

TUGAS AKHIR

**KOMPARASI PERFORMA, BIAYA, DAN SKALABILITAS CLOUD
RUN DAN APP ENGINE PADA BACKEND REST API**



DIAJUKAN OLEH:

FADILLA MA'SUM HAQQONI

NIM. 22106050039

**STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA**

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2026

HALAMAN PENGESAHAN



**KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-833/Un.02/DST/PP.00.9/05/2026

Tugas Akhir dengan judul : Komparasi Performa, Biaya, dan Skalabilitas Cloud Run dan App Engine pada Backend REST API

yang dipersiapkan dan disusun oleh:

Nama : FADILLA MA'SUM HAQQONI
Nomor Induk Mahasiswa : 22106050039
Telah diujikan pada : Rabu, 22 April 2026
Nilai ujian Tugas Akhir : A

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR



Ketua Sidang

Ir. Muhammad Taufiq Nuruzzaman, S.T. M.Eng., Ph.D.
SIGNED

Valid ID: 69fd5f0bbe44



Penguji I

Dr. Ir. Bambang Sugiantoro, M.T., IPU.,
ASEAN Eng.
SIGNED

Valid ID: 69facc95aa1c1



Penguji II

Eko Hadi Gunawan, M.Eng.
SIGNED

Valid ID: 69fc2e4571d16



Yogyakarta, 22 April 2026
UIN Sunan Kalijaga

Dekan Fakultas Sains dan Teknologi

Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

Valid ID: 6a013cbd7a23b

LEMBAR KEASLIAN TUGAS AKHIR

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Fadilla Ma'sum Haqqoni
NIM : 22106050039
Program Studi : Informatika
Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya, bahwa skripsi saya yang berjudul "**Komparasi Performa, Biaya, dan Skalabilitas Cloud Run dan App Engine pada Backend REST API**" merupakan penelitian saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan bukan plagiasi karya orang lain kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Sebagai salah satu syarat untuk memperoleh gelar Sarjana Program Studi Informatika pada Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga.

Yogyakarta, 6 April 2026
Yang membuat pernyataan,



Fadilla Ma'sum Haqqoni
22106050039

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

LEMBAR PERSETUJUAN SKRIPSI/TUGAS AKHIR



Universitas Islam Negeri Sunan Kalijaga



FM-UINSK-BM-05-03/R0

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir

Lamp : -

Kepada
Yth.
Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga
DI Yogyakarta

Assalamu 'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka saya selaku pembimbing berpendapat bahwa skripsi Saudari:

Nama : Fadilla Ma'sum Haqqoni
NIM : 22106050039
Judul Skripsi : Komparasi Performa, Biaya, dan Skalabilitas Cloud Run dan App Engine pada Backend REST API

Sudah dapat diajukan kepada Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Informatika.

Dengan ini kami berharap agar skripsi / tugas akhir Saudari dapat segera dimunaqsyahkan. Atas perhatiannya saya ucapkan terima kasih.

Wassalamu 'alaikum wr. wb.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Yogyakarta, 01 April 2026

Pembimbing,

Ir. Muhammad Taufiq Nuruzzaman, S.T. M.Eng., Ph.D.
NIP. 19791118 200501 1 003

LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR

Tugas akhir ini tidak dipublikasikan secara umum, namun dapat diakses di perpustakaan dalam lingkungan UIN Sunan Kalijaga Yogyakarta. Dokumen ini diperbolehkan untuk dijadikan referensi kepustakaan, dengan ketentuan bahwa setiap pengutipan harus mendapatkan izin dari penulis serta mencantumkan sumber sesuai kaidah penulisan ilmiah. Hak kepemilikan dokumen tugas akhir ini berada pada UIN Sunan Kalijaga Yogyakarta.



INSTISARI

Adopsi *serverless* Cloud Run dan App Engine belum didukung data empiris *head-to-head* yang memadai di ekosistem industri. Penelitian ini bertujuan mengevaluasi kinerja, efisiensi biaya, dan skalabilitas kedua platform dalam menjalankan *backend* RESTful API guna memberikan landasan keputusan arsitektur awan yang objektif. Eksperimen kuantitatif dilakukan pada aplikasi "Notes App" berbasis Go menggunakan instrumen K6 melalui skenario *average load* (150 VUs) dan *stress test* (3.000 VUs) dengan 10 kali replikasi. Hasil eksperimen menunjukkan bahwa pada beban moderat, performa kedua platform setara, namun App Engine terbukti lebih ekonomis (Rp1,45 per 1.000 permintaan sukses) berkat siklus pembaruan kuota gratis harian. Sebaliknya, saat *stress test*, terjadi fenomena *cost reversal* di mana Cloud Run lebih unggul dengan *p95 latency* 325,59 ms (berbanding 1.646,15 ms pada App Engine) dan *error rate* 0,0027%. Keunggulan ini didukung oleh skalabilitas instans Cloud Run yang sangat instan, sementara lambatnya *autoscaling* App Engine memicu antrean permintaan yang menyebabkan saturasi pada utilitas CPU. Selain itu, Cloud Run terbukti lebih efisien pada beban tinggi (Rp6,88 per 1.000 permintaan sukses) berkat skema penagihan per detik yang meniadakan biaya instans *idle*. Kesimpulannya, Cloud Run optimal untuk trafik fluktuatif berskala besar, sedangkan App Engine cocok untuk beban harian yang moderat.

Kata Kunci: *Serverless Computing, Google Cloud Run, Google App Engine, Performance Testing, Go, RESTful API, Efisiensi Biaya.*

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

ABSTRACT

The adoption of serverless Cloud Run and App Engine lacks sufficient head-to-head empirical data in the industry. This study evaluates the performance, cost efficiency, and scalability of both platforms running a RESTful API backend to provide an objective foundation for cloud architecture decisions. Quantitative experiments were conducted on a Go-based "Notes App" using K6 for average load (150 VUs) and stress test (3,000 VUs) scenarios with 10 replications. Results indicate that under moderate loads, both platforms perform equally, but App Engine is more economical (Rp1.45/1,000 requests) due to its daily free quota. Conversely, during stress testing, a cost reversal occurred: Cloud Run outperformed with a p95 latency of 325.59 ms (versus 1,646.15 ms for App Engine) and a 0.0027% error rate. This superiority is driven by Cloud Run's instant instance scalability, whereas App Engine's slow autoscaling triggered request queues leading to CPU utility saturation. Furthermore, Cloud Run proved more cost-efficient under high loads (Rp6.88/1,000 requests) due to its per-second billing scheme eliminating idle instance costs. In conclusion, Cloud Run is optimal for large-scale, fluctuating traffic, while App Engine is ideal for moderate daily loads.

Keywords: *Serverless Computing, Google Cloud Run, Google App Engine, Performance Testing, Go, RESTful API, Cost Efficiency.*

HALAMAN MOTTO

Yang harus dihadapi, tetap harus dihadapi.

~ Shi Hao

Bagaimanapun orang melihatmu, kau hanya perlu ingat, kau tidak hidup untuk orang lain.

Kau hidup untuk dirimu sendiri.

~ Xiao Yan

Memberi ikan kepada seseorang tidak sebaik mengajarnya cara menangkap ikan

~ Li Baoping



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

HALAMAN PERSEMBAHAN

Puji syukur ke hadirat Allah SWT, Sang Pemilik segala ilmu, yang telah memberikan kekuatan dan ketabahan hingga langkah terakhir dalam penyusunan skripsi ini dapat terselesaikan. Karya ini, yang dibangun di atas dedikasi setiap fase eksperimen dan ketelitian dalam sepuluh kali replikasi pengujian, saya persembahkan dengan penuh cinta untuk:

1. Kedua Orang Tua Tercinta, yang doa-doanya mengalir lebih deras daripada throughput jutaan requests. Terima kasih atas dukungan moral dan materi yang menjadi bahan bakar utama dalam setiap fase ramp-up kehidupan saya.
2. Dosen Pembimbing, yang telah memberikan arahan berharga serta dukungan penuh dalam menjaga kelancaran proses akademik penelitian ini hingga tahap penyelesaian.
3. Diri Saya Sendiri, Fadilla Ma'sum Haqqoni. Terima kasih telah bersabar dan tekun dalam menyelesaikan seluruh proses pengumpulan data melalui 10 kali replikasi untuk setiap platform dan setiap skenario pengujian. Kamu telah membuktikan bahwa jawaban yang paling masuk akal hanya bisa ditemukan melalui ketelitian dan dedikasi dalam mencari bukti empiris yang nyata.



KATA PENGANTAR

Segala puji bagi Allah SWT, Tuhan Semesta Alam, yang telah melimpahkan rahmat, hidayah, dan kekuatan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Analisis Performa dan Biaya Layanan Serverless Google Cloud Run dan Google App Engine pada Arsitektur Backend RESTful API". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga Yogyakarta.

Penulis menyadari bahwa proses penyelesaian karya ini bukanlah perjalanan yang singkat. Dibutuhkan ketekunan dalam setiap tahap eksperimen untuk memastikan setiap data yang diambil memiliki landasan yang kuat. Oleh karena itu, dengan kerendahan hati, penulis menyampaikan ucapan terima kasih yang tulus kepada:

1. Prof. Noorhaidi Hasan S.Ag., M.A., M.Phil., Ph.D. selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Prof. Dr. Dra. Hj. Khurul Wardati, M.Si. selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Bapak Dr. Muhammad Mustakim, S.T. M.T. selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Dr. Agus Mulyanto, S.Si., M.Kom., ASEAN Eng. selaku dosen penasihat akademik yang telah memberikan arahan dan bimbingan selama masa perkuliahan penulis.
5. Ir. Muhammad Taufiq Nuruzzaman, S.T. M.Eng., Ph.D. selaku dosen pembimbing yang telah memberikan bimbingan, masukan yang berharga, serta dukungan penuh dengan penuh kesabaran dalam setiap tahapan penyusunan Tugas Akhir ini.
6. Seluruh dosen dan karyawan Program Studi Informatika UIN Sunan Kalijaga Yogyakarta yang telah memberikan ilmu, bantuan, serta pelayanan selama penulis menempuh pendidikan.
7. Kedua orang tua yang senantiasa memberikan doa, kasih sayang, dukungan, serta pengorbanan yang tiada henti demi keberhasilan penulis.
8. Sahabat serta teman-teman seperjuangan yang telah memberikan dukungan, bantuan, serta kebersamaan selama proses perkuliahan hingga penyusunan Tugas Akhir ini.
9. Semua pihak yang tidak dapat disebutkan satu per satu yang telah memberikan bantuan, dukungan, dan doa dalam penyelesaian Tugas Akhir ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan di masa mendatang. Akhir kata, penulis berharap karya ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan, khususnya di bidang teknologi komputasi awan (*Cloud Computing*) dan efisiensi biaya operasional TI (*Cloud FinOps*).

Yogyakarta, 05 April 2026

Penyusun,



Fadilla Ma'sum Haqqoni

22106050039



DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN PENGESAHAN	ii
LEMBAR KEASLIAN TUGAS AKHIR.....	iii
LEMBAR PERSETUJUAN SKRIPSI/TUGAS AKHIR	iv
LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR	v
INSTISARI.....	vi
<i>ABSTRACT</i>	vii
HALAMAN MOTTO.....	viii
HALAMAN PERSEMBAHAN.....	ix
KATA PENGANTAR	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	7
2.1. Tinjauan Pustaka.....	7
2.2. Landasan Teori.....	10
2.2.1. Cloud Computing.....	11
2.2.2. Model Layanan <i>Cloud</i> (IaaS, PaaS, SaaS).....	11

2.2.3.	<i>Serverless Computing</i>	12
2.2.4.	Google Cloud Platform (GCP).....	12
2.2.5.	Google Cloud Run	13
2.2.6.	Google App Engine.....	14
2.2.7.	Kontainerisasi dan Docker	14
2.2.8.	Bahasa Pemrograman Go (Golang)	15
2.2.9.	Gin Web Framework.....	15
2.2.10.	GORM (Go Object Relational Mapping)	15
2.2.11.	Cloud SQL	15
2.2.12.	RESTful API.....	16
2.2.13.	<i>Performance Testing</i>	17
2.2.14.	Metrik Evaluasi Kinerja Sistem	17
2.2.15.	<i>Load Testing</i> dengan K6	19
2.2.16.	Cloud Monitoring.....	19
2.2.17.	Efisiensi Biaya dan Analisis Cloud Billing Reports	20
BAB III METODE PENELITIAN		21
3.1.	Alat dan Bahan Penelitian.....	21
3.1.1.	Perangkat Keras (<i>Hardware</i>)	21
3.1.2.	Perangkat Lunak (<i>Software</i>)	21
3.1.3.	Bahan Penelitian	24
3.2.	Metode Penelitian	25
3.2.1.	Arsitektur Lingkungan Eksperimen.....	26
3.2.2.	Konfigurasi Lingkungan <i>Cloud (Deployment)</i>	28
3.2.3.	Skenario Pengujian Kinerja (<i>Load Testing</i>).....	29
3.2.4.	Prosedur Analisis Data.....	32
3.2.5.	Tahapan Penelitian.....	33
BAB IV HASIL DAN PEMBAHASAN		38

4.1.	Gambaran Umum Hasil Eksperimen	38
4.2.	Analisis Kinerja	39
4.2.1.	Analisis <i>p95 Latency</i>	39
4.2.2.	Analisis <i>Throughput (Requests Per Second / RPS)</i>	43
4.2.3.	Analisis Error Rate.....	48
4.2.4.	Analisis <i>Resource Utilization (CPU dan Memory)</i>	53
4.2.5.	Sintesis Kinerja dan Efisiensi	59
4.3.	Analisis Skalabilitas.....	61
4.3.1.	Skenario <i>Average Load (150 Virtual Users)</i>	62
4.3.2.	Skenario <i>Stress Test (3.000 Virtual Users)</i>	63
4.3.3.	Interpretasi Komparatif Skalabilitas	64
4.4.	Analisis Efisiensi Biaya	65
4.4.1.	Skenario <i>Average Load (150 Virtual Users)</i>	66
4.4.2.	Skenario <i>Stress Test (3.000 Virtual Users)</i>	67
4.4.3.	Ringkasan Komparatif Biaya.....	69
4.5.	Sintesis Hasil Pengujian dan Evaluasi Komparatif.....	70
4.5.1.	Ringkasan Komparatif Antar Platform	70
4.5.2.	Analisis Sintesis Lintas Dimensi	72
4.5.3.	Keterkaitan dengan Rumusan Masalah.....	74
4.5.4.	Batasan Sintesis	75
BAB V PENUTUP		75
5.1.	Kesimpulan	75
5.2.	Saran	76
DAFTAR PUSTAKA		78
LAMPIRAN.....		Error! Bookmark not defined.

DAFTAR GAMBAR

Gambar 3.1 Arsitektur Lingkungan Eskperimen	28
Gambar 4.1 Grafik Perbandingan p95 Latency Skenario Average Load Test	41
Gambar 4.2 Grafik Perbandingan p95 Latency Skenario Stress Test.....	42
Gambar 4.3 Grafik Perbandingan Throughput Skenario Average Load Test	45
Gambar 4.4 Grafik Perbandingan Throughput Skenario Stress Test\.....	46
Gambar 4.5 Grafik Perbandingan CPU Utilizaiton Skenario Average Load Test	54
Gambar 4.6 Grafik Perbandingan Memory Utilization Skenario Average Load Test	55
Gambar 4.7 Grafik Perbandingan CPU Utilization Skenario Stress Test.....	57
Gambar 4.8 Grafik Perbandingan Memory Utilization Skenario Stress Test.....	58
Gambar 4.9 Grafik Perbandingan Skalabilitas Skenario Average Load Test.....	62
Gambar 4.10 Grafik Perbandingan Skalabilitas Skenario Stress Test.....	63

DAFTAR TABEL

Tabel 2.1 Ringkasan Tinjauan Pustaka.....	9
Tabel 4.1 Ringkasan Rata-Rata p95 Latency Skenario Average Load Test.....	40
Tabel 4.2 Ringkasan Rata-Rata p95 Latency Skenario Stress Test.....	42
Tabel 4.3 Ringkasan Rata-Rata Throughput Skenario Average Load Test.....	44
Tabel 4.4 Ringkasan Rata-Rata Throughput Skenario Stress Test.....	46
Tabel 4.5 Ringkasan Error Rate Skenario Average Load Test.....	48
Tabel 4.6 Waktu Terjadinya Error pada Skenario Average Load	49
Tabel 4.7 Ringkasan Rata-Rata Error Rate Skenario Stress Test.....	50
Tabel 4.8 Waktu Kemunculan Error pada Skenario Stress Test.....	51
Tabel 4.9 Ringkasan Rata-Rata CPU Utilizaiton Skenario Average Load Test.....	54
Tabel 4.10 Ringkasan Rata-Rata Memory Utilization Skenario Average Load Test.....	56
Tabel 4.11 Ringkasan Rata-Rata CPU Utilization Skenario Stress Test.....	57
Tabel 4.12 Ringkasan Memory Utilization Skenario Stress Test.....	58
Tabel 4.13 Ringkasan Rata-Rata Skalabilitas Skenario Average Load Test.....	62
Tabel 4.14 Ringkasan Rata-Rata Skalabilitas Skenario Stress Test	63
Tabel 4.15 Rincian Biaya Skenario Average Load Test.....	66
Tabel 4.16 Rincian Biaya Skenario Stress Test.....	68
Tabel 4.17 Ringkasan Komparatif Hasil Pengujian.....	71



DAFTAR LAMPIRAN

Lampiran 1 Ringkasan Tugas Akhir	Error! Bookmark not defined.
Lampiran 2 Daftar Singkatan.....	Error! Bookmark not defined.
Lampiran 3 Instrumen Penelitian.....	Error! Bookmark not defined.
Lampiran 4 Hasil Penelitian Skenario A (Output CLI Skenario A)	Error! Bookmark not defined.
Lampiran 5 Hasil Penelitian Skenario B (Output CLI)	Error! Bookmark not defined.



BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi *cloud computing* telah mendorong transformasi besar dalam cara perusahaan dan pengembang mengelola sumber daya komputasi. Model ini memungkinkan pengguna untuk menggunakan infrastruktur, platform, atau perangkat lunak melalui jaringan internet tanpa harus mengelola secara langsung sumber daya fisik yang digunakan [1]. Dengan karakteristik elastisitas dan efisiensi biaya, pendekatan ini menjadi fondasi utama dalam pengembangan sistem berbasis *cloud* modern.

Salah satu evolusi dari paradigma tersebut adalah *serverless computing*, yaitu pendekatan komputasi yang memungkinkan pengembang membangun aplikasi tanpa perlu mengelola server secara manual serta menggunakan model pembayaran berbasis penggunaan aktual (*pay-per-use*) [2]. Menurut Hassan et al. [3], *serverless computing* menawarkan efisiensi tinggi melalui mekanisme *autoscaling* otomatis, biaya operasional berbasis konsumsi, serta kemudahan integrasi dengan layanan lain. Namun demikian, tantangan seperti fenomena *cold start* dan keterbatasan kontrol terhadap lingkungan eksekusi masih menjadi fokus penelitian yang terus berkembang [4]. Meskipun *cold start* merupakan salah satu karakteristik penting dalam arsitektur *serverless*, penelitian ini tidak berfokus pada pengukuran fenomena tersebut secara khusus, melainkan pada evaluasi kinerja umum seperti *latency*, *throughput*, *error rate*, biaya, serta respons *autoscaling* pada skenario beban terkontrol.

Google Cloud Platform (GCP) merupakan salah satu penyedia layanan utama di kalangan pengembang karena menyediakan beragam layanan *serverless*, termasuk Google Cloud Run dan Google App Engine, yang memungkinkan pengembang men-*deploy* aplikasi tanpa perlu mengelola server secara langsung. App Engine adalah layanan *Platform-as-a-Service (PaaS)* dengan otomatisasi penuh terhadap konfigurasi dan *scaling* [5], sedangkan Cloud Run berbasis kontainer yang menawarkan kendali lingkungan yang lebih luas karena mendukung berbagai *runtime environments* [6].

Laporan industri juga menunjukkan adanya peningkatan penggunaan layanan *serverless*. Menurut *State of Serverless Report 2023* yang dirilis oleh Datadog, sekitar 60% pengguna Google Cloud telah menggunakan layanan *serverless*, naik sekitar 7% dibandingkan tahun sebelumnya [7]. Peningkatan ini menunjukkan bahwa layanan seperti Cloud Run dan App Engine menjadi pilihan utama dalam pengembangan aplikasi modern karena dinilai lebih efisien dibandingkan model komputasi tradisional.

Berdasarkan penelusuran literatur awal, sebagian besar penelitian yang tersedia saat ini cenderung berfokus pada perbandingan fitur secara kualitatif [8], [9] atau membandingkan layanan *serverless* lintas penyedia *cloud*, misalnya GCP dibandingkan dengan AWS [10]. Evaluasi kuantitatif yang membandingkan efisiensi internal di dalam ekosistem Google Cloud, khususnya antara Cloud Run dan App Engine Standard pada beban kerja yang identik, masih terbatas. Keterbatasan data komparatif head-to-head ini menunjukkan masih adanya celah penelitian dalam menyediakan dasar empiris yang kuat untuk mendukung pengambilan keputusan arsitektur. Padahal, literatur menunjukkan bahwa pemilihan lingkungan *serverless* yang tepat merupakan salah satu tantangan krusial bagi praktisi dalam fase perancangan sistem [11].

Tanpa adanya metrik evaluasi yang spesifik, proses penentuan platform rentan berdampak langsung terhadap efisiensi operasional dan kualitas layanan. Karakteristik arsitektur platform yang berbeda terbukti menghasilkan perbedaan signifikan pada metrik kinerja seperti *p95 latency*, *throughput*, dan *error rate* ketika sistem menghadapi beban yang tinggi [12], [13]. Sebaliknya, pemilihan platform yang kurang efisien dari sisi biaya dapat meningkatkan *cost per successful request* tanpa peningkatan kinerja yang sebanding [11]. Oleh karena itu, diperlukan analisis empiris *head-to-head* yang mengukur kinerja, respons *autoscaling*, dan biaya berbasis *list price* pada periode pengujian yang identik guna membantu pengambilan keputusan arsitektur secara objektif dan terukur

Berangkat dari kondisi tersebut, penelitian ini bertujuan untuk melakukan evaluasi komparatif terhadap kinerja, efisiensi biaya, serta kemampuan skalabilitas Google Cloud Run dan Google App Engine Standard pada arsitektur *backend* RESTful API. Temuan dari penelitian ini diproyeksikan mampu menyajikan basis data empiris guna mendukung pengambilan keputusan arsitektur yang objektif, sehingga praktisi dapat memilih layanan yang optimal dalam menjaga stabilitas dan efisiensi operasional sistem

1.2. Rumusan Masalah

Perbedaan performa dan efisiensi biaya pada layanan *serverless* Google Cloud Run serta App Engine Standard melahirkan kebutuhan akan pengujian komparatif yang objektif dan terukur. Berdasarkan hal tersebut, rumusan masalah dalam penelitian ini didefinisikan sebagai berikut:

1. Bagaimana perbandingan kinerja (*performance*) antara Google Cloud Run dan Google App Engine Standard diukur dari *p95 latency*, *throughput*, dan *error rate* dalam

menjalankan arsitektur *backend RESTful API* pada kondisi beban moderat (*average load*) maupun beban ekstrem (*stress test*)?

2. Bagaimana perbandingan efisiensi biaya operasional (*cost efficiency*) antara kedua layanan berdasarkan metrik pengeluaran per permintaan sukses (*cost per successful requests*) ketika dihadapkan pada eskalasi beban trafik yang berbeda?
3. Bagaimana perbandingan dinamika karakteristik skalabilitas (*autoscaling*) dari masing-masing platform dalam merespons dan mengalokasikan jumlah *instance* komputasi, baik pada saat mempertahankan kestabilan lalu lintas data harian maupun saat memitigasi lonjakan beban ekstrem?

1.3. Batasan Masalah

Mengingat luasnya cakupan parameter teknis dalam evaluasi layanan *cloud computing*, ruang lingkup pengujian perlu dibatasi agar penelitian tetap fokus, terukur, dan efektif. Oleh karena itu, batasan masalah dalam penelitian ini ditetapkan sebagai berikut:

1. Fokus Layanan dan Arsitektur: Evaluasi komparatif hanya dilakukan pada dua layanan komputasi *serverless* dari Google Cloud Platform (GCP), yaitu Google Cloud Run dan Google App Engine Standard. Objek pengujian dibatasi pada arsitektur *backend RESTful API* sederhana yang tidak menerapkan autentikasi, mekanisme *caching*, maupun pola *microservices*, guna mengisolasi performa murni dari infrastruktur tanpa adanya bias kompleksitas kode. Platform dari penyedia *cloud* lain (seperti AWS atau Azure) tidak diteliti.
2. Standardisasi Lingkungan Pengujian: Seluruh infrastruktur komputasi dan basis data (*Cloud SQL*) ditempatkan pada region yang sama (*asia-southeast2*). Mesin penguji (*load generator*) juga dieksekusi dari *Virtual Machine* di region yang sama untuk meminimalisasi fluktuasi latensi jaringan internet publik. Konfigurasi basis data dan parameter koneksi diatur secara konstan pada kedua platform agar hasil pengujian murni mencerminkan performa komputasi, bukan *bottleneck* pada basis data.
3. Konfigurasi Skalabilitas: Parameter dasar *autoscaling* pada kedua platform dikunci pada batas yang setara (*minimum instance* = 1, *maksimum instance* = 10, dan *maksimum concurrent requests* = 10). Analisis skalabilitas hanya difokuskan pada kelincahan mitigasi (*scale-out* dan *scale-in*) terhadap hantaman beban, tanpa membahas optimasi algoritma *autoscaler* tingkat lanjut maupun konsep penambahan kapasitas perangkat keras (*vertical scaling*).

4. Ruang Lingkup Efisiensi Biaya: Perbandingan biaya hanya memperhitungkan komponen komputasi utama yang ditagihkan (*list price*) beserta pengaruh dari skema kuota gratis permanen bawaan platform (*Always Free Tier*). Subsidi berupa kredit promosi akun baru (*Free Trial \$300*) tidak diikutsertakan. Selain itu, biaya komponen pendukung seperti basis data tidak dimasukkan dalam komparasi karena diperlakukan sebagai variabel kontrol yang nilainya diasumsikan sama.
5. Pengecualian Metrik Tambahan: Penelitian ini berfokus secara eksklusif pada metrik performa saat sistem berjalan stabil (*steady-state*) dan saat menghadapi beban ekstrem. Aspek di luar itu seperti latensi pemanasan instans (*cold start latency*), metrik keamanan (*security*), keandalan jangka panjang (*SLA/uptime*), serta efisiensi pengalaman pengembang (*Developer Experience/CI-CD pipeline*) berada di luar ruang lingkup penelitian ini.

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah untuk menjawab seluruh rumusan masalah melalui rangkaian eksperimen dan analisis metrik yang terukur secara objektif. Berdasarkan hal tersebut, sasaran spesifik yang ingin dicapai dalam studi ini dijabarkan sebagai berikut:

1. Menganalisis dan membandingkan kinerja komputasi (*performance*) antara Google Cloud Run dan Google App Engine Standard melalui pengukuran *latency*, *throughput*, dan *error rate* dalam menjalankan arsitektur *backend RESTful API* pada kondisi beban moderat (*average load*) maupun beban ekstrem (*stress test*)
2. Mengevaluasi dan membandingkan tingkat efisiensi biaya operasional (*cost efficiency*) komputasi dari kedua platform layanan berdasarkan perhitungan pengeluaran per permintaan sukses (*cost per successful requests*) ketika dihadapkan pada eskalasi beban trafik yang berbeda.
3. Menganalisis dan membandingkan dinamika karakteristik skalabilitas (*autoscaling*) dari masing-masing layanan awan dalam merespons dan mengalokasikan jumlah *instance* komputasi, baik pada saat mempertahankan kestabilan lalu lintas data harian maupun saat memitigasi lonjakan beban secara ekstrem.

1.5. Manfaat Penelitian

Kajian komparatif infrastruktur *cloud* ini dirancang untuk menghasilkan temuan empiris yang bernilai secara akademik sekaligus dapat diaplikasikan langsung oleh praktisi sebagai dasar pengambilan keputusan arsitektur. Oleh karena itu, kontribusi dan manfaat signifikan dari penelitian ini diklasifikasikan ke dalam dua aspek utama sebagai berikut:

1. Manfaat Akademis

- a. Memberikan kontribusi ilmiah terhadap pengembangan kajian ilmu di bidang rekayasa perangkat lunak dan komputasi awan (*cloud computing*), khususnya pada analisis komparatif layanan *serverless computing*. Hal ini diharapkan dapat memperkaya khazanah keilmuan mengenai optimasi infrastruktur pada ekosistem Google Cloud Platform (GCP) yang masih terus berkembang secara dinamis.
- b. Menjadi referensi akademik bagi mahasiswa maupun peneliti lain yang ingin melakukan penelitian berbasis *system benchmarking* pada layanan *Platform-as-a-Service* (PaaS). Dokumentasi prosedur pengujian kuantitatif-eksperimental dalam penelitian ini dapat dijadikan panduan teknis bagi studi sejenis yang melibatkan arsitektur berbasis *container* di masa mendatang.
- c. Menyediakan landasan awal untuk penelitian lanjutan yang ingin mengeksplorasi metrik di luar fase *runtime*, seperti ketahanan keamanan (*security*) maupun fenomena *cold start latency*. Dengan adanya data awal ini, peneliti selanjutnya dapat melakukan eksplorasi lebih mendalam terhadap variabel kritis lainnya seperti integrasi *CI/CD pipeline* pada lingkungan *serverless*.

2. Manfaat Praktis

- a. Memberikan wawasan berbasis data (*data-driven insight*) bagi pengembang perangkat lunak dan praktisi teknologi informasi dalam memilih layanan *serverless* yang paling tepat sasaran. Wawasan ini sangat krusial untuk meminimalisasi risiko kesalahan pemilihan arsitektur yang dapat berdampak pada degradasi performa aplikasi saat diakses oleh pengguna secara masif.
- b. Memberikan panduan teknis mengenai efisiensi anggaran bagi organisasi maupun *startup* dalam mengelola biaya operasional aplikasi awan. Implementasi strategi *Cloud FinOps* yang tepat berdasarkan hasil penelitian ini akan membantu institusi dalam mencapai keseimbangan yang proporsional antara tingginya performa layanan dan efisiensi pengeluaran.
- c. Menjadi landasan empiris (*baseline*) bagi pengembang yang berencana membangun arsitektur aplikasi yang lebih kompleks, seperti sistem dengan mekanisme *microservices* atau *caching* berlapis. Melalui pemahaman terhadap batasan kemampuan dasar infrastruktur, pengembang dapat merancang strategi skalabilitas yang lebih matang dan terukur untuk kebutuhan masa depan.

BAB V PENUTUP

5.1. Kesimpulan

Berdasarkan hasil eksperimen dan analisis komparatif komprehensif yang telah dilakukan terhadap layanan Google Cloud Run dan Google App Engine Standard dalam menjalankan arsitektur *backend RESTful API*, penelitian ini menghasilkan beberapa kesimpulan utama. Kesimpulan tersebut secara langsung menjawab rumusan masalah terkait performa, efisiensi biaya, dan karakteristik skalabilitas pada dua kondisi beban kerja yang berbeda, yaitu sebagai berikut:

A. Kinerja (*Performance*)

Pada kondisi beban moderat (*average load*), kedua platform menunjukkan tingkat performa yang relatif setara. Waktu respons (*latency p95*) berada di kisaran 19–20 milidetik, tingkat penyelesaian permintaan (*throughput*) berjalan stabil di angka 133 RPS, dan tingkat kegagalan (*error rate*) mendekati nol persen. Namun, pada kondisi beban ekstrem (*stress test*), Cloud Run terbukti jauh lebih tangguh dan stabil dibandingkan App Engine. Cloud Run mampu mempertahankan *latency p95* di angka 325,59 ms dan *throughput* sebesar 2.431,16 RPS dengan *error rate* yang sangat minim (0,0027%). Sebaliknya, App Engine mengalami kewalahan sistem yang ditandai dengan lonjakan *latency* hingga 1.646,15 ms, penurunan *throughput* ke angka 2.045,35 RPS, dan kegagalan pemrosesan yang cukup masif dengan *error rate* menyentuh 0,8770%.

B. Efisiensi Biaya (*Cost Efficiency*)

Evaluasi efisiensi biaya komputasi memperlihatkan adanya fenomena pembalikan keunggulan (*cost reversal*) yang sangat bergantung pada karakteristik eskalasi beban trafik. Pada skenario *average load*, App Engine terbukti jauh lebih ekonomis dengan mencatatkan biaya hanya sebesar Rp1,45 per 1.000 permintaan sukses, melampaui Cloud Run yang membutuhkan biaya Rp10,19. Keunggulan App Engine pada fase ini sangat dipengaruhi oleh siklus pembaruan kuota gratis (*free tier*) harian. Sebaliknya, pada skenario *stress test*, Cloud Run berbalik menjadi platform yang paling efisien dengan mencatatkan pengeluaran sebesar Rp6,88 per 1.000 permintaan sukses, dibandingkan App Engine yang membengkak hingga Rp17,30. Efisiensi tingkat tinggi pada Cloud Run ini bersumber dari skema penagihan presisi berbasis durasi pemrosesan (*per-second request-based billing*), yang sukses mencegah pemborosan biaya akibat instans yang menganggur selama fase pendinginan (*cooldown*).

C. Skalabilitas (*Scalability*)

Kedua platform layanan *cloud* tersebut menunjukkan strategi dan kelincahan *autoscaling* yang sangat bertolak belakang. Pada skenario beban moderat, App Engine menampilkan tingkat efisiensi sumber daya yang luar biasa dengan hanya mengandalkan satu *instance* aktif untuk menyerap seluruh beban, sementara Cloud Run harus mendistribusikan bebannya ke rata-rata dua *instance*. Akan tetapi, pada skenario beban ekstrem, Cloud Run membuktikan kelincahan ekspansi (*scale-out*) yang jauh lebih mumpuni. Cloud Run secara agresif mendistribusikan beban ke rata-rata sembilan *instance* untuk mencegah antrean pemrosesan, sedangkan App Engine tertahan pada rata-rata delapan *instance*. Lambatnya respons penambahan instans pada App Engine tersebut memicu penumpukan permintaan yang menjadi penyebab utama terjadinya degradasi performa pada sistem.

5.2. Saran

Penelitian ini merumuskan beberapa rekomendasi strategis yang ditujukan bagi para praktisi pengembang perangkat lunak serta memberikan arahan bagi akademisi untuk pengembangan penelitian di masa mendatang. Saran-saran tersebut dijabarkan sebagai berikut:

A. Rekomendasi Praktis bagi Pengembang Perangkat Lunak

Pemilihan arsitektur komputasi *serverless* wajib diselaraskan dengan proyeksi karakteristik lalu lintas data dari aplikasi yang akan dibangun. Pengembang sangat disarankan untuk mengimplementasikan Google App Engine Standard pada sistem aplikasi dengan pola trafik harian yang stabil, dapat diprediksi, dan berskala moderat. Keputusan ini akan memaksimalkan efisiensi instans tunggal dan mengoptimalkan perlindungan kuota gratis (*free tier*). Sebaliknya, untuk aplikasi berskala produksi yang memiliki potensi lonjakan trafik ekstrem, fluktuatif, dan sangat dinamis, Google Cloud Run merupakan pilihan infrastruktur yang jauh lebih ideal. Kemampuan *scale-out* yang agresif dipadukan dengan skema penagihan presisi (*per-second billing*) pada Cloud Run terbukti lebih efektif dalam mencegah antrean pemrosesan sekaligus menekan pemborosan biaya operasional pada kondisi beban puncak.

B. Rekomendasi untuk Penelitian Selanjutnya

Ruang lingkup pengujian pada penelitian ini sengaja difokuskan pada arsitektur *backend RESTful API* sederhana sebagai bentuk *micro-benchmark* murni. Oleh karena itu, penelitian selanjutnya sangat disarankan untuk mengeksplorasi pengujian pada jenis beban kerja yang jauh lebih kompleks dan bervariasi. Pengujian pada sistem komputasi berbasis *stateful*, pemrosesan aliran data waktu nyata (*real-time streaming*), maupun arsitektur *microservices* yang melibatkan kueri basis data tingkat tinggi akan memberikan peta komparasi yang lebih komprehensif mengenai batasan toleransi dari masing-masing layanan.

Selain variasi beban kerja, evaluasi arsitektur *cloud* di masa depan juga dapat diperluas dengan memasukkan metrik pengujian di luar fase sistem berjalan (*runtime*). Peneliti selanjutnya dapat menganalisis tantangan durasi pemanasan instans (*cold start latency*), tingkat kerumitan alur pengerahan kode otomatis (*CI/CD deployment pipeline*), serta komparasi ketahanan keamanan jaringan. Perluasan komparasi lintas penyedia layanan *cloud* (*multi-cloud architecture*), seperti membandingkan ekosistem Google Cloud dengan arsitektur AWS Lambda maupun Azure Functions, juga dapat menjadi arah riset yang sangat relevan untuk memperkaya literatur rekayasa perangkat lunak berbasis *cloud computing*.



DAFTAR PUSTAKA

- [1] P. M. Mell and T. Grance, "The NIST definition of cloud computing," Gaithersburg, MD, Sep. 2011. doi: 10.6028/NIST.SP.800-145.
- [2] E. Van Eyk, L. Toader, S. Talluri, L. Versluis, A. Uta, and A. Iosup, "Serverless is More: From PaaS to Present Cloud Computing," *IEEE Internet Comput.*, vol. 22, no. 5, pp. 8–17, Sep. 2018, doi: 10.1109/MIC.2018.053681358.
- [3] H. B. Hassan, S. A. Barakat, and Q. I. Sarhan, "Survey on serverless computing," *Journal of Cloud Computing: Advances, Systems*, vol. 10, no. 1, Dec. 2021, doi: 10.1186/s13677-021-00253-7.
- [4] M. Golec, G. K. Walia, M. Kumar, F. Cuadrado, S. S. Gill, and S. Uhlig, "Cold Start Latency in Serverless Computing: A Systematic Review, Taxonomy, and Future Directions," *ACM Comput. Surv.*, vol. 57, no. 3, Nov. 2024, doi: 10.1145/3700875.
- [5] I. Barokah and Asriyanik, "Analisis Perbandingan Serverless Computing Pada Google Cloud Platform," *Jurnal Teknologi Informatika dan Komputer MH. Thamrin*, vol. 7, no. 2, pp. 169–187, 2021.
- [6] Z. Z. Maulidia and Venica Liptia, "Serverless Computing: Analisis Cloud Run dan App Engine dalam Profile Website Deployment," *SISTEMASI: Jurnal Sistem Informasi*, vol. 13, no. 2, pp. 492–505, 2024.
- [7] Datadog, "The State of Serverless | Datadog." Accessed: Nov. 08, 2025. [Online]. Available: <https://www.datadoghq.com/state-of-serverless/>
- [8] I. Barokah and Asriyanik, "View of Analisis Perbandingan Serverless Computing Pada Google Cloud Platform (10)," *Jurnal Teknologi Informatika dan Komputer MH. Thamrin*, vol. 7, no. 2, pp. 169–187, 2021.
- [9] Z. Z. Maulidia and Venica Liptia, "Serverless Computing_ Cloud Run and App Engine Analysis in Profile Website Deployment _ Maulidia _ SISTEMASI (6)," *SISTEMASI: Jurnal Sistem Informasi*, vol. 13, no. 2, pp. 492–505, Jan. 2024.

- [10] J. Yang and A. Abraham, "Analyzing the Features, Usability, and Performance of Deploying a Containerized Mobile Web Application on Serverless Cloud Platforms," *Future Internet*, vol. 16, no. 12, Dec. 2024, doi: 10.3390/fi16120475.
- [11] S. Eismann, J. Grohmann, E. Van Eyk, N. Herbst, and S. Kounev, "Predicting the costs of serverless workflows," in *ICPE 2020 - Proceedings of the ACM/SPEC International Conference on Performance Engineering*, Association for Computing Machinery, Inc, Apr. 2020, pp. 265–276. doi: 10.1145/3358960.3379133.
- [12] R. Annisa, A. R. Makarim, M. Afif, W. E. Sulistiono, and S. Ferbangkara, "Analisis Kinerja Layanan Cloud Computing dalam Sistem Cerdas Rekomendasi Tanaman Perkebunan," in *SINTA 7*, Seminar Nasional Ilmu Teknik dan Aplikasi Industri (SINTA), 2025, pp. 285–289.
- [13] K. C. Cheekuri, "SLO-First Autoscaling for Multi-Tenant Microservices: A Control-Theoretic Approach to P95/P99 Latency," *Journal of Information Systems Engineering and Management*, vol. 2025, no. 61s, pp. 967–976, 2025, [Online]. Available: <https://www.jisem-journal.com/>
- [14] A. A. Nafasha, I. P. E. Indrawan, and G. I. Setiawan, "ANALISIS PERBANDINGAN BIAYA DAN SERVERLESS COMPUTING PADA GOOGLE CLOUD PLATFORM," *Jurnal Manajemen dan Teknologi Informasi (JMTI)*, vol. 14, no. 1, pp. 1–9, 2024.
- [15] J. Yang and A. Abraham, "Analyzing the Features, Usability, and Performance of Deploying a Containerized Mobile Web Application on Serverless Cloud Platforms," *Future Internet*, vol. 16, no. 12, Dec. 2024, doi: 10.3390/fi16120475.
- [16] V. Fujiyanti, G. M. Suranegara, and I. N. Ichsan, "Comparative Analysis of Server-Based and Serverless Service Performance on Google Cloud Platform (GCP) (Case Study: Machine Learning Model Deployment)," *Journal of Information Systems and Informatics*, vol. 6, no. 2, pp. 1172–1194, Jun. 2024, doi: 10.51519/journalisi.v6i2.773.
- [17] A. Prasetyo and D. Kriestanto, "Analisis Perbandingan GraphQL dan REST API pada Aplikasi Menu Restoran dengan Node.js," *JIKO (Jurnal Informatika dan Komputer)*, vol. 9, no. 2, p. 274, Jun. 2025, doi: 10.26798/jiko.v9i2.1521.

- [18] K. Azkiya, M. Irsan, and M. F. Fathoni, "Implementation of App Engine and Cloud Storage as REST API on Smart Farm Application," *Jurnal dan Penelitian Teknik Informatika*, vol. 8, no. 2, 2024, doi: 10.33395/v8i2.13386.
- [19] M. Firdaus and R. Afwani, "PENGEMBANGAN RESTFUL API UNTUK APLIKASI KLASIFIKASI JENIS TANAH BERBASIS MOBILE PADA GOOGLE CLOUD (Restful Api Development For Mobile-Based Soil Type Classification Applications On Google Cloud)," *Jurnal Teknologi Informasi, Komputer dan Aplikasinya (JTIKA)*, vol. 6, no. 1, pp. 275–287, 2024, [Online]. Available: <http://jtika.if.unram.ac.id/index.php/JTIKA/>
- [20] M. Amin, "Perbandingan Kinerja Nginx dan Caddy sebagai Web Server untuk Aplikasi PHP," *Insect (Informatics and Security) : Jurnal Teknik Informatika*, vol. 11, no. 1, pp. 88–96, 2025.
- [21] R. L. Rahardian, "Analisa Perbandingan Cloud Management Pada Google Cloud Platform dan Amazon Web Services," *Digital Transformation Technology*, vol. 5, no. 1, pp. 97–106, May 2025, doi: 10.47709/digitech.v5i1.5862.
- [22] Google Cloud, "Google Cloud Documentation." Accessed: Nov. 10, 2025. [Online]. Available: <https://cloud.google.com/docs>
- [23] Google Cloud, "Cloud Run documentation | Google Cloud." Accessed: Nov. 10, 2025. [Online]. Available: <https://cloud.google.com/run/docs>
- [24] Google Cloud, "App Engine documentation | Google Cloud." Accessed: Nov. 10, 2025. [Online]. Available: <https://cloud.google.com/appengine/docs>
- [25] Google Cloud, "What Are Containers? | Google Cloud." Accessed: Dec. 04, 2025. [Online]. Available: <https://cloud.google.com/learn/what-are-containers?hl=en>
- [26] Docker Inc., "What is a Container? | Docker." Accessed: Dec. 04, 2025. [Online]. Available: <https://www.docker.com/resources/what-container/>
- [27] The Go Authors, "Documentation - The Go Programming Language." Accessed: Dec. 04, 2025. [Online]. Available: <https://go.dev/doc/>
- [28] Gin Gonic, "Introduction | Gin Web Framework." Accessed: Dec. 04, 2025. [Online]. Available: <https://gin-gonic.com/en/docs/introduction/>

- [29] GORM, “GORM Guides | GORM - The fantastic ORM library for Golang, aims to be developer friendly.” Accessed: Dec. 04, 2025. [Online]. Available: <https://gorm.io/docs/>
- [30] Google Cloud, “Cloud SQL overview | Cloud SQL for MySQL | Google Cloud Documentation.” Accessed: Dec. 04, 2025. [Online]. Available: <https://docs.cloud.google.com/sql/docs/mysql/introduction>
- [31] R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Ph.D. dissertation, University of California, Irvine, Irvine, California, 2000.
- [32] Grafana Labs, “Types of load testing | Grafana Labs.” Accessed: Dec. 27, 2025. [Online]. Available: <https://grafana.com/load-testing/types-of-load-testing/>
- [33] A. Ismail, A. Yuli Ananta, S. Noor Arief, and E. Nur Hamdana, “PERFORMANCE TESTING SISTEM UJIAN ONLINE MENGGUNAKAN JMETER PADA LINGKUNGAN VIRTUAL,” *Jurnal Informatika Polinema*, 2023.
- [34] L. Xu *et al.*, “A Reinforcement Learning Based Approach to Identify Resource Bottlenecks for Multiple Services Interactions in Cloud Computing Environments,” in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 58–74. doi: 10.1007/978-3-030-67540-0_4.
- [35] Google Cloud, “Cloud Monitoring overview | Google Cloud.” Accessed: Nov. 10, 2025. [Online]. Available: <https://cloud.google.com/monitoring/docs/monitoring-overview>
- [36] Google Cloud, “Analyze billing data and cost trends with Reports | Cloud Billing | Google Cloud Documentation.” Accessed: Apr. 06, 2026. [Online]. Available: <https://docs.cloud.google.com/billing/docs/how-to/reports>
- [37] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*, 19th ed. Bandung: ALFABETA, CV., 2013.