

**ANALISIS PERBANDINGAN *LOAD BALANCING* ANTARA  
NGINX DAN TRAEFIK PADA APLIKASI WORDPRESS  
BERBASIS DOCKER CONTAINER**

Skripsi



oleh:

MATIUS KRISNA GUNARNO

71170196

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS KRISTEN DUTA WACANA

YOGYAKARTA

2021

**ANALISIS PERBANDINGAN *LOAD BALANCING* ANTARA  
NGINX DAN TRAEFIK PADA APLIKASI WORDPRESS  
BERBASIS DOCKER CONTAINER**

Skripsi



Diajukan kepada Fakultas Teknologi Informasi Program Studi Informatika

Universitas Kristen Duta Wacana

Sebagai salah satu syarat dalam memperoleh gelar Sarjana Komputer

Disusun oleh:

**MATIUS KRISNA GUNARNO**

71170196

**PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS KRISTEN DUTA WACANA**

**YOGYAKARTA**

2021

## **HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Kristen Duta Wacana, saya yang bertanda tangan di bawah ini:

Nama : Matius Krisna Gunarno  
NIM : 71170196  
Program studi : Informatika  
Fakultas : Teknologi Informasi  
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Kristen Duta Wacana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

**“ANALISIS PERBANDINGAN *LOAD BALANCING* ANTARA NGINX DAN TRAEFIK PADA APLIKASI WORDPRESS BERBASIS DOCKER CONTAINER”**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Kristen Duta Wacana berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama kami sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta  
Pada Tanggal : 12 Agustus 2021

Yang menyatakan



(Matius Krisna Gunarno)

NIM.71170196

## PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

### **ANALISIS PERBANDINGAN LOAD BALANCING ANTARA NGINX DAN TRAEFIK PADA APLIKASI WORDPRESS BERBASIS DOCKER CONTAINER**

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan sarjana Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, maka saya bersedia dikenai sanksi, yakni pencabutan gelar kesarjanaan saya.

Yogyakarta, 9 Juli 2021



MATIUS KRISNA GUNARNO

71170196

## HALAMAN PERSETUJUAN

Judul : Analisis Perbandingan *Load Balancing* antara NGINX dan Traefik pada Aplikasi WordPress Berbasis Docker Container.

Nama : Matius Krisna Gunarno

NIM : 71170196

Mata Kuliah : Skripsi (Tugas Akhir)

Kode : TIW276

Semester : Genap

Tahun akademik : 2020/2021

Telah diperiksa dan disetujui di  
Yogyakarta,

Pada Tanggal 22 Mei 2021

### Dosen Pembimbing I

  
Willy Sudiarto  
Raharjo

Digitally signed by: Willy  
Sudiarto Raharjo  
Reason: I am approving this  
document  
Email: willysr@staff.ukdw.ac.id  
Date: 2021/05/24 15:12:16  
+07'00'

Willy Sudiarto Raharjo, S.Kom, M.Cs

### Dosen Pembimbing II



Nugroho Agus Haryono, S.Si., M.Si.

## HALAMAN PENGESAHAN

### SKRIPSI

Analisis Perbandingan Load Balancing antara NGINX dan Traefik pada Aplikasi  
WordPress Berbasis Docker Container.

Oleh: 71170196 / Matius Krisna Gunarno

Dipertahankan di depan dewan Penguji Skripsi  
Program Studi Informatika Fakultas Teknologi Informasi  
Universitas Kristen Duta Wacana – Yogyakarta  
Dan dinyatakan diterima untuk memenuhi salah satu

Syarat memperoleh gelar

Sarjana Komputer

Pada tanggal

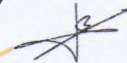


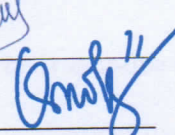
11 Juni 2021

Yogyakarta, 2 Agustus 2021

Mengesahkan,

Dewan Penguji:


1. Willy Sudiarto Raharjo, S.Kom, M.Cs
2. Nugroho Agus Haryono, S.Si., M.Si.
3. Prihadi Beny Waluyo, S.Si., MT
4. Ir. Gani Indriyanta, MT

  
\_\_\_\_\_  
  
\_\_\_\_\_  
  
\_\_\_\_\_  
  
\_\_\_\_\_

Dekan

  
  
Restyandito, S.Kom, MSIS., Ph.D

Ketua Program Studi

  
Gloria Virginia, S.Kom., MAI., Ph.D

## UCAPAN TERIMA KASIH

Puji syukur saya sampaikan kepada Tuhan saya berdoa: Bapa, Putra, dan Roh Kudus, yang memberikan tuntunan dan kekuatan, sehingga skripsi berjudul “Analisis Perbandingan *Load Balancing* antara NGINX dan Traefik pada Aplikasi WordPress Berbasis Docker Container” ini dapat diselesaikan. Ucapan terima kasih ini juga saya persembahkan kepada:

Pembimbing-pembimbing saya, Pak Willy Sudiarto Raharjo, S.Kom, M.Cs dan Pak Nugroho Agus Haryono, S.Si., M.Si. Maaf juga kalau saya banyak salahnya.

Para dosen penguji selain yang disebut di atas, Prihadi Beny Waluyo, S.Si., MT, dan Ir. Gani Indriyanta, MT.

Keluarga yang saya cintai: Ibu, Bapak di Surga, Mbak Wai, Mbak Ngah, Dedek, dan masih banyak lagi.

Keluarga yang khusus menyokong hidup saya selama di Yogyakarta: Mas Eko, Mbak Wai, Dek Keysha, dan masih banyak lagi.

Anna, yang ikut terlibat dan melihat saya dalam suka duka hidup bersama skripsi di semester tua.

Kepada mereka, teman-teman seangkatan, atau pun yang beda satu-dua tahun.

Teman-teman DevOps Blibli.com yang sedikit banyak memberikan ilmu luar biasa, khususnya mengenai tetek-bengek yang berkaitan dengan karya ini.

Semua bagian hidup saya dalam menjalani perkuliahan di Prodi Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana.

Matius Krisna Gunarno

## INTISARI

### ANALISIS PERBANDINGAN *LOAD BALANCING* ANTARA NGINX DAN TRAEFIK PADA APLIKASI WORDPRESS BERBASIS DOCKER CONTAINER

Pada penelitian ini, penulis berusaha untuk membandingkan performa NGINX dan Traefik dengan menerapkannya pada arsitektur aplikasi WordPress yang berjalan di atas docker container, pada platform *cloud computing* milik AWS, yaitu AWS EC2. Dalam penelitian ini, metrik yang diukur dalam membandingkan kedua aplikasi tersebut adalah *request per second*, *latency*, dan *error rate*. Alat yang digunakan untuk mengukur metrik serta alat untuk melakukan pengujian (memberikan beban *traffic* pada sistem) adalah hey.

Penelitian terbagi atas pengujian menggunakan *static content* dan *dynamic content*. Di dalamnya, skenario pengujian dibagi lagi berdasarkan *concurrency*-nya (250, 500, 750, 1000). *Static content* yang diuji adalah *readme.html*, sedangkan *dynamic content* menggunakan *wp-login.php*. Terdapat pula pengujian dengan mengunggah ke dalam WordPress dan jumlah *node* yang diperbanyak menjadi 5 *node*.

Hasil penelitian yang didapat menunjukkan bahwa NGINX lebih unggul, baik dari *request per second*, *latency*, dan *error rate*. NGINX menunjukkan hasil yang konsisten sesuai dengan jumlah permintaan yang dikirim. Sedangkan Traefik secara umum cukup mendekati NGINX walau masih terdapat kekurangan, terutama pada *latency* yang melonjak ketika harus melayani 90% lebih dari total permintaan yang dikirimkan. Disarankan untuk menambah variasi skenario dan layanan AWS yang digunakan untuk pengembangan lebih lanjut.

Kata kunci: NGINX, Traefik, *load balancer*.



## DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI.....	i
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
UCAPAN TERIMA KASIH.....	v
INTISARI.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	x
DAFTAR GRAFIK.....	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Manfaat Penelitian.....	3
1.5 Metodologi Penelitian.....	3
1.6 Metodologi Pengujian.....	4
1.8 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	6
2.1 Tinjauan Pustaka.....	6
2.2 Landasan Teori.....	13
2.2.1 Virtualisasi dan <i>Virtual Machine</i> .....	13
2.2.2 Docker Container.....	15
2.2.3 Load Balancing.....	16
2.2.4 NGINX.....	17
2.2.5 Traefik.....	18
2.2.6 AWS.....	19
2.2.7 AWS EC2.....	19
BAB III METODOLOGI PENELITIAN.....	21
3.1 Kebutuhan Non Fungsional.....	21

3.2	Kebutuhan Fungsional.....	22
3.3	Metodologi Pengembangan dan Pengujian Sistem .....	24
3.4	Arsitektur Sistem.....	28
BAB IV HASIL DAN PEMBAHASAN .....		29
4.1	Uji <i>Static Content</i> .....	29
4.1.1	<i>Request per Second</i> .....	29
4.1.2	<i>Latency</i> .....	31
4.1.3	<i>Error Rate</i> .....	34
4.2	Uji <i>Dynamic Content</i> .....	35
4.2.1	<i>Request per Second</i> .....	35
4.2.2	<i>Latency</i> .....	37
4.2.3	<i>Error Rate</i> .....	40
4.3	Uji Unggah Konten (POST).....	41
4.3.1	<i>Request per Second</i> .....	41
4.3.2	<i>Latency</i> .....	43
4.3.3	<i>Error Rate</i> .....	46
4.4	Uji <i>Dynamic Content</i> dengan 5 node .....	47
4.4.1	<i>Request per Second</i> .....	47
4.4.2	<i>Latency</i> .....	49
4.4.3	<i>Error Rate</i> .....	52
4.5	Pembahasan.....	53
4.5.1	NGINX.....	53
4.5.2	Traefik.....	55
BAB V KESIMPULAN DAN SARAN.....		57
5.1	Kesimpulan.....	57
5.2	Saran.....	57
DAFTAR PUSTAKA .....		58
LAMPIRAN.....		61
	Data Mentah .....	62
	Skenario Pengujian.....	94
	Konfigurasi Sistem .....	100

## DAFTAR GAMBAR

Gambar 2.1 Hypervisor tipe 1.....	14
Gambar 2.2 Hypervisor tipe 2.....	14
Gambar 2.3 Perbandingan antara container dan VM.....	16
Gambar 2.4 Ilustrasi cara kerja Traefik.....	18
Gambar 3.1 Arsitektur sistem.....	28

©UKDW

## DAFTAR TABEL

Tabel 4.1 <i>Request per Second</i> skenario <i>static content</i> .....	29
Tabel 4.2 <i>Latency static content</i> pada NGINX.....	31
Tabel 4.3 <i>Latency static content</i> pada Traefik.....	31
Tabel 4.4 <i>Error Rate</i> pada <i>static content</i> .....	34
Tabel 4.5 <i>Request per Second</i> skenario <i>dynamic content</i> .....	35
Tabel 4.6 <i>Latency dynamic content</i> pada NGINX.....	37
Tabel 4.7 <i>Latency dynamic content</i> pada Traefik.....	37
Tabel 4.8 <i>Error Rate</i> pada <i>dynamic content</i> .....	40
Tabel 4.9 <i>Request per Second</i> skenario unggah konten.....	41
Tabel 4.10 <i>Latency</i> unggah konten pada NGINX.....	43
Tabel 4.11 <i>Latency</i> unggah konten pada Traefik.....	43
Tabel 4.12 <i>Error Rate</i> pada unggah konten .....	46
Tabel 4.13 <i>Request per Second</i> skenario <i>dynamic conten</i> dengan <i>5 node</i> .....	47
Tabel 4.14 <i>Latency dynamic content</i> pada NGINX dengan <i>5 node</i> .....	49
Tabel 4.15 <i>Latency dynamic content</i> pada Traefik dengan <i>5 node</i> .....	49
Tabel 4.16 <i>Error Rate</i> pada <i>dynamic content</i> dengan <i>5 node</i> .....	52

## DAFTAR GRAFIK

Grafik 4.1 <i>Request per Second</i> skenario <i>static content</i> .....	30
Grafik 4.2 <i>Latency static content</i> pada NGINX.....	33
Grafik 4.3 <i>Latency static content</i> pada Traefik.....	33
Grafik 4.4 <i>Error Rate</i> pada <i>static content</i> .....	35
Grafik 4.5 <i>Request per Second</i> skenario <i>dynamic content</i> .....	36
Grafik 4.6 <i>Latency dynamic content</i> pada NGINX.....	39
Grafik 4.7 <i>Latency dynamic content</i> pada Traefik.....	39
Grafik 4.8 <i>Error Rate</i> pada <i>dynamic content</i> .....	41
Grafik 4.9 <i>Request per Second</i> skenario unggah konten.....	42
Grafik 4.10 <i>Latency</i> unggah konten pada NGINX.....	45
Grafik 4.11 <i>Latency</i> unggah konten pada Traefik.....	45
Grafik 4.12 <i>Error Rate</i> pada unggah konten .....	47
Grafik 4.13 <i>Request per Second</i> skenario <i>dynamic conten</i> dengan 5 <i>node</i> ....	48
Grafik 4.14 <i>Latency dynamic content</i> pada NGINX dengan 5 <i>node</i> .....	51
Grafik 4.15 <i>Latency dynamic content</i> pada Traefik dengan 5 <i>node</i> .....	51
Grafik 4.16 <i>Error Rate</i> pada <i>dynamic content</i> dengan 5 <i>node</i> .....	53

# BAB I PENDAHULUAN

## 1.1 Latar Belakang Masalah

Perkembangan dunia komputer, khususnya pada peningkatan kualitas layanan dan ketersediaan yang diberikan oleh sebuah aplikasi, semakin pesat setiap tahunnya. Terlebih ketika terjadi pandemi COVID-19 yang memaksa kita untuk lebih banyak berinteraksi dengan layanan daring. Untuk memenuhi kebutuhan tersebut diperlukan sebuah arsitektur komputer yang mampu menangani permintaan pengguna secara maksimal, serta metode tertentu yang memungkinkan sebuah layanan aplikasi bisa berjalan baik dengan kegagalan yang minim. Jika terjadi kegagalan, akan ada *service* cadangan yang bisa mengakomodir permintaan tersebut, sehingga dicapai *service* yang bersifat *high availability*. Tentu saja, bukan hal mudah untuk membangun sebuah layanan aplikasi web yang bersifat *high availability*, terlebih jika harus menangani permintaan yang sangat tinggi. Namun dengan banyaknya metode, alat, dan perangkat lunak tertentu yang bisa menangani hal tersebut, kita diberikan lebih banyak opsi *load balancing* dalam rangka mewujudkan sistem yang memiliki *high availability*. Di antaranya adalah aplikasi NGINX dan Traefik.

Pada penelitian ini, penulis berusaha untuk membandingkan kedua aplikasi *load balancing* tersebut dengan menerapkannya pada arsitektur aplikasi WordPress pada platform *cloud computing* milik AWS, yaitu AWS EC2. Dalam penelitian ini, metrik yang diukur dalam membandingkan kedua aplikasi tersebut adalah *request per second*, *latency*, dan *error rate*. Alat yang digunakan untuk mengukur metrik serta alat untuk melakukan pengujian (memberikan beban *traffic* pada sistem) adalah hey. Penelitian ini diharapkan dapat memberikan informasi pada kita bagaimana perbandingan antara NGINX dan Traefik dalam melakukan *load balancing* pada suatu *container* WordPress.

## 1.2 Rumusan Masalah

Pada penelitian ini, fokus permasalahan yang berusaha dijawab penulis adalah bagaimanakah performa dari NGINX dan Traefik dalam melakukan *load balancing* pada *container* milik WordPress, serta perbandingannya satu sama lain berdasarkan data-data metrik yang ada.

## 1.3 Batasan Masalah

- 1) Penelitian ini dilakukan di atas lingkungan *cloud* milik AWS yang tersusun atas mesin EC2 (tempat aplikasi dijalankan).
- 2) Algoritma *load balancing* yang digunakan adalah *Round Robin*.
- 3) Protokol yang digunakan adalah protokol HTTPS, dengan HTTP *Method* GET dan POST.
- 4) Jenis konten yang diujikan adalah *static content* dan *dynamic content*.
- 5) NGINX dan Traefik dijalankan di dalam sebuah *container* dan dijalankan pada *instance* yang berbeda, tetapi sama persis secara topologi.
- 6) Tidak membahas tentang aplikasi WordPress itu sendiri dan *tools* untuk pengujian.
- 7) Tidak membahas keamanan sistem.

## 1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah untuk mengetahui bagaimana perbandingan antara NGINX dan Traefik, serta mengetahui performa masing-masing aplikasi tersebut dalam melakukan *load balancing* layanan WordPress yang berjalan di atas docker container.

## 1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah memberikan gambaran mendetail tentang performa dari NGINX dan Traefik, khususnya dalam hal *load balancing* suatu *container*, serta dapat dijadikan acuan bagi pengguna bila ingin menggunakan salah satu aplikasi tersebut.

## 1.5 Metodologi Penelitian

Metode yang digunakan dalam penelitian ini terdiri atas studi literatur dan pembangunan arsitektur sistem. Singkatnya dijabarkan sebagai berikut:

### 1.5.1 Studi literatur

Peneliti akan mengumpulkan literatur/pustaka yang berkaitan dengan *load balancing*, serta sumber-sumber lain yang dianggap perlu dan memadai.

### 1.5.2 Pembangunan Arsitektur Sistem

Berikut tahap-tahap pembangunan sistem yang menjadi dasar dalam penelitian ini:

- 1) Membuat skema arsitektur WordPress yang berbasis *container*.
- 2) Melakukan instalasi *tools* dasar.
- 3) Melakukan instalasi WordPress menggunakan Docker.
- 4) Melakukan instalasi NGINX dan Traefik.
- 5) Melakukan instalasi aplikasi yang diperlukan dalam pengujian.



## 1.6 Metodologi Pengujian

Pengujian dilakukan dengan membandingkan data metrik dari kedua *instance* yang diuji. Metrik tersebut didapatkan melalui aplikasi *monitoring* yang saling bekerjasama dalam menghimpun data dari setiap *instance*. Data metrik yang dimaksud itu sendiri adalah *request per second*, *latency*, dan *error rate*. Semua itu akan diuji menggunakan hey, yaitu aplikasi *load tester* sederhana yang mampu memberikan beban kepada *load balancer* serta hasil pengujian secara keseluruhan.

Dalam skenario yang akan dilakukan dalam penelitian akan ada dua jenis pengujian, yaitu pengujian menggunakan laman yang bersifat *static content* dan yang menggunakan *dynamic content*.

## 1.8 Sistematika Penulisan

Sistematika penulisan yang ada dalam penelitian kali ini terdiri dari struktur seperti berikut:

Bab I merupakan pendahuluan yang berisi gambaran besar dari penelitian yang penulis lakukan. Gambaran besar tersebut dinyatakan dalam latar belakang penelitian, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta gambaran besar dari metodologi penelitiannya. Di dalam bab ini, penulis berusaha memberikan gambaran umum yang mudah dipahami mengenai penelitian ini, serta memberi latar belakang yang jelas beserta batasan-batasan yang dianggap perlu.

Bab II merupakan himpunan dari landasan pengetahuan yang terdapat dalam penelitian, yaitu tinjauan pustaka yang merupakan referensi dari penelitian serupa dan landasan teori yang berisi segala teori relevan dan penting. Di dalam bab ini, penulis berusaha memberikan landasan keilmuan dan pengetahuan yang mampu menunjang penelitian ini, sekaligus memberikan pengetahuan mengenai istilah-istilah yang banyak muncul di dalam penelitian.

Bab III merupakan gambaran besar akan rancangan dan metode penelitian yang dilakukan, baik penggunaan perangkat keras, perangkat lunak, bagaimana penelitian dijalankan, dan lain sebagainya. Bab ini terdiri atas kebutuhan fungsional, kebutuhan non fungsional, serta metodologi pembangunan dan

pengujian sistem, termasuk gambaran rancangan arsitektur sistemnya. Di dalam bab ini, penulis memberikan gambaran teknis mengenai pelaksanaan penelitian dan kebutuhan-kebutuhan yang diperlukan di dalam penelitian.

Bab IV berisi hasil dari penelitian beserta pembahasannya. Semua data yang berhasil dihimpun kemudian dijabarkan dan diteliti sesuai dengan kecenderungan pada data yang dimaksud. Hasil dan pembahasan terbagi sesuai dengan skenario-skenario pengujian yang sudah ditentukan sebelumnya, dan diharapkan mampu menghasilkan informasi yang faktual dan dapat dipertanggungjawabkan.

Bab V merupakan bagian akhir yang berisi simpulan atas seluruh penelitian yang dilakukan dan saran-saran yang dirasa perlu untuk pengembangan penelitian lebih lanjut. Bab ini merupakan kelanjutan langsung daripada bab sebelumnya, di mana penulis menarik hasil akhir secara umum atas keseluruhan penelitian dan ada. Penulis juga memperhatikan kekurangan-kekurangan penelitian yang telah dilakukan agar sekiranya dapat diperbaiki lebih lanjut oleh peneliti selanjutnya yang hendak melanjutkan penelitian ini.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Secara umum, ketika NGINX dan Traefik dijalankan di atas Docker Container untuk melakukan *load balancing* dua *container* WordPress, NGINX mengungguli Traefik hampir dalam setiap skenario pengujian, baik dalam melayani permintaan *static content*, *dynamic content*, unggah konten, dan *dynamic content* menggunakan 5 *node*. Perbedaan performa paling mencolok dapat dilihat pada uji *latency*, di mana ketika memasuki persentil 90% ke atas, terjadi peningkatan distribusi *latency* yang signifikan pada Traefik, baik pada *static content* dan *dynamic content*. Lalu, *Request per second* juga memberi andil perbedaan yang mencolok, di mana NGINX mampu menyesuaikan peningkatan jumlah permintaan yang dilayani dengan *concurrency*-nya, sedangkan Traefik tidak mampu melakukan hal serupa sebaik NGINX, terlebih pada pengujian *static content*. *Error rate* pada NGINX dan Traefik terlihat tidak banyak perbedaan yang terlalu signifikan meskipun NGINX unggul tipis (hanya berselisih 0.5% hingga 5% pada setiap skenario), dan Traefik unggul pada kondisi khusus (*concurrency* 750 dan 1000) pada pengujian unggah konten dan 5 *node*.

#### 5.2 Saran

Penulis berharap agar penelitian yang selanjutnya dapat memanfaatkan layanan AWS dengan lebih luas lagi, seperti RDS (layanan *database*), ALB (*load balancer*), ataupun CloudFront (*content delivery network*). Semakin luas layanan AWS yang menjadi ruang lingkup penelitian, maka semakin luas pula kemungkinan-kemungkinan baru yang tidak tercakup dalam penelitian ini.

Penulis juga memberi saran agar penelitian selanjutnya dapat mengembangkan skenario uji yang lebih bervariasi, terutama pada sisi *HTTP Method* yang sayangnya masih terbatas pada penelitian ini (GET dan POST). Tidak menutup kemungkinan juga terdapat skenario-skenario lain yang bisa diteliti lebih lanjut, seperti perbedaan algoritma, jumlah *node*, dan lain sebagainya.

## DAFTAR PUSTAKA

- Ahmed, A., & Pierre, G. (2018, July). *Docker container deployment in fog computing infrastructures*. In 2018 IEEE International Conference on Edge Computing (EDGE) (pp. 1-8). IEEE.
- Bernstein, D. (2014). *Containers and cloud: From LXC to Docker to Kubernetes*. *IEEE Cloud Computing*, 1(3), 81-84.
- Bourke, T. (2001). *Server load balancing*. California: O'Reilly Media, Inc.
- Chiueh, S. N. T. C., & Brook, S. (2005). *A survey on virtualization technologies*. Rpe Report, 142.
- DeJonghe, D. (2018). *Load Balancing in the Cloud: Practical Solutions with NGINX and AWS*. California: O'Reilly Media, Inc.
- Docker, 2020. *What is a Container?*. (September 30, 2020) Diambil dari <https://www.docker.com/resources/what-container>.
- Haryanto, M. D., & Riadi, I. (2014). Analisis dan Optimalisasi Jaringan Menggunakan Teknik Load Balancing (Studi Kasus Jaringan UAD Kampus 3) (Doctoral dissertation, Universitas Ahmad Dahlan).
- Fadillah, Fanny & Sari, Rizqi Indah. (2019). *IMPLEMENTASI LOAD BALANCING NGINX ALGORITMA WEIGHTED ROUND ROBIN*. Universitas Siliwangi.
- Islam, T., & Hasan, M. S. (2017, October). *A performance comparison of load balancing algorithms for cloud computing*. In 2017 International Conference on the Frontiers and Advances in Data Science (FADS) (pp. 130-135). IEEE.
- Julianto, R., Yahya, W., & Akbar, S. R. (2017). *Implementasi Load Balancing Di Web Server Menggunakan Metode Berbasis Sumber Daya CPU Pada*

*Software Defined Networking*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2548, 964X.

Mathew, S., & Varia, J. (2014). Overview of Amazon Web Services. Amazon Whitepapers.

Nasser, H., & Witono, T. (2016). *Analisis Algoritma Round Robin, Least Connection, Dan Ratio Pada Load Balancing Menggunakan Opnet Modeler*. Jurnal Informatika, 12(1).

Permana, Y., Ritzkal, R., & Afrianto, Y. (2020). *Load Balancing Method Performance Analysis on Haproxy and Router OS*. Jurnal Mantik, 4(3), 1588-1596.

Portnoy, M. (2012). *Virtualization essentials* (Vol. 19). John Wiley & Sons.

Rosalia, M., Munadi, R., & Mayasari, R. (2016). *Implementasi High Availability Server Menggunakan Metode Load Balancing Dan Failover Pada Virtual Web Server Cluster*. eProceedings of Engineering, 3(3).

Satwika, I., & Semadi, K. N. (2020). *PERBANDINGAN PERFORMANSI WEB SERVER APACHE DAN NGINX DENGAN MENGGUNAKAN IPV6*. SCAN-Jurnal Teknologi Informasi dan Komunikasi, 15(1), 10-15.

Sirajuddin, S., Affandi, A., & Setijadi, E. (2012). *Rancang Bangun Server Learning Management System Menggunakan Load Balancer dan Reverse Proxy*. Jurnal Teknik ITS, 1(1), A50-A52.

Soltész, S., Pözl, H., Fiuczynski, M. E., Bavier, A., & Peterson, L. (2007). *Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors*. In Proceedings of the 2Nd ACM SIGOPS/EuroSys european conference on computer systems 2007 (pp. 275-287).

Traefik, 2020. *Concepts*. (September 30, 2020) Diambil dari  
<https://doc.traefik.io/traefik/getting-started/concepts>.

©UKDW