

**ANALISIS KOMPARATIF PERFORMA INGRESS
CONTROLLER NGINX DAN TRAEFIK PADA
KUBERNETES CLUSTER**

SKRIPSI

Diajukan untuk memenuhi salah satu syarat mencapai derajat Sarjana
Program Studi Informatika



disusun oleh

ANDHIKA BAGUS PRIYOTOMO

22.11.4780

Kepada

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2026

**ANALISIS KOMPARATIF PERFORMA INGRESS
CONTROLLER NGINX DAN TRAEFIK PADA
KUBERNETES CLUSTER**

SKRIPSI

untuk memenuhi salah satu syarat mencapai derajat Sarjana
Program Studi Informatika



disusun oleh

ANDHIKA BAGUS PRIYOTOMO

22.11.4780

Kepada

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2026

HALAMAN PERSETUJUAN

SKRIPSI

**ANALISIS KOMPARATIF PERFORMA INGRESS CONTROLLER
NGINX dan TRAEFIK PADA KUBERNETES CLUSTER**

yang disusun dan diajukan oleh

Andhika Bagus Priyotomo

22.11.4780

telah disetujui oleh Dosen Pembimbing Skripsi
pada tanggal 23 Januari 2026

Dosen Pembimbing,



Hastari Utama, S.Kom., M.Cs.
NIK. 190302230

HALAMAN PENGESAHAN
SKRIPSI
ANALISIS KOMPARATIF PERFORMA INGRESS CONTROLLER
NGINX dan TRAEFIK PADA KUBERNETES CLUSTER

yang disusun dan diajukan oleh

Andhika Bagus Priyotomo

22.11.4780

Telah dipertahankan di depan Dewan Penguji
pada tanggal 23 Januari 2026

Susunan Dewan Penguji

Nama Penguji

Tanda Tangan

Theopilus Bayu Sasongko, S.Kom., M.Eng.
NIK. 190302375

Muhammad Rudyanto Arief, S.T., M.T
NIK. 190302098

Hastari Utama, S.Kom., M.Cs.
NIK. 190302230



Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer
Tanggal 23 Januari 2026

DEKAN FAKULTAS ILMU KOMPUTER



Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN SKRIPSI

Yang bertandatangan di bawah ini,

Nama mahasiswa : Andhika Bagus Priyotomo
NIM : 22.11.4780

Menyatakan bahwa Skripsi dengan judul berikut:

**ANALISIS KOMPARATIF PERFORMA INGRESS CONTROLLER NGINX
DAN TRAEFIK PADA KUBERNETES CLUSTER**

Dosen Pembimbing: Hastari Utama, S.Kom., M.Cs.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 23 Januari 2026

Yang Menyatakan,



Andhika Bagus Priyotomo

HALAMAN PERSEMBAHAN

Alhamdulillah, puji syukur kepada Allah SWT yang telah memberikan berkah, kemudahan, serta kelancaran sehingga dapat menyelesaikan skripsi ini dengan sebaik-baiknya, dan tepat waktu. Skripsi ini dipersembahkan untuk:

1. Kedua orang tua yang memberikan segala bentuk doa dan dukungan sehingga penulis dapat menyelesaikan skripsi ini dengan sebaik-baiknya.
2. Bapak Hastari Utama, S.Kom., M.Cs., selaku dosen pembimbing terimakasih atas waktu, perhatian, nasihat serta masukan yang diberikan selama bimbingan sehingga skripsi ini dapat diselesaikan dengan tepat waktu.
3. Dan terakhir, untuk diri sendiri, ANDHIKA BAGUS PRIYOTOMO, terima kasih atas segala usaha, kerja keras, dan pengorbanan dalam melewati segala tantangan. Perjalanan ini dibangun di atas proses yang tidak mudah, tetapi akhirnya sampai pada titik dimana skripsi ini dapat diselesaikan.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas rahmat dan karunia-Nya, sehingga skripsi ini dapat diselesaikan dengan baik sebagai salah satu syarat untuk memperoleh gelar Sarjana.

Penyusunan skripsi ini tidak lepas dari dukungan dan bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada:

1. Kedua orang tua dan keluarga, atas doa, dan dukungan yang sangat berarti.
2. Bapak Hastari Utama, S.Kom., M.Cs., selaku Dosen Pembimbing, atas waktu, bimbingan, arahan, serta masukan yang sangat berarti dalam penyusunan skripsi ini.
3. Seluruh pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang.

Yogyakarta, 23 Januari 2026

Andhika Bagus Priyotomo

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN SKRIPSI.....	iv
HALAMAN PERSEMBAHAN	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMBANG DAN SINGKATAN	xiv
DAFTAR ISTILAH	xv
INTISARI	xvi
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	6
2.1 Studi Literatur	6
2.2 Dasar Teori.....	14

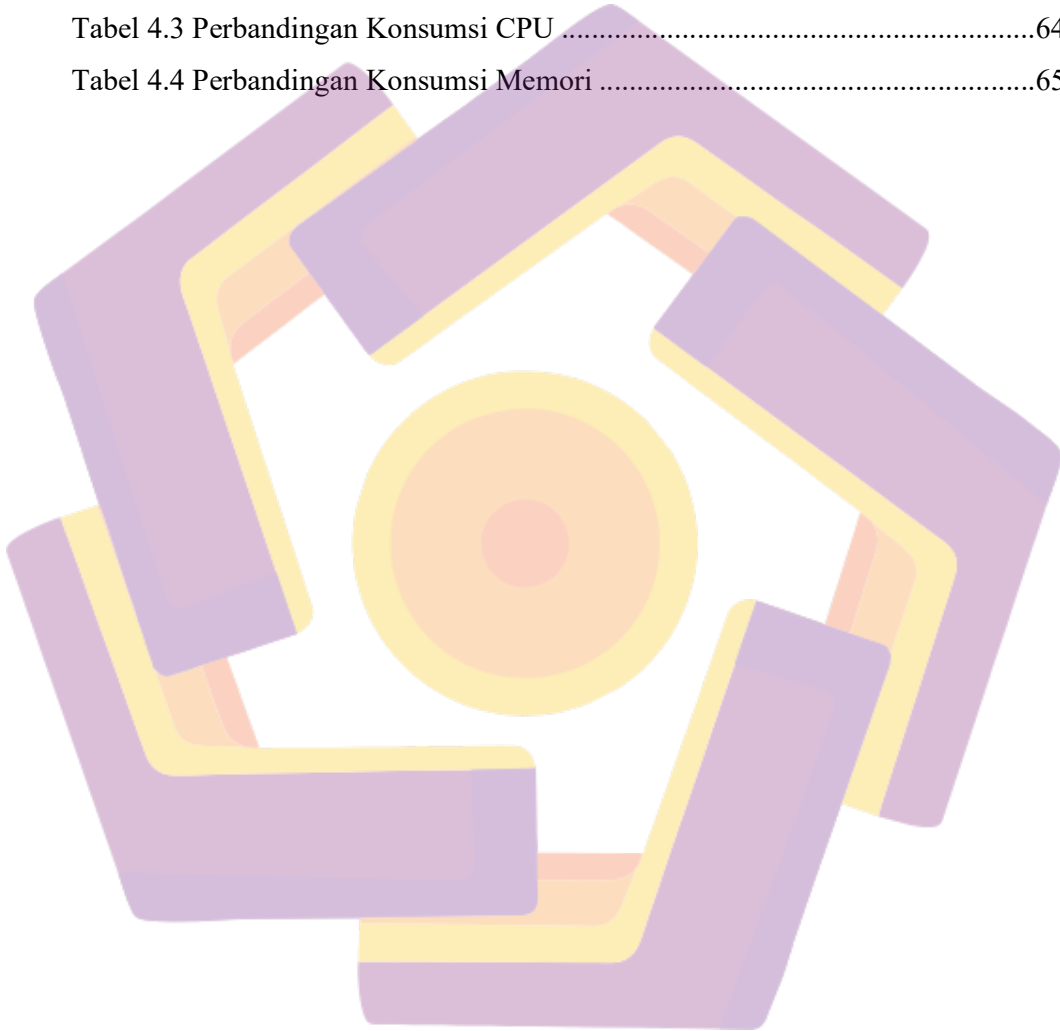
2.3	Kubernetes.....	14
2.4	Ingress Controller.....	17
2.5	Kontainer.....	18
2.6	AWS.....	18
2.7	AWS EKS	19
2.8	Metrik.....	20
2.9	<i>Load Testing</i>	22
2.10	K6	23
2.11	Prometheus	23
2.12	Grafana	24
2.13	<i>Infrastructure as Code</i>	24
2.14	Terraform.....	24
2.15	Helm	25
BAB III METODE PENELITIAN		26
3.1	Alur Penelitian.....	26
3.2	Persiapan	27
3.2.1	Konfigurasi Terraform.....	27
3.2.2	Konfigurasi k6	27
3.2.3	Konfigurasi Helm Repository.....	28
3.3	<i>Provisioning</i>	29
3.4	Instalasi <i>Ingress</i>	29
3.4.1	Instalasi NGINX <i>Ingress Controller</i>	29
3.4.2	Instalasi NGINX <i>Ingress Resource</i>	29
3.4.3	Instalasi Traefik <i>Ingress Controller</i>	30
3.4.4	Instalasi Traefik <i>Ingress Resource</i>	30

3.5	Instalasi Monitoring	30
3.5.1	Instalasi Prometheus	30
3.5.2	Konfigurasi Dashboard Grafana	31
3.6	Skenario Pengujian	31
3.6.1	<i>Load Testing</i>	32
3.6.2	Monitoring Infrastruktur	33
3.7	<i>Destroy</i>	33
3.8	Evaluasi	33
3.8.1	Latensi	33
3.8.2	<i>Throughput</i>	34
3.8.3	<i>Error Rate</i>	34
3.8.4	CPU	34
3.8.5	Memori	35
3.9	Perbandingan Hasil	35
3.10	Alat dan Bahan	35
3.10.1	Alat	35
3.10.2	Bahan	36
BAB IV HASIL DAN PEMBAHASAN		37
4.1	Persiapan	37
4.1.1	Konfigurasi Terraform	37
4.1.2	Konfigurasi K6	43
4.1.3	Konfigurasi Helm Repository	44
4.2	<i>Provisioning</i>	44
4.3	Instalasi <i>Ingress</i>	46
4.3.1	Instalasi NGINX <i>Ingress Controller</i>	46

4.3.2	Instalasi NGINX <i>Ingress Resource</i>	47
4.3.3	Instalasi Traefik <i>Ingress Controller</i>	47
4.3.4	Instalasi Traefik <i>Ingress Resource</i>	48
4.4	Instalasi Monitoring	48
4.4.1	Instalasi Prometheus	48
4.4.2	Konfigurasi Dashboard Grafana	49
4.5	Skenario Pengujian.....	51
4.5.1	<i>Load Testing</i>	51
4.5.2	Monitoring Infrastruktur	53
4.6	<i>Destroy</i>	54
4.7	Evaluasi	55
4.7.1	Latensi.....	55
4.7.2	<i>Throughput</i>	57
4.7.3	<i>Error Rate</i>	59
4.7.4	CPU	60
4.7.5	Memori	62
4.8	Perbandingan Hasil	63
4.9	Hasil Penelitian	65
BAB V PENUTUP		66
5.1	Kesimpulan.....	66
5.2	Saran.....	67
REFERENSI		68

DAFTAR TABEL

Tabel 2.1 Keaslian Penelitian	9
Tabel 4.1 Perbandingan Latensi.....	63
Tabel 4.2 Perbandingan Metrik Throughput dan Error Rate	64
Tabel 4.3 Perbandingan Konsumsi CPU	64
Tabel 4.4 Perbandingan Konsumsi Memori	65

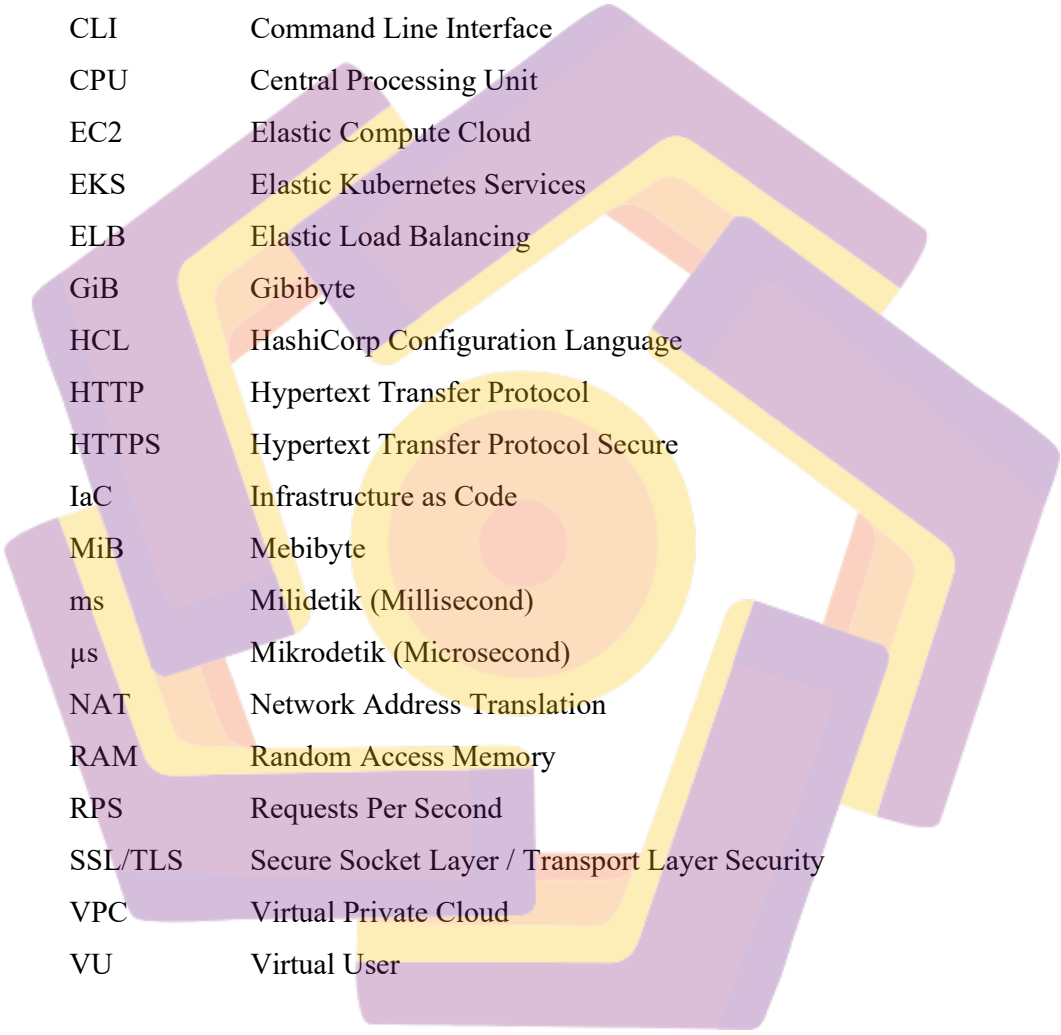


DAFTAR GAMBAR

Gambar 2.1 Arsitektur Kubernetes Cluster.....	16
Gambar 2.2 Arsitektur NGINX Ingress Controller.....	17
Gambar 2.3 Arsitektur Traefik.....	18
Gambar 2.4 Arsitektur AWS.....	19
Gambar 2.5 Arsitektur AWS EKS	20
Gambar 2.6 Arsitektur Prometheus.....	24
Gambar 3.1 Alur Penelitian	26
Gambar 3.2 Rancangan Arsitektur Pengujian.....	32
Gambar 4.1 Konfigurasi Dasar Provider Terraform.....	37
Gambar 4.2 Konfigurasi VPC Cluster NGINX	38
Gambar 4.3 Konfigurasi VPC Cluster Traefik	38
Gambar 4.4 Konfigurasi Terraform Cluster NGINX.....	39
Gambar 4.5 Konfigurasi Terraform Cluster Traefik.....	40
Gambar 4.6 Konfigurasi Security Group Instance k6 NGINX.....	41
Gambar 4.7 Konfigurasi Security Group Instance k6 Traefik.....	41
Gambar 4.8 Instance EC2 Load Generator	42
Gambar 4.9 Output Alamat IP Publik.....	42
Gambar 4.10 Konfigurasi Target dan Metrik Skrip Pengujian K6	43
Gambar 4.11 Konfigurasi Alamat Endpoint Cluster.....	43
Gambar 4.12 Konfigurasi Helm Repository	44
Gambar 4.13 Inisialisasi Modul Terraform	45
Gambar 4.14 Eksekusi Provisioning.....	45
Gambar 4.15 Output Terraform	45
Gambar 4.16 Instalasi NGINX Ingress Controller	46
Gambar 4.17 Instalasi NGINX Ingress Resource	47
Gambar 4.18 Instalasi Traefik Ingress Controller.....	47
Gambar 4.19 Instalasi Traefik Ingress Resource	48
Gambar 4.20 Instalasi Prometheus Beserta Komponen.....	48
Gambar 4.21 Import Dashboard Grafana Kubernetes	49

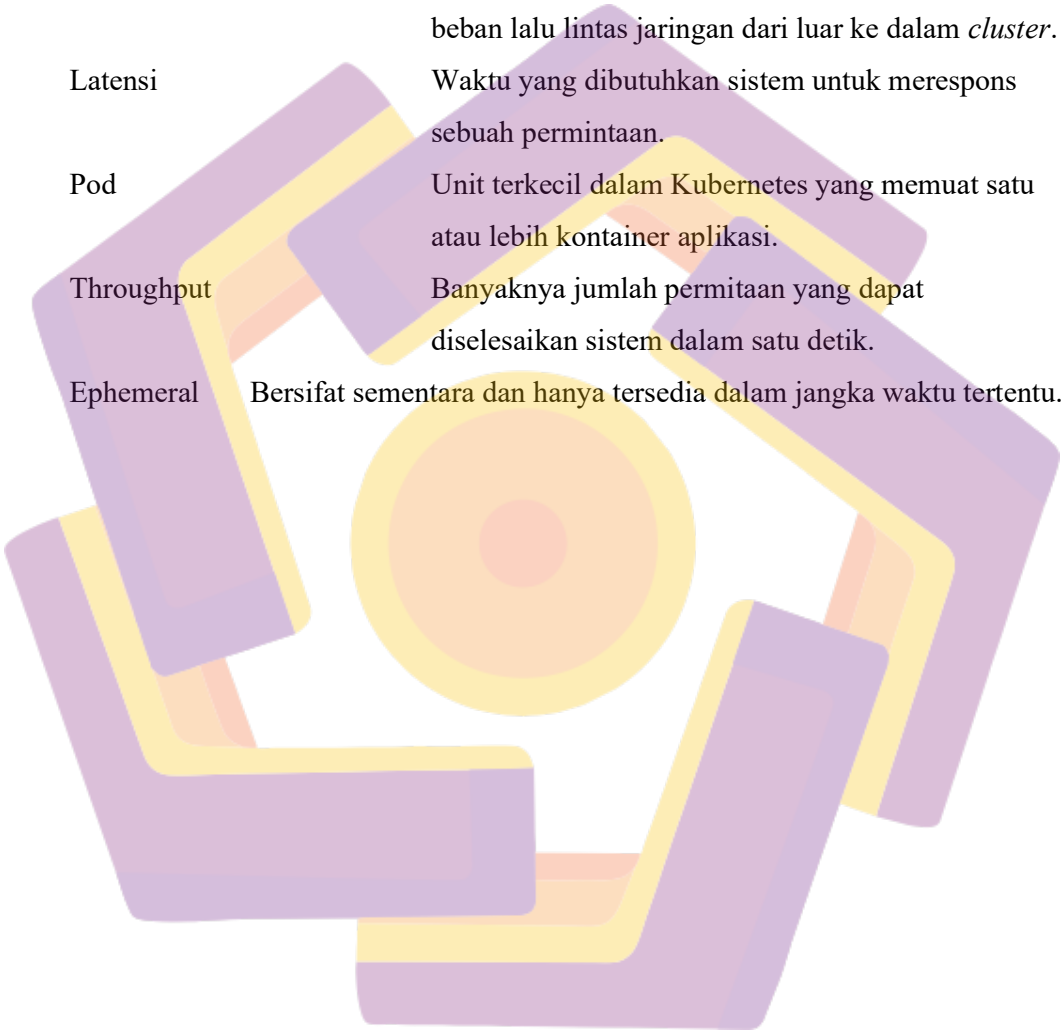
Gambar 4.22 Prometheus Sebagai Datasource	49
Gambar 4.23 Tampilan Dashboard Kubernetes Views.....	50
Gambar 4.24 Import Dashboard k6	50
Gambar 4.25 Tampilan Dashboard k6	50
Gambar 4.26 Instalasi k6 Pada Instance NGINX	51
Gambar 4.27 Instalasi k6 Pada Instance Traefik.....	51
Gambar 4.28 Mengunduh Skrip k6 NGINX.....	51
Gambar 4.29 Mengunduh Skrip k6 Traefik.....	52
Gambar 4.30 Konfigurasi Endpoint Pengujian NGINX	52
Gambar 4.31 Konfigurasi Endpoint Pengujian Traefik	52
Gambar 4.32 Eksekusi Skrip k6 NGINX Cluster	52
Gambar 4.33 Eksekusi Skrip k6 Traefik Cluster	52
Gambar 4.34 Monitoring NGINX Ingress Controller.....	53
Gambar 4.35 Monitoring Traefik Ingress Controller.....	53
Gambar 4.36 Terraform Destroy.....	54
Gambar 4.37 Hasil Terraform Destroy	54
Gambar 4.38 Grafik Latensi NGINX Ingress Controller.....	55
Gambar 4.39 Output Pengujian NGINX Ingress Controller.....	55
Gambar 4.40 Grafik Latensi Traefik Ingress Controller.....	56
Gambar 4.41 Output Pengujian Traefik Ingress Controller.....	56
Gambar 4.42 Grafik Throughput NGINX Ingress Controller	57
Gambar 4.43 Output Throughput NGINX Ingress Controller.....	57
Gambar 4.44 Grafik Throughput Traefik Ingress Controller.....	58
Gambar 4.45 Output Pengujian Traefik Ingress Controller.....	58
Gambar 4.46 Error Rate NGINX Ingress Controller	59
Gambar 4.47 Error Rate Traefik Ingress Controller	59
Gambar 4.48 Status Error Rate NGINX Ingress Controller	59
Gambar 4.49 Grafik Penggunaan CPU NGINX Ingress Controller	60
Gambar 4.50 Penggunaan CPU Traefik Ingress Controller.....	61
Gambar 4.51 Penggunaan Memori NGINX Ingress Controller	62
Gambar 4.52 Penggunaan Memori Traefik Ingress Controller.....	62

DAFTAR LAMBANG DAN SINGKATAN



AWS	Amazon Web Services
API	Application Programming Interface
AZ	Availability Zone
CLI	Command Line Interface
CPU	Central Processing Unit
EC2	Elastic Compute Cloud
EKS	Elastic Kubernetes Services
ELB	Elastic Load Balancing
GiB	Gibibyte
HCL	HashiCorp Configuration Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaC	Infrastructure as Code
MiB	Mebibyte
ms	Milidetik (Millisecond)
μ s	Mikrodetik (Microsecond)
NAT	Network Address Translation
RAM	Random Access Memory
RPS	Requests Per Second
SSL/TLS	Secure Socket Layer / Transport Layer Security
VPC	Virtual Private Cloud
VU	Virtual User

DAFTAR ISTILAH



Infrastructure as Code	Metode pengelolaan infrastruktur TI melalui kode otomatisasi, bukan konfigurasi manual.
Ingress Controller	Komponen yang mengatur akses dan distribusi beban lalu lintas jaringan dari luar ke dalam <i>cluster</i> .
Latensi	Waktu yang dibutuhkan sistem untuk merespons sebuah permintaan.
Pod	Unit terkecil dalam Kubernetes yang memuat satu atau lebih kontainer aplikasi.
Throughput	Banyaknya jumlah permintaan yang dapat diselesaikan sistem dalam satu detik.
Ephemeral	Bersifat sementara dan hanya tersedia dalam jangka waktu tertentu.

INTISARI

Kubernetes telah menjadi standar industri dalam orkestrasi kontainer, namun pemilihan *Ingress Controller* yang tepat, seperti antara NGINX dan Traefik, masih menjadi tantangan teknis yang krusial dalam perencanaan infrastruktur *cloud*. Ketidaktepatan dalam pemilihan komponen ini dapat berdampak pada degradasi performa aplikasi dan inefisiensi penggunaan sumber daya infrastruktur, khususnya pada kondisi beban lalu lintas yang fluktuatif dan tinggi. Penelitian ini bertujuan untuk melakukan analisis komparatif secara mendalam terhadap performa kedua *Ingress Controller* tersebut pada lingkungan yang terkendali. Metodologi penelitian ini menerapkan pendekatan eksperimental pada lingkungan AWS Elastic Kubernetes Service (EKS) yang dibangun menggunakan *Infrastructure as Code* (IaC) melalui Terraform untuk menjamin konsistensi, isolasi sumber daya, serta reproduisibilitas pengujian. Pengujian performa dilakukan melalui skenario *load testing* menggunakan k6 sebagai *load generator*, dengan monitoring metrik melalui Prometheus dan visualisasi data menggunakan Grafana. Parameter evaluasi difokuskan pada metrik latensi (min, avg, p50, p75, p90, p95, p99, max), *throughput*, *error rate*, serta konsumsi sumber daya CPU dan memori. Hasil penelitian menunjukkan karakteristik performa yang berbeda secara signifikan, NGINX unggul dalam hal performa latensi pada min, avg, p50, p75, p90, p95, kapasitas *throughput* yang lebih besar, serta penggunaan sumber daya CPU dan memori yang lebih efisien namun mengalami *error rate* sebesar 0,40% dari total permintaan sebesar 5.510.671. Di sisi lain, Traefik unggul pada latensi p99, dan max, meski menghasilkan kapasitas *throughput* yang jauh lebih kecil dari NGINX, konsumsi sumber daya CPU dan memori yang lebih intensif, namun dapat mempertahankan keandalan sistem dengan angka *error rate* nol persen. Temuan ini memberikan kontribusi praktis dalam menentukan *Ingress Controller* yang optimal bagi kebutuhan sistem.

Kata kunci: Kubernetes, Load Testing, Ingress Controller, NGINX, Traefik

ABSTRACT

Kubernetes has become the industry standard for container orchestration, but choosing the right Ingress Controller, such as NGINX or Traefik, remains a crucial technical challenge in cloud infrastructure planning. Incorrect selection can lead to application performance degradation and inefficient resource usage, especially under high and fluctuating traffic loads. This research aims to perform a deep comparative analysis of these two Ingress Controllers in a controlled environment. The methodology applies an experimental approach on AWS Elastic Kubernetes Service (EKS) built using Infrastructure as Code (IaC) with Terraform to ensure consistency, resource isolation, and test reproducibility. Performance testing was conducted through load testing scenarios using k6 as the load generator, with metric monitoring via Prometheus and data visualization using Grafana. Evaluation parameters focused on latency metrics (min, avg, p50, p75, p90, p95, p99, max), throughput, error rate, and CPU and memory resource consumption. The results show significantly different performance characteristics where NGINX excels in latency performance across min, avg, p50, p75, p90, and p95 metrics, provides larger throughput capacity, and more efficient resource usage, but experienced an error rate of 0.40% from a total of 5,510,671 requests. On the other hand, Traefik excels in p99 and max latency, and although it produces much lower throughput and more intensive resource consumption, it maintains system reliability with a zero percent error rate. These findings provide a practical contribution to determining the optimal Ingress Controller based on specific system requirements.

Keyword: Kubernetes, Load Testing, Ingress Controller, NGINX, Traefik.