

ANALISIS PERBANDINGAN BIAYA DAN SERVERLESS COMPUTING PADA GOOGLE CLOUD PLATFORM

Adelia A'fa Nafasha¹, I Putu Eka Indrawan², Gde Iwan Setiawan³

¹ Universitas PGRI Mahadewa Indonesia, Denpasar, Indonesia; adeliadeaa313@gmail.com

² Universitas PGRI Mahadewa Indonesia, Denpasar, Indonesia; putueka@mahadewa.ac.id

³ Universitas PGRI Mahadewa Indonesia, Denpasar, Indonesia; iwansetiawan@mahadewa.ac.id

ARTICLE INFO

Article history:

Received April 16, 2024

Revised April 23, 2024

Accepted April 25, 2024

Available online April 30, 2024

Keywords: Cloud Computing, Google Cloud, Serverless, Cloud Translation API

Copyright ©2023 by Author. Published by Lembaga Pengembangan Pembelajaran, Penelitian, dan Pengabdian Masyarakat Universitas PGRI Mahadewa Indonesia

Abstract. Cloud computing is a technological advancement enabling virtual access to IT resources via the internet, although its management isn't always straightforward. To address this challenge, the concept of serverless computing has been developed as an innovative solution, allowing users to pay based on their usage needs. However, many companies still struggle with determining the use of serverless and calculating associated costs. This research aims to qualitatively analyze three types of serverless offerings in GCP: App Engine, Cloud Functions, and Cloud Run, using a user stories model. Additionally, it calculates cost estimates using the Google Pricing Calculator. This study provides insights into comparing each serverless type and assists companies in selecting the appropriate solution. The analysis in this research identifies App Engine as one of the serverless options offering optimal and flexible usage at an affordable price. App Engine is well-suited for businesses or developers seeking a straightforward deployment process. However, it is essential to note that the selection of a serverless platform should be tailored to the needs and complexity of the applications to be executed

PENDAHULUAN

Di era digital yang semakin berkembang, teknologi terus memberikan inovasi baru untuk memudahkan kehidupan. Salah satu teknologi berkembang dan menjadi solusi inovatif bagi industri adalah komputasi awan (*cloud computing*). Komputasi awan (*cloud computing*) adalah pemanfaatan teknologi untuk mengakses sumber daya TI seperti penyimpanan, pemrosesan data, server, secara virtual melalui internet yang bersifat elastis dan dapat diskalakan. (Hanafi, 2022). *Cloud computing* memungkinkan percepatan pengembangan aplikasi dengan menggunakan sumber daya yang dapat ditingkatkan atau diturunkan dengan cepat. Manfaat lain termasuk kemudahan analisis data di lingkungan *cloud*, *streaming* audio dan video tanpa hambatan, serta penyimpanan, pencadangan, dan pemulihan data yang efisien (Barokah & Asriyanik, 2021). Namun, dalam penerapannya mengelola komputasi awan tidaklah mudah, para developer mengalami berbagai tantangan dalam mengelola infrastruktur server dan pengembangan aplikasinya seperti ketersediaan, penyeimbangan beban, penskalaan otomatis, keamanan, pemantauan dan sebagainya. Dengan banyaknya tantangan tersebut akhirnya hadir konsep layanan serverless (komputasi awan tanpa server) (Hassan *et al.*, 2021)

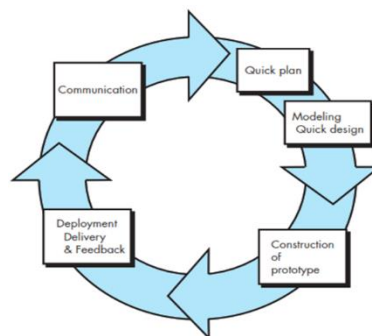
Konsep *serverless computing* muncul sebagai solusi yang lebih inovatif. *Serverless computing*, sebagai model eksekusi cloud, menyediakan layanan *back-end* berdasarkan kebutuhan tanpa memerlukan pelanggan membayar tarif tetap untuk server atau bandwidth. Ini membebaskan pengembang dari tanggung jawab pemeliharaan server, memungkinkan mereka fokus pada pengembangan logika bisnis, dan meningkatkan produktivitas tim. Istilah ini juga disebut sebagai *Back-End as a Service*, tetapi berkembang menjadi solusi fungsi sebagai layanan, termasuk *Function as a Service* (FaaS). *Serverless computing* memungkinkan pengembang membangun aplikasi lebih cepat dengan menghilangkan kebutuhan untuk mengelola infrastruktur, memberikan nilai tambah ke inti bisnis, dan memungkinkan organisasi lebih optimal dalam penggunaan sumber daya serta fokus pada inovasi (Barokah & Asriyanik, 2021)

Salah satu perusahaan yang menyediakan layanan komputasi awan (*cloud computing*) berbasis *serverless* adalah Google Cloud Platform (GCP). GCP menyediakan layanan *serverless* yang disesuaikan berdasarkan kebutuhan pengguna dan biaya yang dikeluarkan juga berbeda-beda setiap layanan tersebut. Namun, banyak perusahaan menghadapi kesulitan dalam memilih layanan *serverless* yang sesuai dengan kebutuhan aplikasi mereka, serta memperkirakan biaya yang akan dikeluarkan. Oleh karena itu, tujuan penelitian ini adalah untuk menganalisis perbandingan layanan *serverless* beserta biaya masing-masing layanan tersebut yang dapat disesuaikan dan menjadi pilihan terbaik dalam kebutuhan perusahaan

Berdasarkan latar belakang masalah tersebut, penulis mengambil penelitian yang berjudul **“Analisis Perbandingan Biaya dan Serverless Computing pada Google Cloud Platform”** untuk menjelaskan secara rinci dan detail terhadap layanan dan biaya yang dikeluarkan pada penggunaan *serverless* untuk dijadikan perbandingan berdasarkan preferensi dan kebutuhan pengguna

METODE

Pada penelitian ini, sistem yang dirancang adalah dengan melakukan taksonomi karakteristik untuk membandingkan karakteristik biaya dan layanan *serverless* kemudian melakukan analisis kualitatif yang dapat menghasilkan informasi untuk menentukan layanan yang sesuai berdasarkan preferensi kebutuhan pengguna. Dalam penggunaan komputasi awan, layanan *serverless* sangat populer digunakan dan diminati banyak perusahaan karena infrastruktur yang disediakan serta harga yang relatif rendah. Analisis ini bertujuan untuk memberikan wawasan yang berharga bagi organisasi dan pengembang yang mempertimbangkan penggunaan *serverless* dalam proyek mereka (Barokah & Asriyanik, 2021)



Gambar 1. Struktur Tahapan Model Prototyping

Dalam pelaksanaan perancangan sistem, peneliti menggunakan metode pengembangan model prototyping dan *serverless computing* agar pengembang dapat lebih fokus pada logika inti dan membuat prototipe aplikasi dengan cepat, tanpa harus mengelola *server* secara langsung, metode ini juga membantu pengembang memahami cara kerja aplikasi berinteraksi dengan layanan *serverless*, seperti fungsi *trigger* dan *event-driven architectures*. (Barokah & Asriyanik, 2021). Prototyping adalah

suatu metode dalam pengembangan perangkat lunak berfungsi untuk membuat kerangka kerja sistem. Prototipe merupakan tahap awal sebuah sistem untuk merepresentasikan gambaran dalam bentuk rancangan untuk mengetahui cara kerja sistem tersebut. Dengan adanya prototipe agar pengembang dan pengguna dapat berinteraksi dengan baik dan mendapatkan umpan balik (feedback) (Rizal & Fachri, 2023)

Berikut tahapan-tahapan dalam pengembangan metode prototyping :

1. Identifikasi kebutuhan dan Perencanaan
Tahapan awal adalah mengenali serta memahami kebutuhan dari pengguna dan pihak yang terlibat. Tim pengembang perangkat lunak perlu memperoleh informasi detail dan jelas tentang harapan pengguna terhadap aplikasi atau sistem yang akan digunakan melalui metode kualitatif dengan model *user stories*. Hal ini diperlukan agar informasi yang diperoleh dapat menjadi dasar yang akurat dalam menentukan sistem yang sesuai dengan kebutuhan pengguna
2. Perancangan Prototipe
Pada tahapan perancangan prototipe terbagi menjadi dua yaitu, rancangan dasar (*low-fidelity prototype*) dan rancangan detail (*high-fidelity prototype*). Pada tahapan ini pengembang dapat menyesuaikan rancangan yang dibuat berdasarkan estimasi waktu dan kebutuhan. Pada intinya adalah tujuan perancangan ini untuk menyampaikan konsep yang dibuat kepada pengguna
3. Pengujian dan Evaluasi
Setelah prototipe dibuat, selanjutnya melakukan uji coba dan evaluasi untuk memverifikasi kinerja sistem pada GCP yang lebih baik dan berdasarkan kebutuhan pengguna, serta untuk mendapatkan umpan balik. Setelah itu dilakukan analisis melalui hasil monitoring yang ada pada GCP untuk melihat perbandingan antar *serverless*
4. Perbaikan dan Penggunaan
Pada tahapan ini melakukan perbaikan sistem dan pengujian agar sistem benar-benar berfungsi dengan baik dan sesuai yang diharapkan pengguna. Setelah semua dilakukan, tahap terakhir adalah peluncuran atau proses penggunaan dan penyebaran sistem (*deploying*) (Rizal & Fachri, 2023)

HASIL DAN PEMBAHASAN

Pada tahap pembahasan sistem, setelah model dilakukan tahap *deployment* pada GCP dan menghasilkan API terjemah Bahasa Spanyol, selanjutnya dilakukan analisis untuk menghasilkan pembahasan yang berguna bagi para pengguna. Dalam proses analisis, penulis melakukan uji coba aplikasi API *translate* ke 15 pengguna secara bersamaan di tanggal 8 sampai 12 Maret 2024 yang di bagi masing-masing *serverless App Engine*, *Cloud Functions* dan *Cloud Run* dengan tabel sebagai berikut:

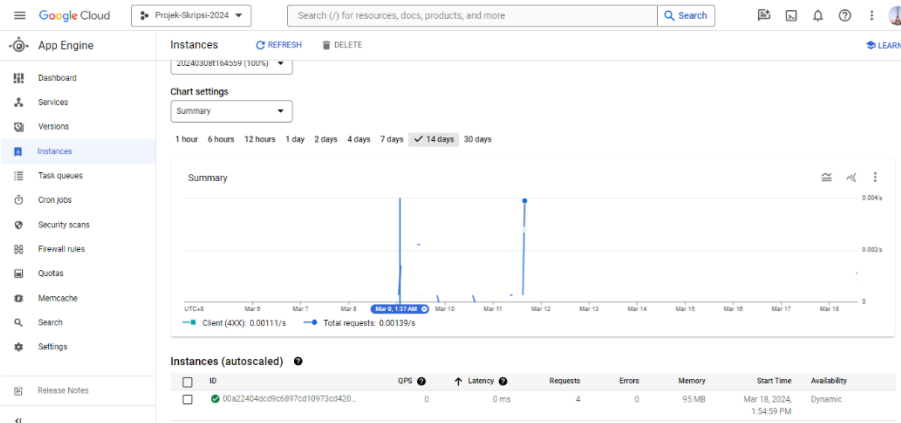
No	App Engine	Cloud Functions	Cloud Run
1	Robi	Mila	Haeru
2	Amelisa	Febiola	Putri
3	Nana	Romil	Nanda
4	Setia	Hatta	Okta
5	Eka	Dean	Yiyin

Tabel 1. Uji Coba Aplikasi API ke *Serverless* dengan 15 *User*

Setelah melakukan uji coba kepada *user*, hasilnya dapat dilihat pada bagian sub nomor berikutnya melalui analisis *monitoring* pada setiap layanan *serverless*

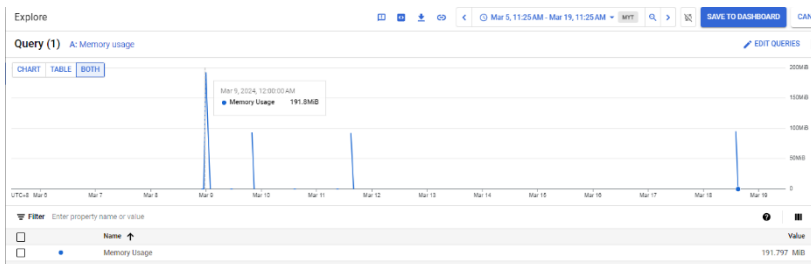
Monitoring Serverless GCP

1. App Engine



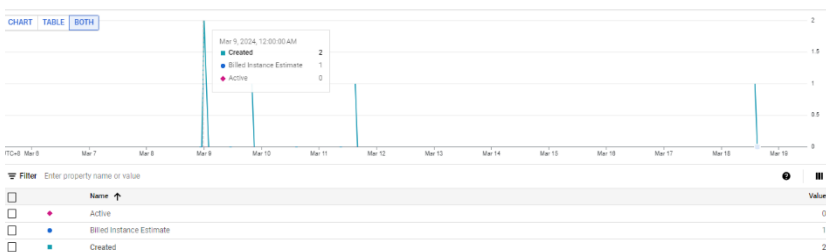
Gambar 2. Hasil *Monitoring* pada App Engine

Pada *monitoring* yang dihasilkan di *App Engine* adalah jumlah keseluruhan waktu dihitung 14 hari sejak tanggal 6 Maret sampai 18 Maret 2024. Setelah melakukan uji coba dengan 5 *user*, terlihat terdapat metrik-metrik yang dihasilkan seperti QPS (*queries per second*), latensi, *error* dan penggunaan memori. Pada puncak response yang dihasilkan terlihat pada tanggal 9 Maret 2024. Berdasarkan data uji coba yang dihasilkan, pada tanggal tersebut terdapat 5 pengguna yang menggunakan aplikasi *API translate*. Pada saat proses permintaan data menerima *request* dari *user* dengan kecepatan 0,00667/s kemudian melakukan response dari *user* sebesar 0,00361/s. Hal ini menandakan bahwa aplikasi berjalan dengan baik dan tidak terdapat *error* yang dihasilkan



Gambar 3. Penggunaan Memori yang dihasilkan pada App Engine

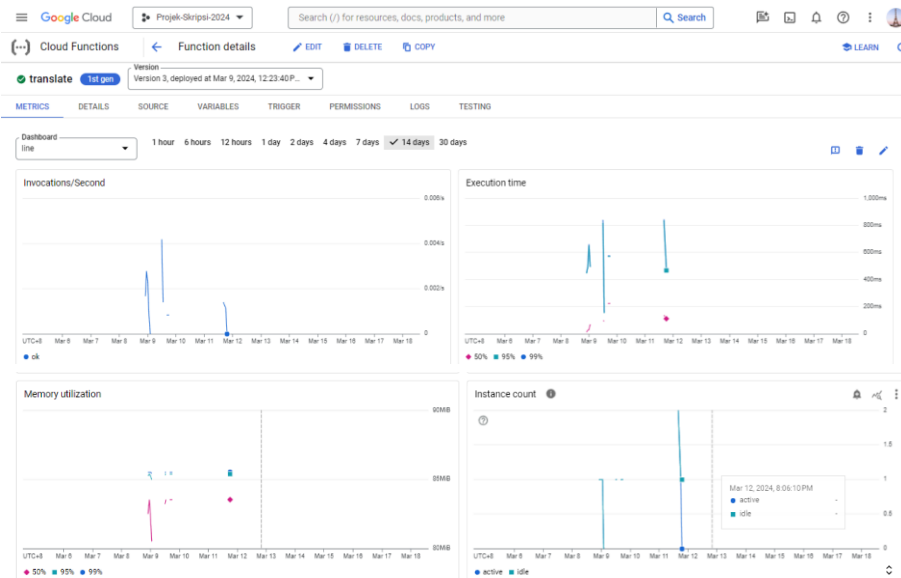
Kemudian pada penggunaan memori, puncaknya di tanggal yang sama yaitu 9 Maret 2024, memori yang dihasilkan sebesar 191.8 MiB (Mebibyte)



Gambar 4. Penggunaan Instance yang dibuat pada App Engine

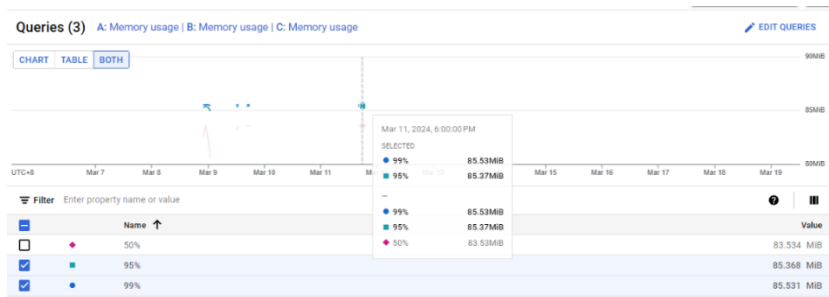
Selanjutnya, pada penggunaan instance, juga mengalami hal yang sama, puncaknya berada di tanggal 9 Maret 2024, yang dihasilkan sebesar 2 instance

2. Cloud Functions



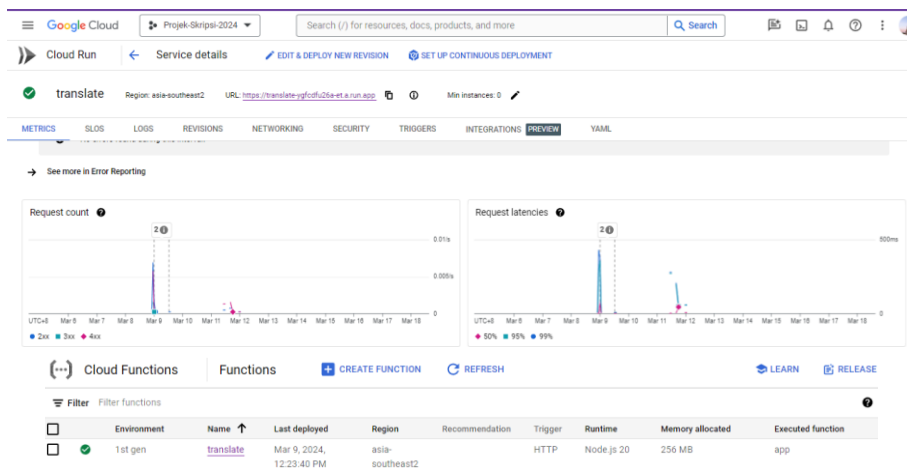
Gambar 5. Hasil *Monitoring* pada Cloud Functions

Monitoring yang dihasilkan pada *Cloud Functions*, dapat dilihat puncak aktivitas berada di tanggal yang sama yaitu 9 Maret 2024. Pada tabel sebelah kiri, “*invocations/seconds*” grafik menunjukkan perhitungan invokasi fungsi per detik yang dihasilkan sebesar 0.00417/s, artinya jumlah beban kerja untuk memanggil fungsi “*translate*” dapat menunjukkan performa baik karena angka tidak melebihi dari 1. Kemudian, pada tabel sebelah kanan “*execution time*” grafik tersebut juga mengalami puncak aktivitas di tanggal yang sama. Pada tabel sebelah kiri bagian “*memory utilization*” memori yang dihasilkan bervariasi dan paling banyak dihasilkan pada tanggal 11 Maret 2024 sebesar 85.53 MiB (Mebibyte). Terakhir, pada penggunaan *serverless* Cloud Functions memori yang dialokasikan sebesar 256 MB (Megabyte).



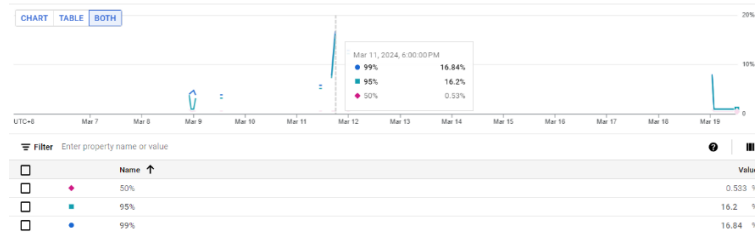
Gambar 6. Memori Penggunaan yang dihasilkan pada Cloud Functions

3. Cloud Run



Gambar 7. Hasil *Monitoring* Pada Cloud Run

Pada monitoring yang terdapat di Cloud Run, juga mengalami puncak aktivitas yang sama, yaitu di tanggal 9 Maret 2024. Namun pada penggunaan CPU Utilization, terjadi peningkatan di tanggal 11 Maret 2024 sebanyak 16.84%. Selanjutnya, pada bagian “*container CPU utilization*” terdapat metrik dengan puncak aktivitas tinggi di tanggal 11 Maret 2024, sebesar 16.84% dan di tanggal 9 Maret 2024 sebesar 3.85% menandakan bahwa data diakses lebih sering di tanggal 11 Maret 2024. Pada bagian “*container memory utilization*” terjadi puncak aktivitas pada tanggal 9 Maret 2024 sehingga penggunaan query meningkat sebesar 29.99%. Pada bagian penyimpanan, memori yang dialokasikan oleh Cloud Run sebesar 512 MiB (Mebibyte)



Gambar 8 Penggunaan CPU Utilization pada Cloud Run

1.1.1 Perhitungan Biaya Penggunaan Serverless Menggunakan Google Pricing Calculator

Image URL	asia-southeast2-docker.pkg.dev/projek-skripsi-2024/c...
Port	8080
Build	(no build information available) ?
Source	(no source information available) ?
Command and args	(container endpoint)
CPU limit	1
Memory limit	512MiB

Gambar 9. Deskripsi penggunaan pada Cloud Run

Tabel 2. Perbandingan Biaya Penggunaan *Serverless* 2024

No	Serverless Comparison		
1	<p>App Engine</p> <p>App Engine standard environment instances</p> <p>Jakarta</p> <p>Instance Type: F1</p> <p>Instance Hours: 730 per month</p> <p>IDR 0.00</p> <p>App Engine APIs and Services</p> <p>Jakarta</p> <p>Outgoing Network Traffic: 0.000 GiB</p> <p>Cloud Storage: 0.093 GiB</p> <p>Memcache: 0.713 GiB hours</p> <p>IDR 869.98</p> <p>Total Estimated Cost: IDR 869.98 per 1 month</p> <p>Estimate Currency IDR - Indonesian Rupiah</p> <p>Cloud Translation</p> <p>Text Translation[*]: 4,000 characters IDR 0.00</p> <p>Language Detection[*]: 4,000 characters IDR 0.00</p> <p>IDR 0.00</p> <p><small>* First 500,000 characters per month is free (applied as \$10 credit every month). This credit does not apply to formatted document translations.</small></p>	<p>Cloud Functions</p> <p>Cloud Functions</p> <p>Projek-skripsi-2024</p> <p>Region: Jakarta</p> <p>Invocations: 0 IDR 0.00</p> <p>RAM (GiB-seconds): 0 per month IDR 0.00</p> <p>CPU (GHz-seconds): 0 per month IDR 0.00</p> <p>Minimum number of instances: 1 IDR 59,988.79</p> <p>IDR 59,988.79</p> <p>Cloud Build</p> <p>Default pool</p> <p>Region: Jakarta</p> <p>Machine type: standard-1</p> <p>Build minutes: 1 IDR 0.00</p> <p>IDR 0.00</p> <p>Total Estimated Cost: IDR 59,988.79 per 1 month</p> <p>Estimate Currency IDR - Indonesian Rupiah</p> <p>Cloud Translation</p> <p>Text Translation[*]: 4,000 characters IDR 0.00</p> <p>Language Detection[*]: 4,000 characters IDR 0.00</p> <p>IDR 0.00</p> <p><small>* First 500,000 characters per month is free (applied as \$10 credit every month). This credit does not apply to formatted document translations.</small></p>	<p>Cloud Run</p> <p>Cloud Run</p> <p>Projek-Skripsi-2024</p> <p>Region: Jakarta</p> <p>CPU allocation type: CPU is only allocated during request processing</p> <p>CPU: 1</p> <p>Memory: 0.5 GiB</p> <p>CPU allocation time: 0.015 vCPU-second IDR 0.00</p> <p>Memory allocation time: 0.008 GiB-second IDR 0.00</p> <p>Number of requests: 1 IDR 0.00</p> <p>Minimum number of instances: 1 IDR 215,860.96</p> <p>IDR 215,860.96</p> <p>Total Estimated Cost: IDR 215,860.96 per 1 month</p> <p>Estimate Currency</p> <p>Cloud Translation</p> <p>Text Translation[*]: 4,000 characters IDR 0.00</p> <p>Language Detection[*]: 4,000 characters IDR 0.00</p> <p>IDR 0.00</p> <p><small>* First 500,000 characters per month is free (applied as \$10 credit every month). This credit does not apply to formatted document translations.</small></p>

1.1.2 Tabel Hasil Perbandingan Serverless Computing

Tabel 3. Hasil Ringkasan Perbandingan Serverless Computing

Perbedaan	App Engine	Cloud Functions	Cloud Run
Bandwidth	0,00667/s	0.00417/s	0.1684/s
Skalabilitas	Penskalaan otomatis berdasarkan beban aplikasi	Penskalaan otomatis berdasarkan jumlah pemanggilan fungsi	Penskalaan otomatis berdasarkan jumlah permintaan yang masuk ke dalam container.
Kasus Penggunaan	Memerlukan lebih sedikit overhead manajemen	Berbasis peristiwa, pendekatan ringan	Memerlukan banyak fleksibilitas dan kontrol
Portabilitas	Menjalankan aplikasi di lingkungan tertentu	Menjalankan aplikasi di lingkungan apapun	Menjalankan aplikasi di lingkungan apapun
Penyimpanan	191.8 MiB	85.53 MiB	512 MiB
Ketentuan Biaya	Penagihan berdasarkan konsumsi sumber daya dan lebih rendah	Penagihan berdasarkan jumlah panggilan dan durasi fungsi	Penagihan per detik dan konsumsi RAM, CPU lebih besar
Kegunaan Utama	Dapat Menghosting aplikasi, membuat lingkungan pengembangan, pengujian, staging, produksi (autentikasi, hosting, scheduler)	Membangun dan menghubungkan berbagai macam layanan cloud, beban kerja berbasis peristiwa (IoT, ETL, webhook, <i>Kafka</i> , <i>Apache</i> , Analitik)	Menggunakan alur kerja container dan terintegrasi ekosistem container (Cloud Build, Artifact Registry, Docker)
Biaya yang di keluarkan	Rp. 869.98	Rp. 59.988.79	Rp. 215,869.96

SIMPULAN

Berdasarkan hasil penelitian dapat disimpulkan bahwa ketiga platform *serverless* mampu melakukan proses *deployment* dengan baik dan berhasil digunakan oleh pengguna. Namun, ketiganya memiliki

kekurangan dan kelebihan masing-masing. Dalam penelitian ini, *serverless* yang paling cocok dan fleksibel untuk digunakan, serta memiliki harga yang relatif rendah sesuai dengan kasus permasalahan, adalah *App Engine*. *App Engine* tidak memerlukan penggunaan container dan proses *deployment* tidak berbasis peristiwa. Hal ini sangat sesuai bagi bisnis atau pengembang yang menginginkan proses *deployment* yang sederhana dan biaya yang terjangkau. Jika memerlukan kontrol runtime penuh dan berbasis container pilihlah *Cloud Run*, Jika membutuhkan platform mudah digunakan pilih *App Engine* dan jika perlu menjalankan fungsi secara individual dan berbasis peristiwa pilihlah *Cloud Functions*. Jadi, tetap diingat bahwa pemilihan platform *serverless* harus disesuaikan dengan kebutuhan dan kompleksitas aplikasi yang akan dijalankan.

DAFTAR PUSTAKA

- Andi, H. K. (2021). Analysis of Serverless Computing Techniques in Cloud Software Framework. *Journal of ISMAC*, 3(3), 221–234. <https://doi.org/10.36548/jismac.2021.3.004>
- Barokah, I., & Asriyanik. (2021). Analisis Perbandingan Serverless Computing Pada Google Cloud Platform. *Jurnal Teknologi Informatika dan Komputer MH. Thamrin.Cloud Google Learn PaaS, SaaS, IaaS*. (2023, 8 10). *From Google Cloud Platform*. <https://cloud.google.com/learn/paas-vs-iaas-vs-saas>
- Bonso, J. (2023). *Google Cloud Functions vs App Engine vs Cloud Run vs GKE*. *Tutorials Dojo*. <https://tutorialsdojo.com/google-cloud-functions-vs-app-engine-vs-cloud-run-vs-gke/>
- Cloudwithease.(2024). *GCP AppEngine vs Cloud Run vs Cloud Function*. <https://cloudwithease.com/gcp-app-engine-vs-cloud-run-vs-cloud-function/>
- Dicoding. (2023). *Dicoding Academy*. *From Dicoding*: <https://www.dicoding.com/academies/133/tutorials/4641>
- Forrest, C., & Sanders, J. (2020). *Google Cloud Platforms:An Insider's Guide*. TechRepublic, 9.
- Google, (2024). *Sporting Goods Company Ups Its Game with Google App Engine*. <https://cloud.google.com/files/Hudora.pdf>
- Hassan, H. B., Barakat, S. A., & Sarhan, Q. I. (2021). Survey on serverless computing. *Journal of Cloud Computing*, 10(1). <https://doi.org/10.1186/s13677-021-00253-7>
- H. Zuchri Abdussamad, S. M. (2021). *Metode Penelitian Kualitatif*. Syakir Media Press.
- Lisdorf, A. (2021). *Cloud Computing Basics*. In A. Lisdorf, *A Non-Technical Introduction* (p. 35). Copenhagen: Apress.
- Nurhaliza, K. (2021). *Mengenal Flowchart Dan Pseudocode Dalam Algoritma Dan Pemrograman*. *Definitions*, 12. <https://doi.org/10.32388/tf77dy>
- Okhuoya, B., & Uzoma, B. (2022). *Cloud Computing*. researchgate.net, 13.
- Pakpahan, Andrew Fernando Prasetyo, Adhi Negara, Edi Surya Gurning, Kasta Situmorang, Risanti Febrine Ropita Tasnim, Tasnim Sipayung, Parlin Dony Sesilia, Ayudia Popy Rahayu, Puspita Puji Purba, Bonaraja Chaerul, Muhammad Yuniwati, Ika Siagian, Valentine Rantung, Gilny Aileen Joan (2021). *Metodologi Penelitian Ilmiah*. Yayasan Kita Menulis.
- Putu, I., Indrawan, E., Luh, N., Ambaradewi, G., Dewa, I., & Arimurti, P. R. (2023). *Sistem Informasi Tanaman Berbasis Quick Response Code Pada Taman Botani*. 13(1), 2087–5312. <https://ojs.mahadewa.ac.id/index.php/jmti>
- Sundus, A., & Rawda, A. (2022). *Developing an Academic Institution System Based On The Google Cloud Platform*. *Libyan Journal of Applied Acience and Technology*, 9.
- Ratros Y. (2019). *Serverless on Google Cloud Platform: an Introduction with Serverless Store Demo*. Medium. <https://medium.com/google-cloud/serverless-on-google-cloud-platform-an-introduction-with-serverless-store-demo-41992dec085>
- Rizal, C., & Fachri, B. (2023). *Rekayasa Teknik Informatika dan Informasi Implementasi Model Prototyping Dalam Perancangan Sistem Informasi Desa*. *RESOLUSI: Rekayasa Teknik Informatika dan Informasi*, 3(3), 52–57. <http://djournals.com/resolusi/article/view/611/396>

- Sudipta Banerjee. (2020). *Server vs Serverless Cloud Computing-Comparing Amazon EC2 and AWS Lambda*. Medium. <https://sudiptobanerjee14.medium.com/server-vs-serverless-cloud-computing-comparing-amazon-ec2-and-aws-lambda-f3684e9b840a>
- Tech, G. C. (2023). Deploy the same App to App Engine, Cloud Functions and Cloud Run? GoogleCloudTech,ServerlessExpedition. https://youtu.be/eTotLOVR7MQ?si=LDK6DNKJAjStGW_u
- Timothy. (2019). Journey to Serverless on Google Cloud Platform. Medium. <https://medium.com/google-cloud/journey-to-serverless-on-google-cloud-platform-67b8d392ffa2>