalasi Nginx

Instalasi web server dan *reverse proxy* pada setiap server dengan menggunakan perntah:

```
$ sudo apt-get install nginx
```

dan untuk memastikan Nginx sudah berjalan dengan benar, gunakan perintah ***service* Nginx *status.***



Gambar 9 Nginx Status Active

2.3.3 Instalasi Let's Encrypt Certificate

Selanjutnya melakukan instalasi *Let's Encrypt* pada server *reverse proxy* sebagai server untuk terminasi *HTTPS*. syarat untuk melakukan instalasi sertifikat ini yaitu web server sudah terinstal pada server, memiliki domain publik dan membuat *DNS record* untuk validasi kepemilikan domain.

Certbot digunakan untuk mempermudah dalam melakukan *request* dan validasi terhadap *Let's Encrypt*. Berikut adalah perintah yang digunakan.

```
$ sudo apt-get update
$ sudo apt-get install certbot
$ sudo apt-get install python3-certbot-nginx
```

Setelah itu tambahkan *file* konfigurasi web di */etc/*Nginx*/sites-enable/rolladx.xyz.conf* dan tambahkan pada informasi domain yang digunakan.

```
server {
        listen 80;
        root /usr/share/nginx/html;
        index index.html;
        server_name rolladx.xyz
www.rolladx.xyz;
```

Lalu jalankan perintah certbot untuk memulai proses *request SSL* certificate pada le'ts encrypt dengan parameter seperti berikut ini:

```
sudo certbot certonly \
  --manual \
  -d *.rolladx.xyz \
  --server https://acme-
v02.api.letsencrypt.org/directory
```

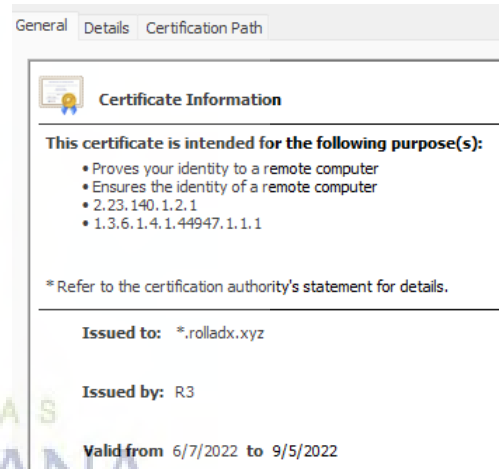
pada tahapan akhir, daftarkan dns record yang sudah di generate oleh certbot di domain yang sudah ditentukan.

Gambar 10 DNS Record

Jika berhasil, maka *file* certificate akan ada didalam folder /etc/letsencrypt/live/rolladx.xyz/

```
root@reverse-proxy-nginx-1-vm:~# ls /etc/letsencrypt/live/rolladx.xyz/
README  cert.pem  chain.pem  fullchain.pem  privkey.pem
root@reverse-proxy-nginx-1-vm:~#
```

Gambar 11 Certificate File



Gambar 12 Wildcard Certificate

2.3.4 Konfigurasi Algoritma *Load balancing & Termination*

Selanjutnya menambahkan konfigurasi load balancer pada *file* rolladx.xyz.conf sesuai dengan algoritma penelitian.

A. *Round robin*

Algoritma ini akan mendistribusikan beban koneksi secara merata keseluruh server yang ada dengan berurutan.

```
upstream lb-rolladx {
        server 10.148.0.3 max_fails=3 fail_timeout=5s;
        server 10.148.0.4 max_fails=3 fail_timeout=5s;
        server 10.148.0.5 max_fails=3 fail_timeout=5s;
}
```

Gambar 13 Round robin

B. *Least connection*

Algoritma *least connection* akan membagi koneksi kepada server dengan jumlah koneksi aktif yang paling sedikit yang bertujuan untuk mengurangi beban server.

```
upstream lb-rolladx {
        least_conn;
        server 10.148.0.3 max_fails=3 fail_timeout=5s;
        server 10.148.0.4 max_fails=3 fail_timeout=5s;
        server 10.148.0.5 max_fails=3 fail_timeout=5s;
}
```

Gambar 14 Least connection

C. *Termination*

Dengan mengaktifkan *HTTPS* pada *reverse proxy*, maka server tersebut akan melakukan enkripsi dan dekripsi pada lalu lintas data yang mengarah ke web server menggunakan *HTTP*.

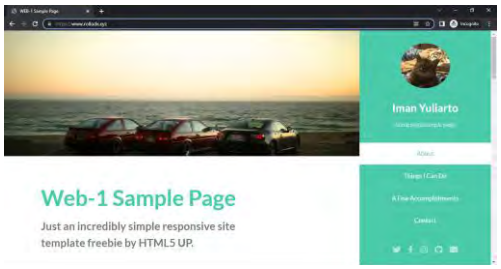
```
server {
        listen 443 ssl;
        ssl on;
        ssl_certificate /etc/letsencrypt/live/rolladx.xyz/cert.pem;
        ssl_certificate_key /etc/letsencrypt/live/rolladx.xyz/privkey.pem;
        ssl_trusted_certificate /etc/letsencrypt/live/rolladx.xyz/fullchain.pem;
        server_name rolladx.xyz www.rolladx.xyz;
        location / {
                proxy_pass http://lb-rolladx;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
        }
}
```

Gambar 15 konfigurasi SSL

Pada pengaturan *file* .conf Nginx, tambahkan perintah seperti pada gambar 15. Perintah dengan awalan *SSL* berfungsi untuk mengarahkan sertifikat yang digunakan. Lalu untuk melewatkan *request HTTPS* kepada web server backend yaitu dengan perintah *proxy*_pass. *Proxy*_set_header berfungsi untuk meneruskan informasi dari klien kepada web server backend.

2.3.5 Pengaturan Web Server

Penelitian ini menggunakan server Nginx yang digunakan pada web server backend dan menggunakan halaman statis *html* sebagai pengujian. Ketiga server tersebut dibedakan pada halaman awal yaitu dengan masing – masing hostname.



Gambar 16 Pengujian Akses Server

Gambar 16 merupakan pengujian akses ke *reverse proxy* dengan megunakan domain yang sudah diarahkan kealamat IP publik server tersebut.

2.3.6 Pengujian Sistem

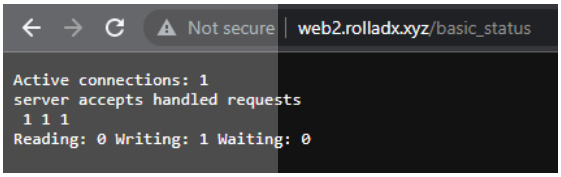
Pada tahapan ini fungsi dari keseluruhan sistem akan dilakukan pengujian. Memastikan bahwa seluruh server telah berjalan dengan baik dengan melakukan *request* koneksi terhadap server *reverse proxy* dan melihat hasil dari sample page yang mewakilkan masing – masing web server.

Seluruh algoritma *load balancing* akan diuji pada tahapan ini untuk memastikan konfigurasi sudah berjalan dengan benar.

Untuk memastikan jumlah koneksi aktif pada setiap web server, peneliti menggunakan module *ngx_http_stub_status_module* yang berfungsi untuk menampilkan status koneksi aktif, *server accepts, server handled* dan *server requests*. Berikut perintah yang ditambahkan pada setiap web server:

```
location = /basic_status {
    stub_status; }
```

dan untuk melihat informasi dari jumlah akses yang masuk yaitu dengan membuka *url "/basic_status"* pada setiap web server.



Gambar 17 Basic Status Page

Gambar 17 menunjukan bahwa server web2 memiliki jumlah koneksi yang aktif yaitu 1 dan menerima, menangani dan *request* yang masuk juga berjumlah 1.

2.4 Skenario Sistem Pengujian

Tujuan dari pengujian ini yaitu untuk mengetahui perbandingan performa dari setiap algoritma *load balancing* dan perbedaan dari beban backend server dengan menggunakan SSL pada *reverse proxy* atau pada web server.

Pengujian juga dilakukan untuk mengetahui nilai rata – rata dari setiap parameter yang diujikan. Pemberian *request* dalam satu waktu digunakan untuk memberi beban kepada server. Hasil dari setiap percobaan akan dibandingkan dan dilakukan analisis. Sehingga akan didapatkan perbandingan performa antara algoritma dan fitur *Termination* yang akan disimpulkan pada akhir pengujian.

Performa dihitung berdasarkan waktu Responsse yaitu waktu yang diperlukan sistem untuk melayani *request* dan juga throughput yaitu banyaknya *request* yang dapat dilayani setiap detiknya. *Apache*

benchmark dapat melakukan simulasi *request* dalam satu waktu dan akan mencatat hasil dari setiap pengujian.

Tools apache benchmark akan mengirimkan banyak koneksi dengan rate yang berbeda – beda. Untuk menjalankan pengujian menggunakan *tools* tersebut, berikut perintah yang digunakan:

```
$ ab -n 5000 -c 100 https://(domainwebsite)/
```

Dimana parameter *-n* menentukan jumlah *request* yang dilakukan dan *-c* menentukan jumlah user yang aktif secara bersamaan atau biasa disebut *concurrent session*.

Pengujian dilakukan dalam kondisi server backend tidak menjalankan aplikasi apapun selain nginx dan melakukan restart service nginx untuk mereset koneksi yang aktif menjadi 0.

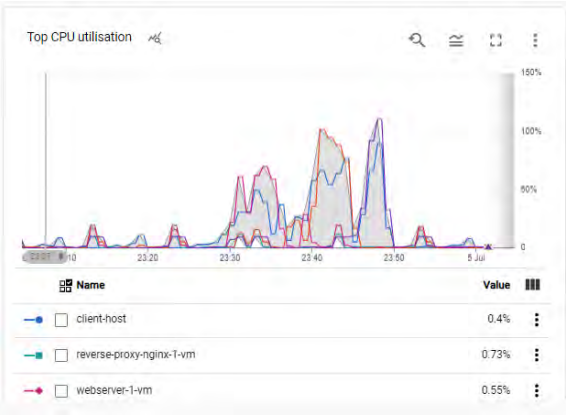
Skenario pengujian dibagi menjadi 3 sesi yaitu pengujian koneksi langsung dengan SSL pada backend server, dan pengujian load balancer dengan SSL pada *reverse proxy*.

Setiap paremeter pengujian dilakukan sebanyak 3 kali. Parameter pengujiannya yaitu sebagai berikut:

- Uji 1 = 5000 koneksi, 100 user.
- Uji 2 = 10000 koneksi, 500 user.
- Uji 3 = 15000 koneksi, 500 user.

Hasil dari pengujian tersebut akan didapatkan nilai *Time per Request* dalam satuan *milisecond (ms)*, total waktu pengujian, dan *Transfer Rate* dalam satuan *KB/s*.

Untuk mengetahui beban server dari setiap pengujian akan dilakukan pencatatan menggunakan fitur *VM Monitoring* dari Google Cloud. Data yang didapatkan berupa pemakaian *CPU* dan *Memory*.



Gambar 18 Monitoring CPU

3. Hasil dan Pembahasan

Pengujian yang dilakukan menggunakan *tools* ab melakukan *request* terhadap website yang sudah dikonfigurasi dengan menggunakan sistem reverse proxy. Maka dari itu format pengujian adalah sebagai berikut:

```
$ ab -n 5000 -c 100
https://domainnameserver/
```

Berikut disajikan tabel rata – rata dari hasil pengukuran terhadap dengan parameter pengujian yang yang dijelaskan pada bagian sebelumnya.

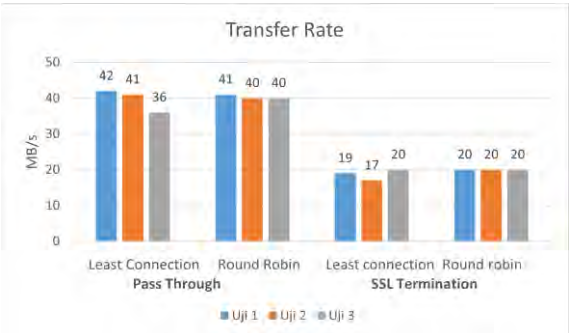
A. *Transfer Rate*

Tabel 2 Hasil Pengujian Tranfer Rate

Parameter Pengujian		Throughput / KB			
Algoritma Loadbalancing		passthrought		SSL Termination	
Uji	Pengujian	Round robin	Least Connection	Round robin	Least Connection
Pertama	1	36544	40256	19866	19920
Kedua	2	38736	37655	19841	19912
Ketiga	3	38446	38032	19834	18539

Pada tabel 2 merupakan hasil pengukuran *Transfer Rate* yang diperoleh dari pengujian menggunakan *tools Apache benchmark*.

Kedua metode pada load balancer yang memenggunakan HTTPS *proxy* yaitu *passthrough* dan *termination* masing - masing dilakukan pengujian dengan algoritma *load balancing* yaitu *least connection* dan *round robin*. Nilai pada tabel tersebut merupakan rata rata kecepatan transfer data dari server menuju klien.



Gambar 19 Perbandingan *Transfer Rate* Pada HTTPS *Proxy*

Dari hasil pengukuran nilai *Transfer Rate* yang tersedia pada gambar 19, pengujian yang dilakukan sebanyak 3 kali memiliki rata – rata sebanyak 90 MB/s pada

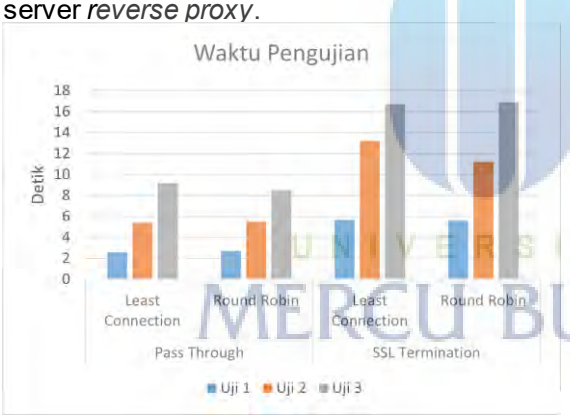
passthrough dan 48 MB/s pada *Termination*. Oleh karena itu metode *passthrough* memiliki *Transfer Rate* yang lebih cepat daripada metode *termination*. Selanjutnya pada *load balancing*, algoritma *round robin* memiliki rata – rata hasil lebih baik sebanyak 3% dibanding *least connection*.

B. Waktu Pengujian

Tabel 3 Waktu Hasil Pengujian

No	Waktu Pengujian (Detik)				
	Parameter	Pass Through		SSL *Termination*	
		LC	RR	LC	RR
1	Uji 1	2.6	2.7	5.7	5.6
2	Uji 2	5.4	5.5	13.2	11.2
3	Uji 3	9.2	8.5	16.7	16.9

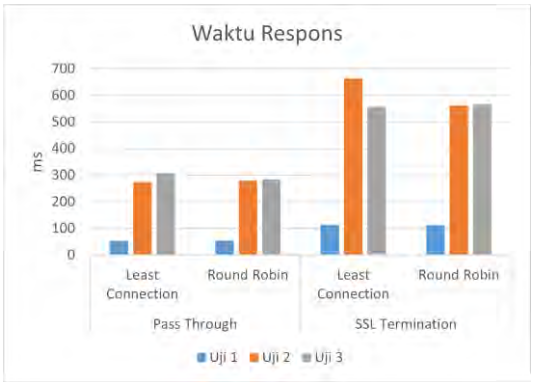
Pada tabel 3 merupakan rata – rata hasil pengukuran waktu dalam satuan detik yang dilakukan setiap pengujian akses server *reverse proxy*.



Gambar 20 Waktu Pengujian

Dari hasil perbandingan waktu pengujian yang pada gambar 20, metode HTTPS *Proxy passthrough* memiliki rata – rata 6.6 detik lebih cepat dibandingkan metode *termination* yang memiliki rata – rata 12.2 detik. Lalu untuk metode *load balancing*, metode *Round robin* dengan rata – rata 9.4 detik dan *Least connection* dengan rata – rata sebanyak 9.5 detik.

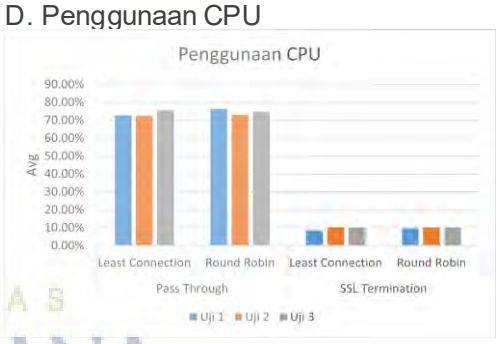
C. *Time Per Request*



Gambar 21 *Time Per Request*

Dari hasil perbandingan *time per request* pada gambar 21, metode *passthrough* memiliki waktu 0.6 *milisecond* detik lebih cepat dibandingkan metode *termination* yang memiliki waktu 1.2 *milisecond*. Lalu untuk metode *load balancing* tidak ada perbedaan waktu reponse pada kedua metode tersebut.

D. Penggunaan CPU

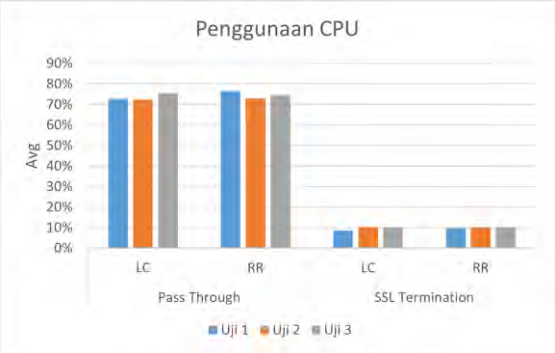


Gambar 22 Rata - Rata Penggunaan CPU

Gambar 22 merupakan hasil pengukuran nilai rata - rata pemakaian CPU pada setiap web server. Gambar tersebut menunjukan bahwa metode *termination* yang memiliki rata - rata pemakaian CPU dibawah 10% lebih baik dibandingkan *passthrough* yang memiliki rata – rata penggunaan 70%. Lalu perbandingan setiap metode *load balancing* memiliki penggunaan CPU yang bervariasi pada *least connection*, namun pada *round robin* memiliki penggunaan CPU yang sama.

E. Perbandingan Server Tunggal dengan Load Balancer.

Pada bagian ini akan dibandingkan performa dari server tunggal dan server yang sudah menggunakan load balancer.



Gambar 23 Penggunaan CPU

Dari hasil pengukuran nilai rata - rata pemakaian CPU pada setiap web server, pada gambar 23 menunjukan bahwa metode yang paling sedikit menggunakan CPU untuk mengolah request yang sama adalah metode *round robin* dengan SSL *termination*.

Tabel 4 *Success Rate*

No	Parameter	Single Server	*Reverser proxy*
1	Uji1	100%	100%
2	Uji2	99.91%	100%
3	Uji3	99.87%	100%

Tabel 4 menampilkan nilai hasil pengujian yang dilakukan pada seluruh *load balancing* pada *reverser proxy* dan server tunggal. hasilnya pada pengujian yang menggunakan server tunggal memiliki penurunan performa pada pengujian *10000 request 500 concurrent* dan *15000 request 500 concurrent*. Maka dapat disimpulkan bahwa *reverse proxy* memiliki performa kesuksesan 100% pada setiap pengujiannya.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Pada bagian ini merupakan bagian akhir dalam melakukan penelitian setelah melakukan pengujian dan analisa dari sistem yang telah dirancang. Berikut beberapa kesimpulan dari penulis dari hasil penelitian:

1. Penerapan sistem *reverser proxy* untuk kebutuhan load balancer dapat memberikan performa yang lebih baik terhadap layanan kepada klien. Hasil pengukuran terhadap *time per request*, waktu pengujian, *Transfer Rate*, success rate dan kinerja CPU pada *reverser proxy* menunjukan nilai yang lebih baik dibandingkan dengan menggunakan single server.
2. Penggunaan HTTPS *Proxy* dengan metode *termination* lebih meringankan kinerja server dibanding dengan menggunakan metode *passthrough*,
3. Kecepatan server dalam melayani request klien lebih baik pada metode *passthrough* dibanding dengan metode *termination*.
4. Pada penelitian ini algoritma *round robin* dapat memberikan performa lebih baik dibandingkan dengan *least connection* . Dan memberikan beban kinerja yang merata pada setiap server.

4.2 Saran

Terdapat beberapa saran bagi penelitian serupa yang akan datang yaitu:

1. Dapat mengimplementasikan sistem *reverser proxy* pada environment lainnya dengan kondisi lingkungan yang lebih nyata.
2. Mengembangkan algoritma selain *round robin* dan *least connection*.
3. Melakukan penelitian dengan tools yang lain agar dapat mendapatkan parameter yang lebih bervariasi.

Referensi

Bansal, L. (2020). Google App Engine - What is It, Its Advantages, And Why You Should Use It. Retrieved July 6, 2022, from c-sharpcorner.com website: https://www.c-sharpcorner.com/article/google-app-engine-what-is-it-its-advantages-and-why-you-should-use-it/

Bella, M. R. M., Data, M., & Yahya, W. (2018). Web Server *Load balancing* Based On Memory Utilization Using Docker Swarm. *3rd International Conference on Sustainable Information Engineering and Technology, SIET 2018 - Proceedings*, 220–223. https://doi.org/10.1109/SIET.2018.8693212

Bustomi, Z., Syahiruddin, M., Afandi, M. I., Fahmi, K., & Holle, H. (2019). *Load balancing* Web Server Menggunakan Nginx pada Lingkungan Virtual. *Jurnal Informatika …*, (xx), 32–36. Retrieved from http://repository.uin-malang.ac.id/9919/

Dody Firmansyah, M., Bachtiar, S., Sfenrianto, S., & Robert Kaburuan, E. (2019). Sales Information System Using Web for Small Business (Case

Study: Cv. Tanaka Service). *International Journal of Mechanical Engineering and Technology (IJMET)*, *10*(3), 1696–1702. Retrieved from http://www.iaeme.com/IJMET/index.asp1696http://www.iaeme.com/ijmet/issues.asp?JType=IJMET&VType=10&IType=3http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=10&IType=3

Dr. Pierre, D. B. (2020). To Terminate or Not to Terminate Secure Sockets Layer (SSL) Traffic at the Load Balancer. *To Terminate or Not to Terminate Secure Sockets Layer (SSL) Traffic at the Load Balancer*. Retrieved from https://www.researchgate.net/publication/345902285_To_Terminate_or_Not_to_Terminate_Secure_Sockets_Layer_SSL_Traffic_at_the_Load_Balancer

Dwi Anggraini, A., & W. Finaka, A. (2018). 10 Prioritas Nasional: Making Indonesia 4.0 | Indonesia Baik. Retrieved July 6, 2022, from indonesiabaik.id website: https://indonesiabaik.id/infografis/10-prioritas-nasional-making-indonesia-40

Friyanto, A. (2019). Hyper Text Transfer Protokol for Securing Packet Inspection in Intrusion Prevention System Device. *IOP Conference Series: Materials Science and Engineering*, *662*(2), 22021. https://doi.org/10.1088/1757-899x/662/2/022021

Ibrahim, I. M., Ameen, S. Y., Yasin, H. M., Omar, N., Kak, S. F., Rashid, Z. N., … Ahmed, D. M. (2021). Web Server Performance Improvement Using Dynamic *Load balancing* Techniques: A Review. *Asian Journal of Research in Computer Science*, 47–62. https://doi.org/10.9734/ajrcos/2021/v10i130234

Junandia, R. (2020). *Implementasi Metode Load Balancer Dan Failover Untuk Api Sms Gateway Pada Pt. Kb Finansia Multi Finance*. Retrieved from https://repository.bsi.ac.id/index.php/repo/viewitem/27823

Putra, J. P. (2018). *KAJIAN WEB LOAD BALANCING BERBASIS ROUND ROBIN DAN IP HASH UNTUK PENINGKATAN KINERJA LAYANAN SERVER*. *7*, 1–25.

Radivilova, T., Kirichenko, L., Ageyev, D., Tawalbeh, M., & Bulakh, V. (2018). Decrypting SSL/TLS traffic for hidden threats detection. *Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018*, 143–146. https://doi.org/10.1109/DESSERT.2018.8409116

Riskiono, S. D., & Pasha, D. (2020). Analisis Metode *Load balancing* Dalam Meningkatkan Kinerja Website E-Learning. *Jurnal Teknoinfo*, *14*(1), 22. https://doi.org/10.33365/jti.v14i1.466

Robertson, G. (2019). NGINX Is Now Officially Part of F5 - NGINX. Retrieved July 6, 2022, from nginx.com website: https://www.nginx.com/blog/nginx-is-now-officially-part-of-f5/

Satwika, I. K. S., & Semadi, K. N. (2020). Perbandingan Performansi Web Server Apache Dan Nginx Dengan Menggunakan Ipv6. *SCAN - Jurnal Teknologi Informasi Dan Komunikasi*, *15*(1), 10–15. https://doi.org/10.33005/scan.v15i1.1847

Tedyyana, A. (2020). Implementasi Secure Socket Layer Pada Aplikasi Computer Assisted Test Komisi Pemilihan Umum Bengkalis. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, *11*(1), 71–80. https://doi.org/10.31849/digitalzone.v11i1.3859

Wen, Z., Li, G., & Yang, G. (2018). Research and Realization of Nginx-based Dynamic Feedback *Load balancing* Algorithm. *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2541–2546. https://doi.org/10.1109/IAEAC.2018.8577911

Wesley, C., & Ferguson, K. (2021). What is HTTP and how does it work? Hypertext Transfer Protokol Definition. Retrieved July 6, 2022, from techtarget.com website: https://www.techtarget.com/whatis/definition/HTTP-Hypertext-Transfer-Protokol

KERTAS KERJA

Ringkasan

Kertas kerja ini merupakan material kelengkapan artikel jurnal dengan judul di atas. Kertas kerja berisi semua material hasil penelitian hasil Tugas Akhir yang tidak dimuat atau disertakan di artiker jurnal. Di dalam kertas kerja ini disajikan:

1.Literature review

Merupakan tinjaun pustaka yang digunakan untuk memahami suatu topik yang akan dikerjakan dapat berasal dari jurnal, buku, internet maupun sumber resmi lainnya. Di dalam literatur review dapat berupa teori maupun gagasan dari penelitian lain yang mereka peroleh dengan melakukan penelitian.

2.Analisa dan Perancangan

Melakukan analisa serta perancangan sistem yang dibutuhkan mulai dari software, hardware, struktur topologi, jalur komunikasi sistem dll. Digunakan untuk menunjang keberhasilan dalam penelitian.

3.Dataset

Pada dataset berupa hasil pengujian terhadap suatu sistem sebelum dilakukan implementasi sistem yang akan dilakukan. Dataset dapat digunakan sebagai dasar pebandingan apabila telah melakukan penelitian.

4.Source code

Dijelaskan mengenai lingkungan sistem yang digunakan, perintah eksekusi untuk menjalankan program, sorce code suatu sistem. Menjelaskan berbagai konfigurasi sebelum memulai penelitian.

5.Tahapan eksperimen

Merupakan tahapan eksperimen yang dilakukan pada penelitian mulai dari analisa sistem, perancangan sistem, konfigurasi sistem, hingga dapat dilakukan implementasi. Terdapat diagram alur, flow chart dll.

6.Hasil eksperimen

Pada bagian ini memuat semua hasil eksperiman yang dikerjakan. Dapat ditulis dalam berupa tabel maupun gambar pengujian. Terdapat kesimpulan hasil penelitian dan saran dari peneliti.