Serverless Computing: Analisis Cloud Run dan App Engine dalam *Profile Website Deployment*

Serverless Computing: Cloud Run and App Engine Analysis in Profile Website Deployment

[1]Zirlyfera Zakiya Maulidia*, [2] Liptia Venica
[1]Pendidikan Sistem dan Teknologi Informasi, Universitas Pendidikan Indonesia
[2]Mekatronika dan Kecerdasan Buatan, Universitas Pendidikan Indonesia
Jl. Veteran No. 8, Kel. Nagri Kaler, Kec. Purwakarta, Kab. Purwakarta, Jawa Barat, Indonesia 41115
*e-mail: zirlyferazm@upi.edu

(***received*:** 21 Juni 2023, ***revised*:** 30-12 2023, ***accepted*:** 16 Januari 2024)

Abstrak

Setiap individu maupun suatu perusahaan dapat memperoleh akses jaringan (on-demand network) sesuai kebutuhan, seperti untuk server, penyimpanan, dan aplikasi melalui pemanfaatan teknologi *Cloud Computing*. Google menjadi salah satu perusahaan penyedia layanan *Cloud Computing* melalui Google Cloud Platform (GCP) dan menyediakan beberapa layanan komputasi yang dapat digunakan oleh pengguna jika menginginkan *serverless computing* dalam pengelolaan servernya, yaitu Cloud Run dan App Engine. Walaupun keduanya memiliki kesamaan pada pengelolaan server dan infrastrukturnya, terdapat perbedaan yang dapat dianalisis dengan tujuan untuk dapat lebih memahami mengenai layanan *serverless computing* di Google Cloud Platform dan mengetahui perbedaan kedua layanan tersebut berdasarkan proses *deployment* suatu Profile Website. Metode yang digunakan pada penelitian ini adalah metode kualitatif untuk metode pengumpulan informasi dan metode Extreme Programming untuk pengembangan website. Dengan metode tersebut, didapatkan bahwa betul tidak perlu mengkhawatirkan mengenai server maupun infrastruktunya karena telah dikelola secara penuh oleh Google dan terdapat empat hal yang menjadi perbedaan pada Cloud Run dan App Engine, yaitu berdasarkan definisi, fitur, proses *deployme*nt, dan usabilitas.

Kata kunci: *Cloud Computing***,** *Serverless Computing*, Cloud Run, App Engine, *Website Deployment*

Abstract

Cloud Computing enables individuals or companies to obtain network access on-demand, including servers, storage, and applications. Google, as one of the Cloud Computing service providers, offers Google Cloud Platform (GCP) with various computing services. Two serverless computing options Google provides for server management are Cloud Run and App Engine. Although both services share similarities in server management and infrastructure, it is essential to analyze their differences to gain a better understanding of serverless computing on the Google Cloud Platform. This study utilizes a qualitative method for gathering information and follows the Extreme Programming method for website development. This approach shows that users need not be concerned about servers or infrastructure management as Google takes care of it entirely. Four things are different in Cloud Run and App Engine: definition, features, deployment process, and usability.

Keywords: *Cloud Computing***,** *Serverless Computing, Cloud Run, App Engine, Website Deployment*

1 Pendahuluan

Teknologi selalu mengalami perkembangan yang pesat dari tahun ke tahun, tetapi dari perkembangan tersebut memungkinkan untuk menciptakan beragam peluang dan tantangan. Selain itu, setiap teknologi yang sudah tersedia pun memiliki kelebihan dan kekurangannya masing-masing. Salah satunya *Cloud Computing*. Setiap individu maupun suatu perusahaan dapat memperoleh akses jaringan (*on-demand network*) sesuai kebutuhan, seperti untuk server, penyimpanan, dan aplikasi melalui

 ISSN:2302-8149
e-ISSN:2540-9719

pemanfaatan teknologi *Cloud Computing* [1]. *International Data Corporation* (IDC) yang merupakan suatu perusahaan riset pasar global telah memproyeksikan nilai pasar bisnis untuk *Cloud Computing* di Indonesia, yaitu dapat menembus US$ 933,63 juta di Tahun 2023 [2]. Dengan angka tersebut, dapat diketahui bahwa sebagian besar sektor kehidupan di Indonesia mulai mengadopsi teknologi tersebut, seperti pada bidang ekonomi, transportasi, kesehatan, dan lain-lain mulai memanfaatkan layanan-layanan yang disediakan oleh perusahaan penyedia layanan *Cloud Computing* dalam mengefisienkan pengelolaan infrastruktur sistem informasinya. Dengan teknologi tersebut, konsumen maupun pengusaha merasa terbantu karena mereka tidak perlu melakukan instalasi dan dapat mengakses data-data secara fleksibel menggunakan jaringan internet, sehingga dapat meningkatkan produktivitas proses bisnis [3].

Google merupakan salah satu perusahaan teknologi terbesar yang menjadi perusahaan penyedia layanan *Cloud Computing* melalui Google Cloud Platform (GCP). Platform tersebut menyediakan berbagai layanan untuk komputasi, penyimpanan data, jaringan, dan sebagainya. Terdapat layanan komputasi yang membuat penggunanya tidak perlu khawatir dalam mengelola manajemen server, yaitu *serverless computing*. Hal tersebut disebut juga dengan *Function as a Service* (*FaaS*), yaitu pengguna hanya perlu mengimplementasikan fungsinya tanpa mengelola server atau runtime environment [4]. GCP menyediakan beberapa layanan komputasi yang dapat digunakan oleh pengguna jika menginginkan *serverless computing* dalam pengelolaan servernya, seperti Cloud Run dan App Engine. Berdasarkan hasil penelitian yang dilakukan oleh Ilmi Barokah dan Asriyanik, kedua layanan tersebut memiliki kelebihan dan kekurangan masing-masing, tetapi terdapat karakteristik yang berbeda untuk setiap layanan dan pengguna dapat memilih berdasarkan kebutuhan dalam pengembangan aplikasinya [5]. Salah satu perbedaan dari keduanya terletak di pemanfaatannya, yaitu pada proses d*eployment*. Untuk mengetahui lebih mendalam mengenai perbedaan tersebut, penelitian ini melakukan *Profile Website deployment* melalui kedua layanan tersebut. Metode yang diterapkan untuk proses perbandingan tersebut dilakukan dengan dua metode, yaitu metode kualitatif dan *Extreme Programming*. Selain itu, perbandingan tersebut memiliki tujuan untuk dapat memahami lebih dalam mengenai konsep *serverless computing* pada layanan Google Cloud Platform.

2 Tinjauan Literatur

Cloud Computing membebaskan *developer* dari manajemen infrastruktur fisik, tetapi *developer* masih perlu mengelola sumber daya secara virtual. Untuk layanan *stateful*, seperti manajemen basis data ke *cloud*, pengguna perlu melakukan porting perangkat lunak agar dapat dijalankan di lingkungan yang berbeda. Ada 8 masalah yang harus diatasi dalam lingkungan cloud [6], seperti redudansi sumber daya, salinan geografis untuk cadangan, load balancing, autoscaling, monitoring, logging, pembaruan sistem, dan migrasi ke instances baru. *Serverless computing* dapat membantu mengatasi masalah tersebut. *Serverless computing* dapat didefinisikan sebagai layanan komputasi daring yang dapat membantu pengguna atau pelanggan untuk fokus dalam mengembangkan fungsionalitas aplikasinya, tanpa perlu khawatir dengan infrastruktur, pengelolaan server, dan biaya yang besar [6]–[8].

GCP menyediakan beberapa layanan *serverless computing*, termasuk Cloud Run dan App Engine. Hal tersebut dapat membuat *developer* fokus pada pengembangan aplikasi tanpa perlu mengurus server secara langsung. Penggunaan layanan *serverless computing* pada pengembangan aplikasi memiliki tiga keuntungan utama [7]. Pertama, proses *deployment* dan maintenance menjadi lebih mudah karena pengguna tidak perlu mengelola server atau infrastruktur secara langsung. Platform-as-a-Service (PaaS) menyediakan kemudahan dalam manajemen aplikasi dengan mendaftarkan fungsinya dan menerima kredensial untuk dipanggil. Kedua, *serverless computing* menawarkan skalabilitas yang terjangkau melalui multiplexing sumber daya. Permintaan dikombinasikan dan dikirim melalui saluran transmisi tunggal, meningkatkan efisiensi. Selain itu, biaya dapat ditekan dengan memilih virtual machine standar yang sesuai dengan kebutuhan. Ketiga, *serverless computing* mendukung munculnya *marketplaces* baru. Dalam era sistem operasi yang beragam pada perangkat, *developer* dapat mengembangkan fungsionalitasnya dan menjualnya kepada orang lain. *Serverless computing* membantu dalam pengelolaan penggunaan sumber daya, sehingga aplikasi-aplikasi dapat terkelola dengan baik.

Cloud Run dapat dikatakan sebagai salah satu layanan *serverless computing* yang dikelola oleh Google pada Google Cloud Platform untuk menjalankan maupun men-*deploy* suatu *container* aplikasi maupun mengimplementasikan beban kerja *stateful* dan *stateless* secara bersamaan dengan fleksibel,

 ISSN:2302-8149
e-ISSN:2540-9719

karena *developer* maupun pengguna tidak perlu mengelola infrastruktur, pembayaran sesuai penggunaan, dan terdapat penskalaan otomatis [5][9][10]. Cloud Run juga menyediakan banyak fitur [10], seperti terdapat dukungan untuk berbagai bahasa pemrograman, integrasi dengan alur kerja kontainer, kemampuan penskalaan otomatis, keamanan yang terjamin, penyimpanan sementara dan persisten, integrasi dengan berbagai layanan monitoring dan pelaporan kesalahan, kemampuan memproses traffic web dan peristiwa asinkron, serta fleksibilitas dan portabilitas melalui spesifikasi Knative. Selain itu, Cloud Run juga menyediakan endpoint HTTPS yang stabil, dukungan untuk domain kustom, dan kemampuan untuk menggunakan protokol HTTP/2, WebSockets, dan gRPC. Semua fitur ini memberikan keleluasaan dan kemudahan bagi *developer* dalam mengelola dan mengembangkan aplikasi mereka menggunakan Cloud Run.

App Engine dapat dikatakan sebagai sebuah platform dengan *serverless computing* untuk membantu *developer* dalam memproses pengembangan dan penghostingan aplikasi website dengan skala besar maupun Back-End API yang dikelola secara sepenuhnya oleh Google, sehingga tidak perlu mempermasalahkan perihal infrastrukturnya dan pembayarannya sesuai dengan penggunaan [5][9][11]. App Engine juga menyediakan banyak fitur yang menguntungkan bagi *developer* [11], seperti kemampuan untuk membangun aplikasi dengan berbagai bahasa populer, lingkungan yang dikelola sepenuhnya oleh Google, terdapat alat pemantauan dan pemecahan masalah yang efektif, kemudahan dalam *hosting* berbagai versi aplikasi, terdapat dukungan keamanan dengan aturan akses dan sertifikat SSL/TLS, serta ekosistem layanan Google Cloud yang terus berkembang. Semua ini memungkinkan *developer* untuk mengembangkan aplikasi dengan lebih efisien.

Selain itu, beberapa penelitian telah dilakukan untuk membandingkan kinerja dan usabilitas antara layanan *Serverless* dan *Server-Based* di GCP. Penelitian pertama yang dilakukan oleh Anoop Abraham dan Jeong Yang dengan judul "A Comparative Analysis of Performance and Usability on Serverless and Server-Based Google Cloud Services" [12]. Fokus penelitian tersebut adalah untuk menganalisis perbandingan kinerja dan usabilitas antara layanan *serverless computing* dan berbasis server dari Google Cloud. Terdapat evaluasi terkait berbagai metrik seperti waktu respons, penggunaan sumber daya, dan skalabilitas. Penelitian ini mengungkapkan bahwa layanan *serverless computing* menawarkan skalabilitas yang lebih baik dan konsumsi sumber daya yang lebih rendah dibandingkan layanan berbasis server. Penelitian kedua yang dilakukan oleh Xingzhi Niu, Dimitar Kumanov, Ling-Hong Hung, Wes Lloyd, dan Ka Yee Yeung dengan judul "Leveraging Serverless Computing to Improve Performance for Sequence Comparison" [13]. Fokus penelitian tersebut adalah pada penggunaan komputasi serverless cloud, khususnya pada layanan serverless yang disediakan oleh platform *cloud* tertentu. Terdapat analisis terkait faktor-faktor seperti kinerja dan efisiensi biaya. Penelitian ini mengungkapkan bahwa dengan kemampuan s*erverless computing* dan memanfaatkan ratusan CPU, hasilnya menjadi dapat lebih mudah diakses, tersedia sesuai permintaan, dan biaya yang rendah. Berdasarkan kedua penelitian tersebut, terdapat kelebihan apabila menggunakan *serverless computing* dalam hal skalabilitas, penggunaan sumber daya, dan efisiensi biaya. Namun, belum ada penelitian yang secara spesifik membandingkan Cloud Run dan App Engine untuk *Profile Website deployment*.

3 Metode Penelitian

Metode Pengumpulan Informasi

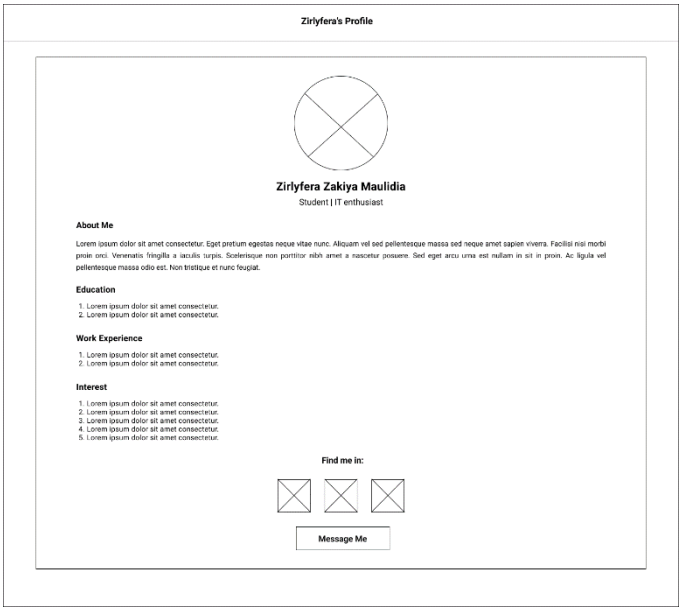
Untuk pengumpulan informasi pada penelitian ini menggunakan metode kualitatif dengan jenis studi kasus dan studi literatur. Studi kasus pada penelitian ini mengenai "Profile Website Deployment," sehingga dari proses *deployment* tersebut akan diperoleh hasil berupa deskripsi yang rinci dan mendalam terkait perbedaan untuk dua layanan komputasi yang termasuk *serverless computing*, yaitu Cloud Run dan App Engine. Sedangkan, studi literatur digunakan untuk mengkaji dan mengumpulkan data dari berbagai sumber referensi, seperti buku, jurnal, dokumentasi yang disediakan oleh Google untuk penggunaan Google Cloud Platform, dan referensi lain yang berkaitan dengan penelitian ini [14].

Metode Pengembangan Website

Untuk *Profile Website* yang akan di-*deploy* dikembangkan dengan metode *Extreme Programming*, karena untuk proses rekayasa perangkat lunak dengan pendekatan berorientasi objek dan tim berskala kecil hingga medium. Metode ini terdapat 4 tahapan, yaitu [15]:

1) *Planning*
Tahap ini untuk merencanakan dan mendeskripsikan hal-hal yang akan dimuat di dalam *Profile Website*, yaitu terdapat *header* dan *main content*. Pada *header*, hanya mencantumkan kalimat "Zirlyfera's Profile" untuk menandakan bahwa tampilan yang sedang dilihat oleh *user* merupakan garis besar dari data diri penulis. Sedangkan, pada *main content*, terdapat empat bagian, yaitu *profile*, *about me*, *profile detail*, dan *contact*. Pertama, pada *profile*, berisi foto profil, nama lengkap, dan keterangan singkat mengenai pekerjaan. Kedua, pada *about me*, berisi ringkasan mengenai data diri yang mencakup posisi saat ini, kepribadian, *soft-skills* maupun *hard-skills* yang dimiliki maupun sedang diminati untuk dipelajari. Ketiga, pada *profile detail*, berisi rincian dari bagian *about me* karena terdapat dua poin untuk memperlihatkan riwayat penulis pada *Education* dan *Work Experience*, lalu minat (*interest*) juga dituliskan kembali supaya dapat terlihat lebih jelas. Terakhir, pada *contact*, berisi tautan menuju portofolio, LinkedIn, dan e-mail.

2) *Design*
Hasil *planning* divisualisasikan dalam bentuk *Wireframe* menggunakan aplikasi Figma. *Wireframe* tersebut dapat dilihat pada Gambar 1 berikut.

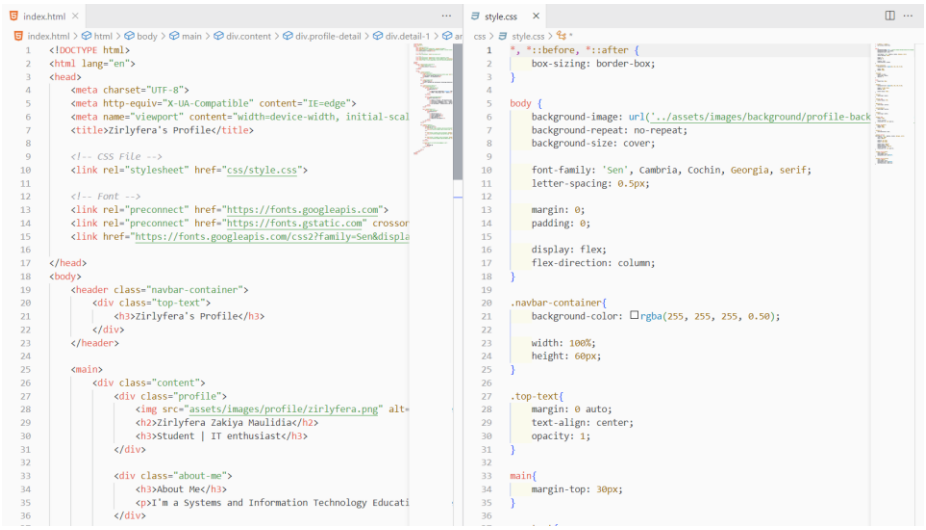


Gambar 1 *Wireframe Profile Website*

Gambar 1 menunjukkan tata letak setiap komponen untuk *header* dan *main content* yang akan diimplementasikan pada tahapan *Coding* untuk *Profile Website*.

3) *Coding*
Hasil design diimplementasikan dalam bentuk kode dengan menggunakan HTML dan CSS di Visual Studio Code. Sebagian kode tersebut dapat dilihat pada Gambar 2 berikut.

 ISSN:2302-8149
e-ISSN:2540-9719

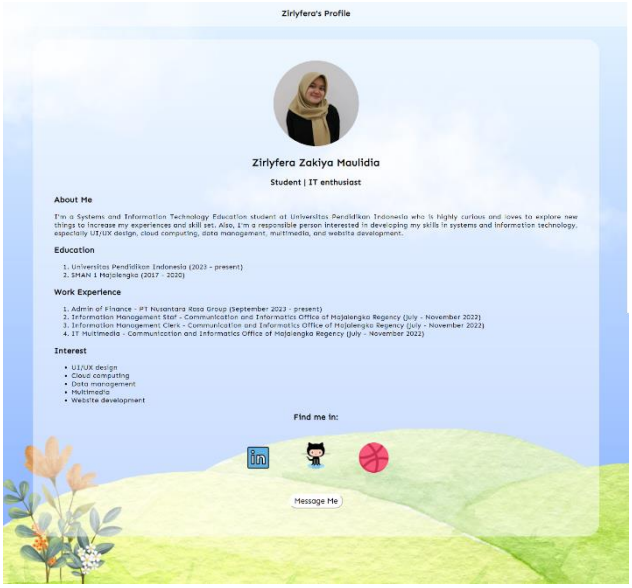


Gambar 2 Sebagian Kode *Profile Website*

Gambar 2 menunjukkan bahwa *Profile Website* dibuat dengan kode yang cukup ringkas dan sesuai kebutuhan, sehingga pada berkas HTML berisi 99 baris kode dan pada berkas CSS berisi 108 kode untuk membangun *Profile Website*.

4) *Testing*

Terakhir, *Profile Website* diuji untuk melihat hasil dari ketiga tahapan sebelumnya dan mengetahui kekurangan atau ketidaksesuaian yang dapat diperbaiki. Penulis mengujinya dengan melihat hasil tampilannya setelah seluruh kode ditulis dan mengecek secara berkala tampilannya setiap terdapat kode yang diubah. Tampilan akhir dari *Profile Webiste* dapat dilihat pada Gambar 3 berikut.



Gambar 3 Tampilan *Profile Website*

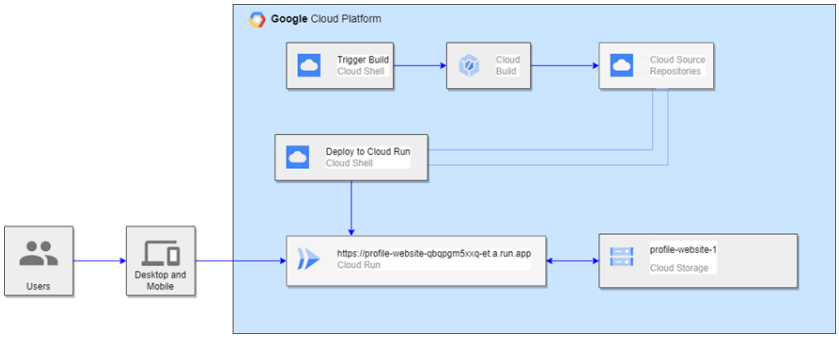
Gambar 3 menunjukkan bahwa isi *Profile Website* telah sesuai dengan yang direncanakan pada tahap *planning*, tampilannya telah sesuai dengan *wireframe* yang telah dibuat pada tahap *design*, dan kode yang telah diimplementasikan pada tahap *coding* telah berfungsi dengan tepat, sehingga tampilannya tidak tumpang tindih.

 ISSN:2302-8149
e-ISSN:2540-9719

4 Hasil dan Pembahasan

Proses *Deployment* dengan Cloud Run

Untuk *Profile Website deployment* dengan Cloud Run diperlukan perancangan *Cloud Architecture* terlebih dahulu, supaya dapat mengetahui layanan dan alur proses yang sesuai pada Google Cloud Platform. Rancangan arsitekturnya dapat dilihat pada Gambar 4.



Gambar 4 *Cloud Architecture Profile Website Deployment* dengan Cloud Run

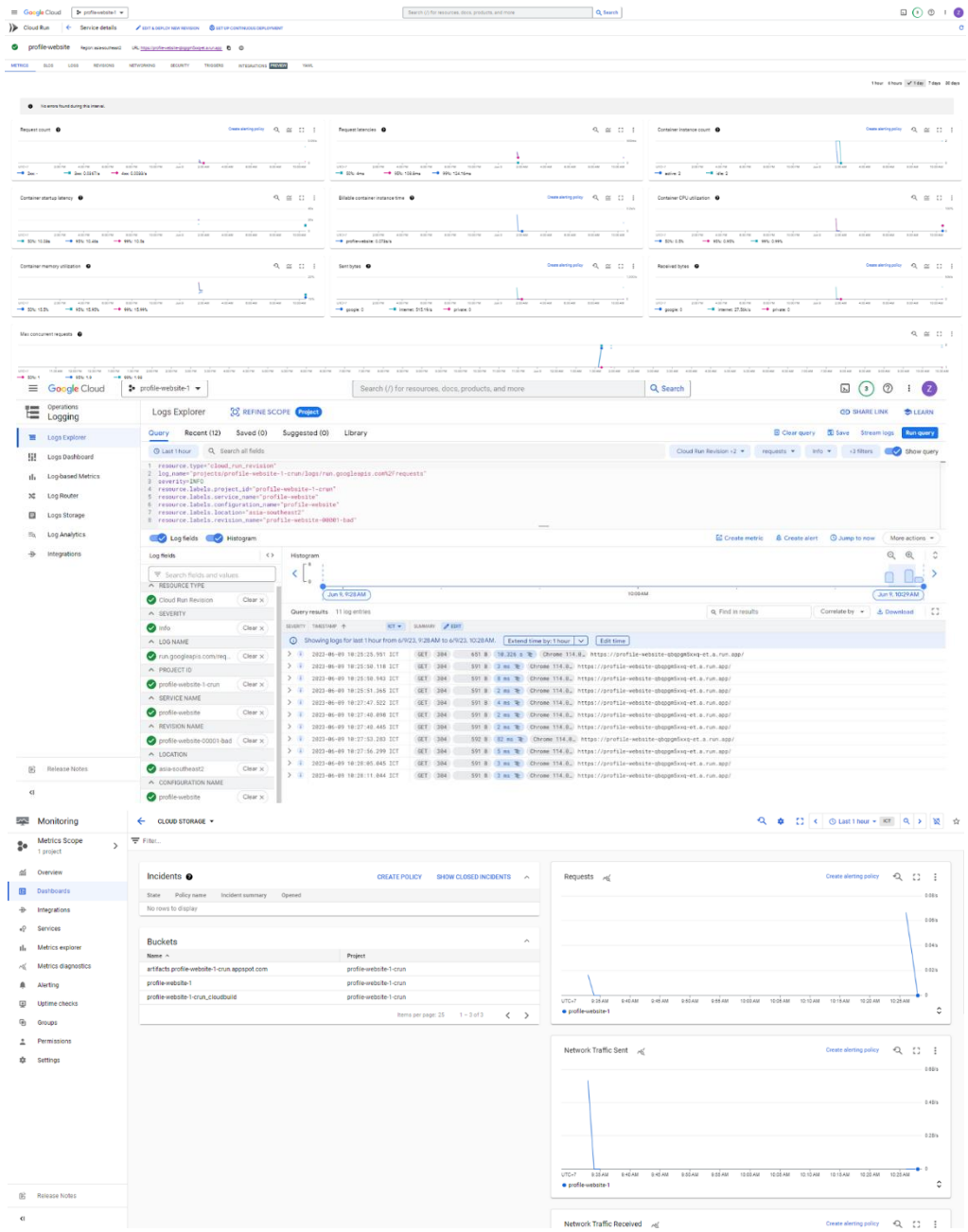
Gambar 4 menunjukkan *Cloud Architecture* untuk proses *Profile Website Deployment* dengan Cloud Run, yaitu dimulai dari Docker Image yang dibuat melalui Cloud Build, lalu Docker Image tersebut diunggah ke Cloud Run. Proses tersebut dilakukan dengan menjalankan instruksi kode di Cloud Shell. Selain itu, Cloud Storage bertugas untuk menjadi wadah *assets* dari *Profile Website* dan dapat diakses secara publik. Apabila proses *deployment* berhasil, maka *user* dapat mengakses dengan *desktop* maupun *mobile* melalui *service* URL yang tertera pada Cloud Run. Terdapat beberapa tahapan pada proses *Profile Website deployment* dengan Cloud Run, yaitu:

1) Membuka Cloud Shell Editor dan Terminal. Pada Terminal, penulis melakukan *clone* kode dari repositori GitHub yang berisi berkas HTML untuk *Profile Website*, yaitu dengan instruksi berikut: `git clone https://github.com/zirlyzkiyaa/profile-website-docker`
   Namun, jika tidak dengan cara tersebut, dapat juga dengan membuat secara manual berkas-berkas yang dibutuhkan sebagai struktur dasar *project* di Cloud Editor.
2) Apabila proses *cloning* atau pembuatan struktur dasar *project* telah selesai, selanjutnya membuka Terminal pada Cloud Shell dan masukkan perintah beriktu untuk masuk ke direktori *profile website*: `cd profile-website-docker`
3) Jika tidak terdapat Dockerfile pada struktur *project*, maka dapat membuat berkas Dockerfile terlebih dahulu. Berkas tersebut memuat *environment* yang diinginkan aplikasi, pengkonfigurasian dependensi yang dibutuhkan oleh aplikasi, perintah atau layanan yang akan dijalankan saat Docker Container dimulai, *port* yang akan diekspos dari Docker Container ke *host*, dan *variable environment* dalam Docker Container. Untuk *Profile Website*, pada berkas Dockerfile berisi kode berikut:
   ```
   FROM node:14.21.2-alpine
   WORKDIR /app
   ENV PORT 8080
   COPY . .
   RUN npm install
   EXPOSE 8080
   CMD ["npm", "run", "start"]
   ```
4) Mengakses Cloud Shell Terminal dan mengeksekusi perintah untuk membuat sebuah Container Image baru menggunakan berkas Dockerfile dengan bantuan Cloud Build. Gunakan kode berikut untuk menjalankan langkah tersebut: `gcloud builds submit --tag gcr.io/profile-website-1-crun/profile-website`
   Setelah proses pembangunan Container Image selesai oleh Cloud Build, Container Image akan secara otomatis disimpan di repositori Google Container Registry.
5) Untuk *deployment*, jalankan instruksi berikut pada terminal di Cloud Shell Terminal: `gcloud run deploy --image gcr.io/profile-website-1-crun/profile-website`

Jika sudah selesai proses *deployment*, *service* URL akan diterima, berikut contoh link yang diterima: `https://profile-website-qbqpgm5xxq-et.a.run.app/`

6) Terakhir, akses *Profile Website* melalui *service* URL yang ditampilkan. Jika, tampilan sesuai dengan *testing* di lokal, maka proses *deployment* dengan Cloud Run telah berhasil.

Setelah proses *Profile Website deployment* berhasil, untuk membantu pemantauan dan penganalisisan infrastruktur *website*, dapat dilihat melalui dashboard Cloud Run, Cloud Logging, dan Cloud Monitoring pada Gambar 5.



Gambar 5 Dashboard, Cloud Logging, dan Cloud Monitoring pada *Profile Website* – Cloud Run

Pada gambar 5 menunjukkan bahwa *Profile Website* yang telah di-*deploy* dengan App Engine telah berfungsi dengan baik, tanpa adanya error. Apabila dilihat dari Dashboard Cloud Run, terdapat beberapa tab yang dapat ditinjau, tetapi pada tab *metrics* yang memuat 10 diagram menjadi hal yang utama untuk memantau website, karena memuat diagram untuk visualisasi *Request Count*, *Request Latencies*, *Container Instance Count*, *Container Startup Latency*, *Billable Container Instance Time*, *Container CPU Utiliza*tion, *Container Memory Utilization*, *Sent Bytes*, *Received Bytes*, dan *Max*

 ISSN:2302-8149
e-ISSN:2540-9719

Concurrent Request. Apabila dilihat dari Cloud Logging, terdapat 8 point Log Fields, dengan rincian pada Tabel 1 berikut:

Tabel 1 Cloud Logging – App Engine

Log Fields	Keterangan
Resource Type	Cloud Run Revision
Severity	Info
Log Name	run.googleapis.com/requests
Project ID	profile-website-1-crun
Service Name	profile-website
Revision Name	profile-website-00001-bad
Location	asia-southeast2
Configuration Name	profile-website

Sedangkan, apabila dilihat dari Cloud Monitoring, website telah bereaksi terhadap setiap permintaan yang diminta dan dapat ditinjau dari 5 diagram, yaitu diagram *Request*, *Network Traffic Sent*, *Network Traffic Received*, *Object Count*, dan *Object Size*.

Proses D*eployment* dengan App Engine

Untuk *Profile Website deployment* dengan App Engine diperlukan perancangan *Cloud Architecture* terlebih dahulu, supaya dapat mengetahui layanan dan alur proses yang sesuai pada Google Cloud Platform. Rancangan arsitekturnya dapat dilihat pada Gambar 6.

Gambar 6 *Cloud Architecture Profile Website Deployment* dengan App Engine

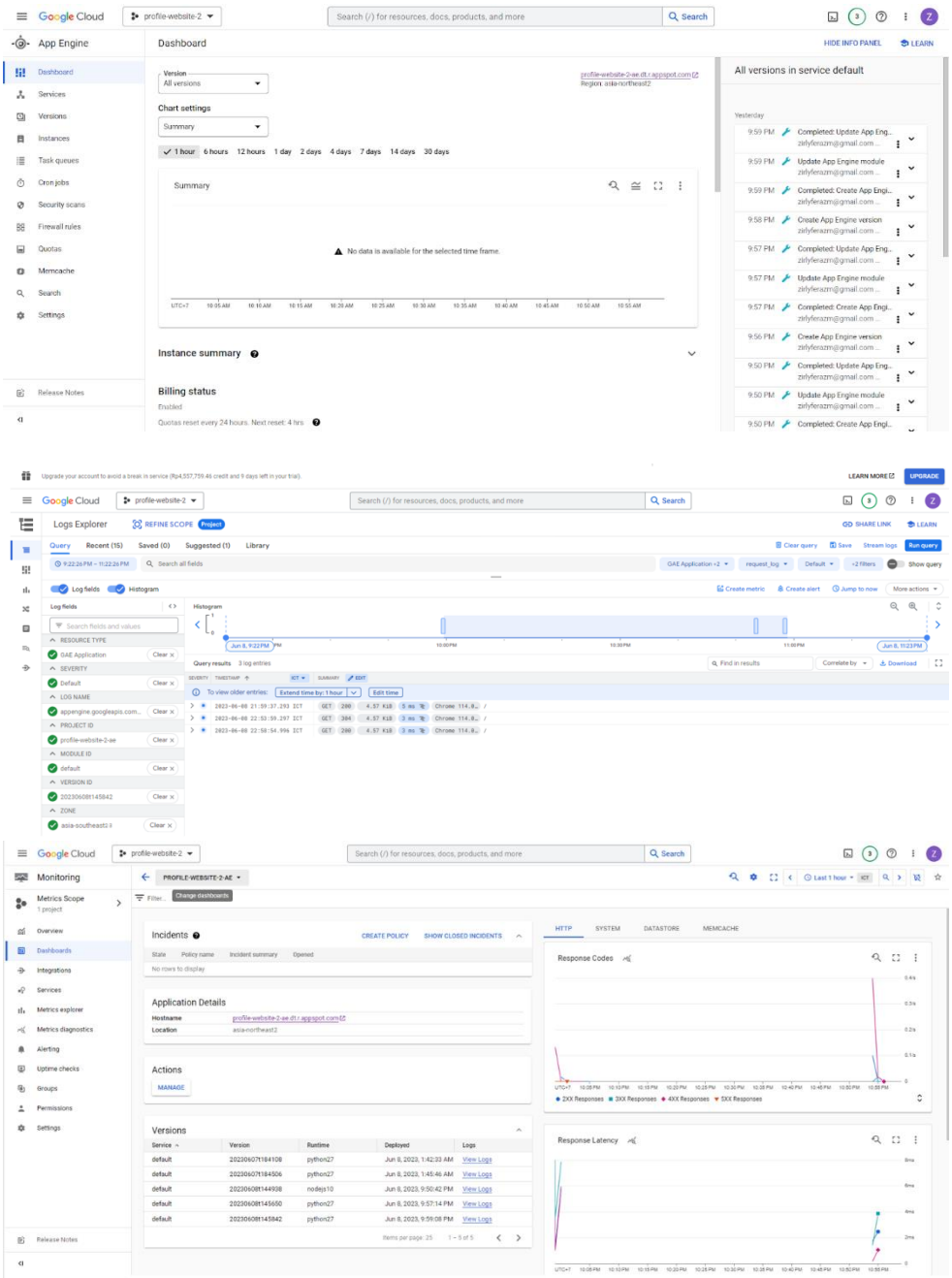
Pada Gambar 6 menunjukkan *Cloud Architecture* untuk proses *Profile Website Deployment* dengan App Engine, yaitu *users* dapat mengakses *Profile Website* dari *desktop* maupun *mobile* dengan mengakses *service URL* yang telah berhasil di-*deploy* dengan Cloud Run dan seluruh *assets* yang mendukung *website* tersebut tersimpan di Cloud Storage yang bersifat publik, sehingga ketika terdapat *users* yang mengakses, Cloud Storage dapat mendukung tampilan *website* dengan baik.
Terdapat beberapa tahapan pada proses *Profile Website deployment* dengan App Engine, yaitu:

1) Pada tampilan App Engine, klik "Create Application", memilih region untuk menempatkan *website* tersebut dan memilih *service account* yang akan digunakan di App Engine. Untuk *profile website*, penulis memilih "asia-southeast2" dan "App Engine default service account".
2) Membuka Cloud Shell Editor dan Terminal. Pada Terminal, penulis melakukan *clone* kode dari repositori GitHub yang berisi berkas HTML untuk Profile Websie, yaitu dengan instruksi:
   `git clone https://github.com/zirlyzkiyaa/profile-website`
   Namun, jika tidak dengan cara tersebut, dapat juga dengan membuat secara manual berkas-berkas yang dibutuhkan sebagai struktur dasar *project* di Cloud Editor.
3) Apabila proses *cloning* atau pembuatan struktur dasar *project* telah selesai, selanjutnya membuka Terminal pada Cloud Shell dan masukkan perintah berikut untuk masuk ke direktori *profile website*: `cd profile-website`
4) Membuat berkas app.yaml. berkas tersebut memuat konfigurasi yang memberi tahu App Engine cara memetakkan URL ke berkas statis *website*.
5) Untuk *deployment*, jalankan instruksi berikut pada terminal di Cloud Shell Terminal:
   `gcloud app deploy`

Jika proses *deployment* telah selesai, *Service* URL *Profile* dapat dilihat dengan menjalankan instruksi: `gcloud app browse`
Berikut contoh tautan yang diterima: `https://profile-website-2-ae.dt.r.appspot.com/`

6) Terakhir, akses *Profile Website* melalui *service* URL yang ditampilkan. Jika, tampilan sesuai dengan testing di lokal, maka proses *deployment* dengan App Engine telah berhasil.

Setelah *Profile Website* berhasil di-*deploy*, untuk membantu pemantauan dan penganalisisan infrastruktur website, dapat dilihat melalui Cloud Logging dan Cloud Monitoring pada Gambar 7.



Gambar 7 Dashboard, Cloud Logging, dan Cloud Monitoring pada *Profile Website* – App Engine

Gambar 7 menunjukkan bahwa *Profile Website* yang di-*deploy* dengan App Engine telah berfungsi dengan baik, tanpa adanya *error*. Apabila dilihat dari Dashboard App Engine, terdapat diagram *Summary* yang memuat visualisasi dari performa dan statu dari website yang telah di-*deploy* melalui

 ISSN:2302-8149
e-ISSN:2540-9719

App Engine. Selain itu, terdapat beberapa tabel juga, seperti untuk *Billing Status*, *Curent Load*, *Application Errors*, *Server Errors*, dan *Client Errors*. Apabila dilihat dari Cloud Logging, terdapat 7 point Log Fields, dengan rincian pada Tabel 2 berikut:

Tabel 2. Cloud Logging – App Engine

Log Fields	Keterangan
Resource Type	GAE Application
Severity	Default
Log Name	appengine.googleapis.com/request_log
Project ID	profile-website-2-ae
Module ID	Default
Revision Name	profile-website-00001-bad
Version ID	20230608t145842
Zone	asia-southeast2-3

Sedangkan, apabila dilihat dari Cloud Monitoring, *website* telah bereaksi terhadap setiap permintaan yang diminta, hal tersebut dapat dilihat pada ketiga diagram, yaitu diagram *Response Code*, *Response Latency*, dan *Response Count by Style*.

Analisis Perbedaan Cloud Run dan App Engine

1) Berdasarkan Definisi

Cloud Run merupakan sebagai salah satu layanan *serverless computing* yang dikelola oleh Google pada Google Cloud Platform untuk menjalankan maupun men-*deploy* suatu *container* aplikasi maupun mengimplementasikan beban kerja *stateful* dan *stateless* secara bersamaan dengan fleksibel. Sedangkan, App Engine merupakan sebuah platform dengan *serverless computing* untuk membantu *developer* dalam memproses pengembangan dan penghostingan aplikasi *website* dengan skala besar maupun Back-End API yang dikelola secara sepenuhnya oleh Google.

2) Berdasarkan Fitur

Pada Cloud Run dan App Engine memiliki beragam fitur pada masing-masing layanan, tetapi dapat dilihat perbedaannya melalui empat hal berikut:

a. Bahasa pemrograman dan *Framework*: Cloud Run mendukung berbagai bahasa pemrograman dan *framework*, seperti Node.js, Go, Java, Kotlin, Scala, Python, .Net, dan juga Docker. Sedangkan, App Engine mendukung beberapa bahasa pemrograman populer, seperti Node.js, Java, Ruby, C#, Go, Python, dan PHP. Berdasarkan hal tersebut, App Engine lebih terbatas pada bahasa pemrograman yang didukung oleh layanannya.
b. Penggunaan *Container*: Cloud Run menggunakan *container* sebagai unit dasar untuk menjalankan aplikasi. Sedangkan, App Engine dapat menjalankan aplikasi yang dikelola sepenuhnya tanpa perlu mengkonfigurasi *container*.
c. Skalabilitas: Cloud Run mendukung penskalaan otomatis yang cepat, karena dapat meningkatkan maupun menurunkan skala aplikasi berdasarkan *request* atau *traffic* yang diterima. App Engine juga dapat melakukan penskalaan otomatis, tetapi tidak secepat Cloud Run dalam menyesuiakan skala.
d. Fleksibilitas: Cloud Run merupakan layanan *serverless computing* dengan fleksibilitas yang tinggi, karena emungkinkan *developer* dapat menggunakan bahasa pemrograman, *library*, dan biner yang beragam. Sedangkan, App Engine lebih terbatas pada bahasa pemrograman dan *framework* yang didukung, sehingga meskipun dapat memberikan kenyamanan dalam menggunakannya karena sederhana, App Engine menjadi tidak memiliki fleksibilitas setinggi Cloud Run.

 ISSN:2302-8149
e-ISSN:2540-9719

3) Berdasarkan Proses *Deployment*

Untuk proses *Profile Website deployment* dengan Cloud Run dan App Engine memiliki caranya masing-masing pada setiap layanan, perbedaannya dapat dilihat pada tabel 3.

Tabel 3 Perbedaan Cloud Run dan App Engine Berdasarkan Proses *Deployment*

	Cloud Run	App Engine
Cloud Infrastructure	Harus membuat Container Image terlebih dahulu pada Cloud Build sebelum men-deploy dan pastikan terdapat Dockerfile.	Dapat men-deploy tanpa membuat Container Image pada Cloud Build dan pastikan terdapat app.yaml
Instruksi untuk deploy	`gcloud run deploy --image gcr.io/profile-website-1-crun/profile-website`	`gcloud app deploy`
Service URL	`https://profile-website-qbqpgm5xxq-et.a.run.app/`	`https://profile-website-2-ae.dt.r.appspot.com/`
Dashboard	Pada tab *metrics* yang memuat 10 diagram menjadi hal yang utama untuk memantau *website*, karena memuat diagram untuk visualisasi *Request Count*, *Request Latencies*, *Container Instance Count*, *Container Startup Latency*, *Billable Container Instance Time*, *Container CPU Utilization*, *Container Memory Utilization*, *Sent Bytes*, *Received Bytes*, dan *Max Concurrent Request*.	Terdapat diagram *Summary* yang memuat visualisasi dari performa dan status dari *website* yang telah di-*deploy* melalui App Engine. Selain itu, terdapat beberapa tabel juga, seperti untuk *Billing Status*, *Current Load*, *Application Errors*, *Server Errors*, dan *Client Errors*.
Cloud Logging	Terdapat 8 point *Log Fields*, yaitu *Resource Type*, *Severity*, *Log Name*, *Project ID*, *Service Name*, *Revision Name*, *Location*, dan *Configuration Name*	Terdapat 7 point *Log Fields*, yaitu *Resource Type*, *Severity*, *Log Name*, *Project ID*, *Module ID*, *Version ID*, dan *Zone*
Cloud Monitoring	Terdapat 5 diagram yang dapat ditinjau, yaitu diagram *Request*, *Network Traffic Sent*, *Network Traffic Received*, *Object Count*, dan *Object Size*.	Terdapat 3 diagram yang dapat ditinjau, yaitu diagram *Response Code*, *Response Latency*, dan *Response Count by Style*.

Dari Tabel 3, diketahui bahwa Cloud Run dan App Engine memiliki proses *deployment* yang cukup berbeda, hal tersebut dapat ditinjau dari *Cloud Infrastructure*, instruksi untuk *deploy*, *service* URL yang diterima, Dashboard, Cloud Logging, dan Cloud Monitoring dari masing-masing layanan, sebagai berikut:

a. Untuk *Cloud Infrastructure*, *deployment* dengan Cloud Run harus membuat Container Image terlebih dahulu menggunakan Dockerfile dan Cloud Build sebelum melakukan *deployment*. Sedangkan, dengan App Engine dapat melakukan *deployment* tanpa harus membuat Container Image terlebih dahulu pada Cloud Build, tetapi perlu membuat berkas app.yaml.
b. Untuk instruksi *deployment*:
   a) Dengan Cloud Run: `gcloud run deploy --image [nama_image]`
   b) Dengan App Engine: `gcloud app deploy`
c. Untuk *service* URL yang diterima:
   a) Dengan Cloud Run: `https://[service_name]-[random].a.run.app/`
   b) Dengan App Engine: `https://[service_name]-[random].r.appspot.com/`

 ISSN:2302-8149
e-ISSN:2540-9719

d. Untuk *Dashboard*, perbedaanya dapat dilihat pada tampilan *dashboard* di masing-masing layanan. Untuk *Dashboard* di Cloud Run, tampilannya lebih untuk memberikan informasi terkait performa dan penggunaan *resource* dari *instance* yang menjalankan aplikasi di Cloud Run, sehingga terdapat 10 diagram pada tab *metrics*, karena untuk memantau performa aplikasi yang telah di-*deploy* melalui Cloud Run. Sedangkan, *Dashboard* di App Engine, tampilannya lebih untuk gambaran ringkas mengenai performa dan status keseluruhan aplikasi yang telah di-*deploy* melalui App Engine.
e. Untuk Cloud Logging, terdapat beberapa perbedaan yang tertera pada *Log Fields*, perbedaan tersebut disebabkan karena perbedaan arsitektur dan cara kedua layanan *serverless computing* tersebut beroperasi. Pada Cloud Logging untuk Cloud Run, terdapat *Service Name* untuk menunjukkan unit utama yang digunakan untuk menjalankan aplikasi, *Revision Name* untuk mengidentifikasi revisi tertentu dari service tersebut, *Location* untuk menunjukkan lokasi ketika *service* dijalankan, dan *Configuration Name* untuk menunjukkan konfigurasi yang digunakan untuk mengatur service tersebut. Sedangkan pada Cloud Logging untuk App Engine, terdapat *Module ID* untuk menunjukkan unit aplikasi yang di-*deploy* dan mengidentifikasi modul tertentu, *Version ID* untuk menunjukkan versi aplikasi yang sedang berjalan, dan *Zone* untuk menunjukkan zona aplikasi tersebut dijalankan.
f. Untuk Cloud Monitoring, terdapat perbedaan yang disebabkan karena adanya perbedaan dalam metrik yang relevan untuk setiap layanan *serverless computing* tersebut dalam rentang waktu tertentu. Pada Cloud Monitoring untuk Cloud Run, terdapat 5 diagram, yaitu *Diagram Request* untuk menunjukkan jumlah permintaan yang diterima oleh service, *Diagram Network Traffic Sent* untuk menunjukkan jumlah data yang dikirim oleh service melalui jaringan, *Diagram Network Traffic Received* untuk menunjukkan jumlah data yang diterima oleh service melalui jaringan, *Diagram Object Count* untuk menunjukkan jumlah objek yang dikirim maupun diterima oleh aplikasi, dan *Diagram Object Size* untuk menunjukkan ukuran objek yang dikirim atau diterima oleh aplikasi. Sedangkan, Cloud Monitoring untuk App Engine, terdapat 3 diagram, yaitu *Diagram Response Code* untuk menunjukkan distribusi kode respons HTTP yang dihasilkan oleh aplikasi, *Diagram Response Latency* untuk menunjukkan waktu respons rata-rata dalam rentang waktu tertentu dari aplikasi yang telah di-*deploy* melalui App Engine, dan *Diagram Response Count by Style* untuk menunjukkan jumlah respons yang dihasilkan berdasarkan *response style*, seperti *cache hit*, *cache miss*, *cache expired*, dan sebagainya. Berdasarkan perbedaan pada Cloud Monitoring masing-masing layanan, diketahui bahwa Cloud Run berfokus pada aspek *traffic network* dan pengolahan objek, sedangkan App Engine berfokus pada aspek respons HTTP dan kinerja aplikasi.

4) Berdasarkan *Usability*

Cloud Run dan App Engine memiliki persamaan dalam memberikan pengalaman yang intuitif dan *user-friendly*, tetapi keduanya memiliki perbedaan yang cukup signifikan terkait *usability*. Cloud Run unggul karena fleksibilitasnya yang tinggi, karena memungkinkan *developer* untuk men-*deploy* containerized applications dengan bahasa pemrograman maupun framework apa pun. Hal tersebut sangat menguntungkan bagi *developer* yang lebih memilih kebebasan dalam menggunakan alat dan alur kerja pilihan mereka. Akan tetapi, dengan fleksibilitas yang tinggi, Cloud Run juga menjadi memerlukan adanya proses pengaturan dan konfigurasi yang mungkin lebih kompleks, seperti mengatur *container port*, kapasitas untuk memory, CPU, *request timeout*, dan *maximum concurrent request per instance*, serta *autoscaling*. Sedangkan, App Engine lebih sederhana untuk digunakan, terutama bagi *developer* yang telah terbiasa dengan bahasa pemrograman dan framework yang didukung pada layanan ini, seperti python, Java, dan Node.js. Namun, karena tidak terlalu fleksibel, *developer* menjadi terbatas pada bahasa pemrograman dan *framework* yang didukung. Berdasarkan hal tersebut, pemilihan antara *deployment* menggunakan Cloud Run dan App Engine perlu disesuaikan dengan kebutuhan. Apabila dibutuhkan fleksibilitas dan *workflow* yang dapat dipersonalisasi, maka Cloud Run menjadi pilihan yang tepat. Sebaliknya, apabila dibutuhkan pengaturan yang sederhana, ingin dapat di-*deploy* lebih cepat tanpa kompleksitas tinggi, maka App Engine menjadi pilihan yang baik.

 ISSN:2302-8149
e-ISSN:2540-9719

5 Kesimpulan

Dari hasil analisis perbedaan *serverless computing* pada layanan Cloud Run dan App Engine di Google Cloud Platform dengan cara melakukan *Profile Website deployment* yang dibantu dengan dua metode penelitian, yaitu metode kualitatif dan *Extreme Programming*, didapatkan bahwa *serverless computing* merupakan layanan komputasi daring yang dapat membantu pengguna atau pelanggan untuk fokus dalam mengembangkan fungsionalitas aplikasinya, tanpa perlu khawatir dengan infrastruktur, pengelolaan server, dan biaya yang besar. Cloud Run memungkinkan pengguna untuk menjalankan dan melakukan *deployment* suatu *container* aplikasi dengan fleksibilitas, baik untuk beban kerja *stateful* maupun *stateless*. Sedangkan App Engine adalah *serverless computing platform* yang membantu *developer* dalam pengembangan dan *hosting* aplikasi website serta Back-End API dengan skala besar yang infrastrukturnya dikelola sepenuhnya oleh Google. Meskipun keduanya memiliki beberapa fitur yang serupa, terdapat perbedaan pada bahasa pemrograman dan *framework*, penggunaan *container*, dan fleksibilitas. Terdapat enam hal yang menjadi pembeda pada proses *Profile Website deployment* pada kedua layanan tersebut, yaitu Cloud Infrastructure, intruksi *deploy*, service URL, Dashboard, Cloud Logging, dan Cloud Monitoring. Selain itu, terkait *usability*, pemilihan antara Cloud Run dan App Engine harus disesuaikan dengan kebutuhan, Cloud Run cocok untuk kebutuhan fleksibilitas yang dapat dipersonalisasi, sementara App Engine cocok untuk kebutuhan pengaturan yang sederhana dan *deployment* yang cepat tanpa kompleksitas tinggi. Berdasarkan hasil penelitian yang telah dilakukan, Cloud Run dan App Engine merupakan layanan *serverless computing* di Google Cloud Platform yang dapat dimanfaatkan untuk *deployment* suatu halaman web secara efisien berdasarkan kebutuhannya dan web tersebut dapat diakses secara langsung dari *service* URL yang diterima.

Referensi

[1] A. Sunyaev, "Cloud Computing," in *Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies*, A. Sunyaev, Ed., Cham: Springer International Publishing, 2020, pp. 195–236.

[2] A. S. Waranggani, "IDC: Pasar Cloud Computing Indonesia Bisa Tembus US$ 933 Juta Tahun 2023," 2023. [Online]. Available: https://www.cloudcomputing.id/berita/pasar-cloud-indonesia-bisa-meningkat-2023.

[3] D. L. Kusworo, A. A. Pratama, M. N. K. Fauzi, and M. Shafira, "Conception of An Independent Surveillance Authority in The Efforts to Protect Population Data," *Administrative and Environmental Law Review*, vol. 3, no. 1, pp. 9–26, 2022, doi: 10.25041/aelr.v3i1.2530.

[4] R. K. Putra, "Cloud-based Distributed Internet Measurement Platform," *Master Thesis*, Aalto University, Espoo, Finland, 2022. [Online]. Available: https://aaltodoc.aalto.fi/handle/123456789/116264

[5] I. Barokah and A. Asriyanik, "Analisis Perbandingan Serverless Computing Pada Google Cloud Platform," *Jurnal Teknlogi Informatika dan Komputer MH. Thamrin*, vol. 7, no. 2, pp. 169–187, 2021, doi: 10.37012/jtik.v7i2.662.

[6] E. Jonas *et al.*, "Cloud Programming Simplified: A Berkeley View on Serverless Computing," *arXiv:1902.03383*, 2019, doi: 10.48550/arXiv.1902.03383.

[7] H. Shafiei, A. Khonsari, and P. Mousavi, "Serverless Computing: A Survey of Opportunities, Challenges and Applications," *arXiv:1911.01296*, 2021, doi: 10.48550/arXiv.1911.01296.

[8] J. Schleier-Smith *et al.*, "What Serverless Computing Is and Should Become: The Next Phase of Cloud Computing," *Commun ACM*, vol. 64, no. 5, pp. 76–84, doi: 10.1145/3406011.

[9] Splunk, "Comparing Google's Serverless Offerings: Cloud Run, Cloud Functions, App Engine," 2020. [Online]. Available: https://www.splunk.com/en_us/blog/devops/gcp-serverless-comparison.html.

[10] Google Cloud Documentation, "Cloud Run," [Online]. Available: https://cloud.google.com/run?hl=id. https://cloud.google.com/run?hl=id.

[11] Google Cloud Documentation, "App Engine," [Online]. Available: https://cloud.google.com/appengine?hl=id.

[12] A. Abraham and J. Yang, "A Comparative Analysis of Performance and Usability on Serverless and Server-Based Google Cloud Services," in *Proceedings of the 2023 International Conference*

 ISSN:2302-8149
e-ISSN:2540-9719

on Advances in Computing Research (ACR'23), 2023, pp. 408–422. doi: 10.1007/978-3-031-33743-7_33.

[13] X. Niu, D. Kumanov, L. H. Hung, W. Lloyd, and K. Y. Yeung, "Leveraging Serverless Computing to Improve Performance for Sequence Comparison," in *ACM-BCB 2019 - Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, Association for Computing Machinery, Inc, 2019, pp. 683–687. doi: 10.1145/3307339.3343465.

[14] C. Bisri, H. Bancin, M. Aznur, M. A. Panjaitan, N. A. Suyadi, and S. Sitompul, "Analisis dan Desain Aplikasi Penjualan Bagi UMKM Berbasis Cloud Computing," *Jurnal Komputer Teknologi Informasi dan Sistem Informasi (JUKTISI)*, vol. 1, no. 3, pp. 185–191, 2023.

[15] S. Oktaviani, A. Priyanto, and C. Wiguna, "Implementasi Extreme Programming Pada Sistem Informasi Program Kreativitas Mahasiswa Berbasis Web," *JSiI: Jurnal Sistem Informasi*, vol. 9, no. 1, pp. 89–94, 2022, doi: 10.30656/jsii.v9i1.3666