Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 9, Nomor 1, Februari 2024, pp 491-503
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

Analisis Perbandingan Compute Engine dan Cloud Run sebagai lingkungan Pengembangan Aplikasi Web di Google Cloud Platform

Cahyo Bintang Kejora[1], Yeremia Alfa Susetyo[2]
[1,2]Program Studi Teknik Infomatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga, Jawa Tengah, Indonesia
Email: [1]672020303@student.uksw.edu, [2]yeremia.alfa@uksw.edu

Abstract

Cloud computing has become a significant issue in the world of Information and Communication Technology. A leading retail company in Indonesia, PT XYZ, faces challenges in choosing cloud computing services for their application development and operations. The company is faced with a choice between Compute Engine and Cloud Run on Google Cloud Platform. This research aims to analyze the comparison between the two services. By designing a web application which will then be deployed on the two cloud services, a comparative analysis of the applications deployed on the two services will then be made. Based on the test results and analysis carried out in this research, Cloud Run stands out in various aspects. From a cost perspective, the setup and performance of Cloud Run offers significant advantages compared to Compute Engine. Thus, this research concludes that Cloud Run is a more optimal choice for PT XYZ in the context of web application development, providing an efficient and economical solution.

Keywords*: web, python, flask, cloud computing, compute engine, cloud run*

Abstrak

Cloud computing telah menjadi isu yang signifikan dalam dunia Teknologi Informasi dan Komunikasi. Sebuah perusahaan retail terkemuka di Indonesia PT XYZ, menghadapi tantangan dalam memilih layanan cloud computing untuk pengembangan aplikasi dan operasional mereka, Perusahaan dihadapkan pada pilihan antara Compute Engine dan Cloud Run pada Google Cloud Platform. Penelitian ini bertujuan untuk menganalisis perbandingan antara kedua layanan tersebut. Dengan merancang sebuah aplikasi web yang kemudian akan di-deploy pada kedua layanan cloud tersebut yang kemudian dibuat analisis perbandingan dari aplikasi yang di-deploy pada kedua layanan. Berdasarkan hasil pengujian dan analisis yang dilakukan dalam penelitian ini, Cloud Run menonjol di berbagai aspek. Dari segi biaya, persiapan dan performa Cloud Run menawarkan keunggulan yang signifikan dibandingkan dengan Compute Engine. Dengan demikian, penelitian ini menyimpulkan bahwa Cloud Run merupakan pilihan yang lebih optimal untuk PT XYZ dalam konteks pengembangan aplikasi web, memberikan solusi yang efisien dan ekonomis.

Kata kunci*: web, python, flask, cloud computing, compute engine, cloud run*

1. PENDAHULUAN

Cloud computing, sebagai isu hangat dalam Teknologi Informasi dan Komunikasi, mencerminkan minat besar dari praktisi *Information and Communication Technology (ICT)* terhadap tren teknologi terbaru. Dengan memperbolehkan akses melalui internet ke sumber daya dan aplikasi, *cloud computing* melampaui hambatan infrastruktur *ICT* tradisional. Layanan *cloud* menawarkan kelebihan seperti kemampuan penyesuaian, efisiensi biaya, dan inovasi di berbagai lokasi[1]. Meningkatkan kecepatan pengembangan aplikasi bisa tercapai dengan memanfaatkan sumber daya *cloud* yang dapat disesuaikan

dengan kebutuhan dan memanfaatkan keuntungan *cloud computing* dalam berbagai aspek dalam pengembangan bisnis[2].

PT XYZ, sebuah perusahaan retail terkemuka di Indonesia, menghadapi tantangan besar dalam memilih layanan *cloud computing* yang sesuai dengan kebutuhan pengembangan aplikasi dan operasional mereka. Dalam menghadapi banyak pilihan pasar *cloud computing*, mereka harus mempertimbangkan aspek penting seperti skalabilitas, fleksibilitas, dan keamanan data. Pemilihan layanan *cloud* yang tepat adalah kunci untuk memastikan performa aplikasi yang optimal[3]. Dalam pengembangan aplikasi web menggunakan *cloud computing*, PT XYZ dihadapkan pada dua layanan utama yang ditawarkan oleh *Google Cloud Platform (GCP)* yaitu, *Compute Engine* yang merupakan *virtual machine* dan *Cloud Run* merupakan *serverless computing*[4].

Compute Engine adalah layanan infrastruktur virtual yang memberikan pengguna kontrol penuh atas lingkungan komputasi mereka. Dengan *Compute Engine*, pengguna dapat membuat, mengonfigurasi, dan mengelola mesin virtual sesuai kebutuhan mereka. Ini memungkinkan pengguna untuk meng-*install* dan menjalankan perangkat lunak apa pun, mengelola keamanan pada tingkat infrastruktur, dan menyesuaikan konfigurasi jaringan sesuai kebutuhan aplikasi mereka[5] . Di sisi lain, *Cloud Run* adalah layanan *serverless computing* yang disediakan oleh *GCP*. Dengan *Cloud Run*, pengembang dapat mengunggah dan menjalankan aplikasi dengan *container* tanpa harus memikirkan infrastruktur yang mendukungnya. Salah satu keunggulan *Cloud Run* adalah kemampuannya untuk menangani beban kerja yang tidak teratur dengan efisien [6]. Selanjutnya, akan dibandingkan layanan *cloud* dari *GCP*, yakni *Cloud Run* dan *Compute Engine*, untuk pengembangan aplikasi web di PT XYZ. Analisis kedua layanan tersebut akan membantu perusahaan memilih solusi yang paling cocok untuk lingkungan pengembangan aplikasi web yang optimal dan efisien sesuai kebutuhan mereka.

Dalam artikel ini, akan menyelidiki perbandingan antara *Compute Engine* dan *Cloud Run*, dengan tujuan mengidentifikasi kelebihan dan kelemahan masing-masing layanan. Penelitian ini akan memberikan pemahaman mendalam mengenai perbedaan dan keunggulan antara *Compute Engine* dan *Cloud Run* dalam konteks pengembangan aplikasi web di *Google Cloud Platform (GCP)*. Tujuannya adalah dalam membuat keputusan didasarkan pada informasi yang akurat dan sesuai dengan kebutuhan khusus aplikasi dan infrastruktur di PT XYZ. Penelitian ini bertujuan untuk mengeksplorasi sejauh mana *Compute Engine* dapat memenuhi persyaratan untuk pengembangan aplikasi web, serta sejauh mana *Cloud Run* dapat memberikan kebutuhan yang diperlukan untuk pengembangan aplikasi web. Dengan pemahaman yang mendalam mengenai kedua layanan tersebut, PT XYZ akan dapat membuat pilihan yang bijaksana saat memilih layanan *cloud computing* yang paling sesuai, dengan tujuan meningkatkan kinerja aplikasi mereka serta mengelola infrastruktur dengan efisien dan efektif.

2. METODOLOGI PENELITIAN

2.1. Tinjauan Pustaka

Pada sebuah penelitian yang berjudul "Analisis Perbandingan *Serverless Computing* Pada *Google Cloud Platform*" membahas tentang teknologi *serverless computing* yang telah menjadi tren utama dalam dunia *cloud computing*, memungkinkan pengembangan aplikasi dengan fokus pada logika inti tanpa perlu mengelola infrastruktur server. *Google Cloud Platform (GCP)* menawarkan berbagai *platform serverless*, termasuk *Cloud Functions, App Engine, Cloud Run*, dan *Google Kubernetes Engine*. Memahami perbedaan karakteristik masing-masing *platform* ini penting untuk memilih solusi yang sesuai dengan kebutuhan pengembangan aplikasi[7]. Penelitian ini memfokuskan pada perbandingan teknologi *serverless computing* dari *GCP*, termasuk *Cloud Functions, App Engine, Cloud Run,* dan *Google Kubernetes Engine*. Berbeda dengan penelitian sebelumnya yang hanya memfokuskan pada *Compute Engine* dan *Cloud Run*, metode penelitian ini melibatkan perancangan aplikasi web yang akan di-*deploy* ke kedua layanan *cloud* tersebut. Analisis perbandingan dilakukan untuk mengevaluasi perbedaan kinerja keduanya dan menentukan layanan mana yang lebih unggul dalam pengembangan aplikasi web.

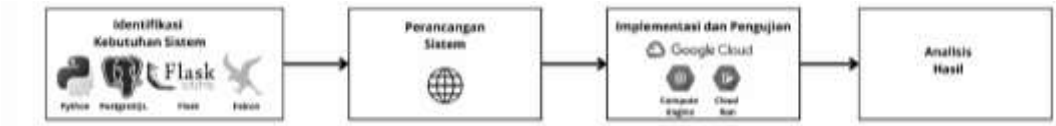
Analisis perbandingan kinerja dan kegunaan aplikasi web seluler dalam *container* pada layanan tanpa *server* dan berbasis server di *Google Cloud Platform (GCP)* oleh penelitian yang berjudul "*A Comparative Analysis of Performance and Usability on Serverless and Server-Based Google Cloud Services*". Tujuan dari penelitian ini adalah menganalisis perbandingan kinerja dan kegunaan aplikasi web seluler dalam *container* pada layanan tanpa *server* dan berbasis *server* di *GCP*. Aplikasi ini memungkinkan pengguna melihat lokasi dan perkiraan waktu kedatangan *bus* San Antonio VIA secara *real-time*. Aplikasi ini di-*deploy* di *Cloud Run* dan *App Engine* tanpa *server* serta *Compute Engine* berbasis *server*. Dalam menganalisis performa, layanan Pemantauan *GCP* digunakan untuk mengukur *matrix* performa termasuk *latency*, *throughput*, dan pemanfaatan *CPU*. Hasil penelitian menunjukkan bahwa *Cloud Run* mengungguli *App Engine* dan *Compute Engine* baik dari segi performa maupun kegunaan[8]. Penelitian sebelumnya membandingkan tiga layanan *cloud* dari *GCP*, sedangkan penelitian ini akan memfokuskan perbandingannya pada *Compute Engine* dan *Cloud Run* dari *GCP*, terutama untuk pengembangan aplikasi web. Penelitian ini bertujuan mendesain sebuah aplikasi web yang akan di-*deploy* pada kedua layanan *cloud GCP*, kemudian dianalisis untuk melihat perbedaan antara keduanya dalam pengembangan aplikasi web, Analisis tersebut akan menentukan layanan mana yang lebih unggul.

Penelitian yang berjudul *"Multi-Factor Performance Comparison of Amazon Web Services Elastic Compute Cluster and Google Cloud Platform Compute Engine"* membahas tentang perbandingan kinerja antara *Elastic Compute Cloud (EC2)* milik *Amazon Web Services (AWS)* dan *Compute Engine (CE)* milik *Google Cloud Platform (GCP)* menggunakan tiga *benchmark* : *STREAM, IOR,* dan *NPB-EP*. Hasilnya menunjukkan bahwa *EC2* memiliki kinerja lebih baik dalam hal *bandwidth* jika dibandingkan dengan *CE*. Keduanya menunjukkan skalabilitas dan keandalan yang baik dalam hal *bandwidth*, meskipun *CE* mengalami sedikit variasi selama uji coba

dua minggu. *CE* lebih unggul dibanding *EC2* dalam uji baca dan tulis *(IOR)* serta uji operasi per detik. Namun, *CE* menunjukkan variasi yang tinggi dalam uji baca dan tulis selama percobaan. Secara keseluruhan, masing-masing *platform* memiliki keunggulan dalam berbagai *benchmark* dan penelitian ini menyimpulkan bahwa *EC2* lebih dapat diandalkan secara umum[9]. Perbedaan utama dengan penelitian sebelumnya adalah bahwa penelitian ini akan membandingkan *Compute Engine* dan *Cloud Run* dari *GCP*, sementara penelitian sebelumnya membandingkan layanan *EC2* dari *AWS*. Metode penelitian ini melibatkan perancangan aplikasi web dan analisis perbandingan kinerja serta fitur dari kedua layanan. Hal ini bertujuan untuk menentukan layanan yang lebih unggul dalam pengembangan aplikasi web.

2.2. Metode Penelitian

Terdapat lima tahapan pada penelitian ini dimulai dari identifikasi kebutuhan sistem hingga analisis hasil. Skema tahapan penelitian dapat dilihat pada Gambar 1. Detail pada setiap tahapan penelitian adalah sebagai berikut:



Gambar 1. Skema Tahapan Penelitian

a) Identifikasi Kebutuhan
Pada tahap ini, identifikasi kebutuhan sistem yang akan dibuat untuk di-*deploy* pada kedua layanan, yakni *Compute Engine* dan *Cloud Run* pada *Google Cloud Platform (GCP)*. Menggunakan *Python* sebagai bahasa utama dalam perancangan sistem, kemudian menggunakan *framework Flask* untuk arsitektur perancangan web bagian antaramuka aplikasi dan *framework Falcon* untuk bagian *Application Programming Interface (API)* aplikasi dan menggunakan *database PostgreSQL* sebagai penyimpanan data dari aplikasi yang akan dirancang dan di-*deploy* pada layanan *cloud computing*.

b) Perancangan Sistem
Pada tahap ini, spesifikasi komponen sistem informasi secara menyeluruh, termasuk merancang tampilan antarmuka aplikasi yang akan dibuat. Perancangan untuk arsitektur tampilan antarmuka aplikasi menggunakan *framework Flask* dan untuk struktur *API* menggunakan *framework Falcon*.

c) Impelementasi dan Pengujian
Pada tahapan ini aplikasi yang telah dirancang kemudian akan di-*deploy* pada kedua layanan *GCP*, yakni *Compute Engine* dan *Cloud Run*. Kemudian akan dilakukan pengujian untuk menganalisis perbedaan yang ada. Pengujian menggunakan *tools Jmeter* dan *GTmetrix* dengan skema yang telah dirancang.

d) Analisis Hasil
Pada tahap akan dilakukan analisis perbandingan, hasil dari implementasi dan pengujian aplikasi pada *Compute Engine* dan *Cloud Run* dievaluasi secara mendalam. Dilakukan pembandingan biaya, persiapan, keamanan,

Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 9, Nomor 1, Februari 2024, pp 491-503
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

dan performa dari kedua layanan tersebut dalam menjalankan aplikasi yang telah dikembangkan. Hasil perbandingan ini akan menjadi dasar untuk menentukan layanan *cloud computing* mana yang lebih cocok dan memberikan performa terbaik sesuai dengan kebutuhan pengembangan aplikasi web di PT XYZ.

3. HASIL DAN PEMBAHASAN

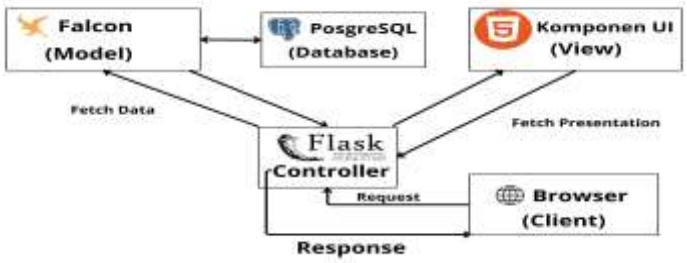
3.1. Identifikasi dan Analisis Kebutuhan Sistem

Pada tahap Identfikasi dan Analisis Kebutuhan Sistem, diperlukan suatu inovasi untuk mendukung semua proses yang diperlukan agar perusahaan dapat beroperasi secara optimal, perlu adanya aplikasi yang dilengkapi dengan layanan yang sesuai untuk memenuhi kebutuhan tersebut. *Cloud computing* telah menjadi solusi terkini dalam era ini sebagai pendukung utama kebutuhan perusahaan. Pemilihan layanan *cloud* harus dilakukan secara cermat, sehingga aplikasi yang diperlukan dapat berjalan dengan efisien di masa mendatang. Dibutuhkan evaluasi perbandingan antara layanan *Compute Engine* dan *Cloud Run* untuk menentukan pilihan terbaik dalam pengembangan aplikasi yang esensial bagi kelancaran operasional perusahaan.

3.2. Desain dan Perancangan sistem

Pada tahapan Desain dan Perancangan sistem, Aplikasi yang ditujukan untuk di-*deploy* pada *Compute Engine* dan *Cloud Run* adalah aplikasi *Point of Sales (POS)*. Aplikasi *POS* adalah sebuah merupakan aplikasi yang diciptakan untuk membantu proses transaksi jual beli barang dan merekam serta memproses transaksi penjualan[10]. Perancangan aplikasi menggunakan dua framework yaitu Falcon Framework sebagai backend dan Flask sebagai Frontend.

Perancangan bagian *backend* aplikasi di rancang *API* untuk terhubung ke aplikasi dengan menggunakan *framework Falcon*. *API* dirancang untuk terhubung ke *database*, *database* yang digunakan adalah *PostgreSQL*, kemudian *API* yang dibuat menjalankan beberapa proses kerja aplikasi seperti *login*, pengelolan *user*, kelola data barang dan transaksi. *API* sebagai perantara komunikasi antara *Client (Frontend)* dan *Database (Backend)*[11]. Perancangan arsitektur *backend* dapat dilihat pada Gambar 2.



Gambar 2. Diagram Perancangan API Backend dan Frontend

Falcon Framework berperan sebagai backend untuk membangun *API* yang menghubungkan antara *database* dan *frontend* dalam aplikasi web. *Falcon*

Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 9, Nomor 1, Februari 2024, pp 491-503
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

bertanggung jawab untuk memproses permintaan (*request*) yang diterima dari *frontend*. *Frontend* dibangun dengan menggunakan *framework Flask* yang berperan ganda sebagai *controller* dan *view*. *Flask* mengatur alur logika aplikasi serta menampilkan tampilan. Flask menggunakan fitur pengambilan data (*fetch data*) untuk mengambil informasi yang diperlukan dari server *Falcon*. Setelah mendapatkan data yang diminta, *Flask* kemudian menggunakan informasi tersebut untuk membangun tampilan yang sesuai untuk ditampilkan. *Flask* juga berfungsi untuk mengarahkan *routing* tampilan ke *API*, permintaan yang diterima dari browser ke fungsi-fungsi yang tepat dalam aplikasi berdasarkan *URL* yang diminta.

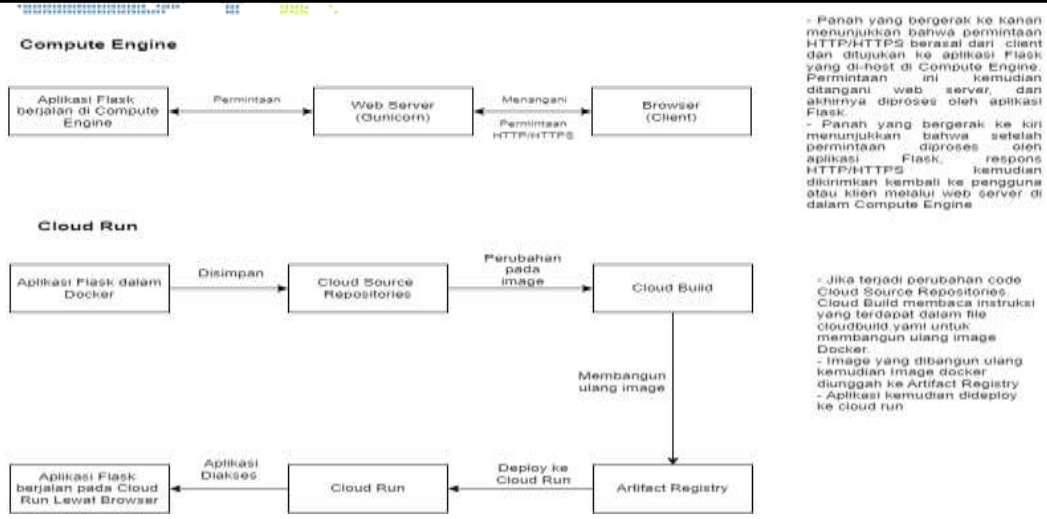
Komponen untuk *backend* akan dieksekusi pada satu *file* yaitu *"app.py", file* ini akan menjalankan seluruh proses yang yang membangun *API* dengan *Framework Falcon* yang menghubungkan *client* dan *database*. Berikut kode program untuk *URL backend* yang di buat pada *file "app.py"* dapat dilihat pada Kode Program 1.

```
...
# transaksi
transaksi = TransaksiResource()
app.add_route('/transaksi', transaksi)
app.add_route('/transaksi/{transaksi_id}', transaksi)
detail_transaksi_resource = DetailTransaksiResource()
# pembayaran
pembayaran = PembayaranResource()
app.add_route('/pembayaran', pembayaran)
...
app = falcon.App()
```

Pada Kode Program 1 setiap *resource* menangani jenis permintaan tertentu (seperti *GET* atau *POST*). *Routes* mendefinisikan *endpoint* yang terkait dengan setiap *resource*, dan *Falcon* memetakan permintaan *HTTP* ke *resource* yang sesuai berdasarkan *route* yang ditentukan. *Endpoint* akan mengorganisir dan menangani permintaan *HTTP* untuk berbagai entitas proses dalam API.

Perancangan untuk *frontend* aplikasi ini menggunakan *framework Flask* yang merupakan *microframework* untuk pengembangan aplikasi web dengan menggunakan bahasa pemograman *Python*[12]. Perbedaannya terdapat pada arsitektur, dimana perbedaannya terdapat pada *service Cloud Run* membutuhkan *Dockerfile* dan *cloudbuild.yaml* sementara untuk *compute engine* tidak memerlukan *file* tersebut untuk menjalankan aplikasi pada *Compute Engine*. *Service* yang akan di-*deploy* di *Cloud Run* memerlukan *file* tambahan berupa *Dockerfile* dan *cloudbuild.yaml*. *Docker* memungkinkan pembuatan dan pengemasan aplikasi ke dalam *container*[13]. *cloudbuild.yaml* berisi instruksi-instruksi yang diperlukan untuk mengonfigurasi lingkungan *build*, membangun *container*, dan *deploy* ke *Cloud Run*[14]. Saat *Cloud Build* menjalankan *cloudbuild.yaml*, itu menggunakan *Docker* sebagai *runtime* untuk memproses instruksi-instruksi dalam *file* tersebut. Saat *Cloud Build* menjalankan *cloudbuild.yaml*, itu menggunakan *Docker* sebagai *runtime* untuk memproses instruksi-instruksi dalam *file* tersebut. Untuk melihat lebih jelas arsitektur untuk kedua layanan *Compute Engine* dan *Cloud Run* disajikan pada Gambar 3.

Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 9, Nomor 1, Februari 2024, pp 491-503
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik



Gambar 3. Arsitektur Compute Engine dan Cloud Run

Perbedaan Arsitektur pengembangan aplikasi web dengan *Cloud Run* dan *Compute Engine* adalah *Cloud Run* memungkinkan menjalankan *container Docker* di lingkungan yang dikelola sepenuhnya oleh *GCP*. Dalam diagram ini, aplikasi *Flask* dibangun dalam *Docker*, diunggah ke *Artifact Registry*, dan kemudian di-*deploy* ke *Cloud Run*. *Cloud Run* akan mengambil *image Docker* dari *Artifact Registry* dan secara otomatis membuat *instance container* dari *image* tersebut rute atau proses tersebut dikendalikan oleh file *cloudbuild.yaml* berisi instruksi untuk mengendalikan lingkungan *build*. Aplikasi *Flask* kemudian dapat diakses sebagai layanan yang dikelola sepenuhnya di *Cloud Run,* kemudian aplikasi yang sudah di-*deploy* dapat diakses lewat browser. Sementara itu *Compute Engine* yang memungkinkan untuk menjalankan mesin virtual di infrastruktur *GCP*. Dalam diagram ini, aplikasi *Flask* di-*deploy* di dalam di *Compute Engine*. Aplikasi tersebut diakses melalui *web server (Gunicorn)* yang bertindak untuk mengelola permintaan *HTTP/HTTPS. Compute Engine* memproses permintaan ini dan mengirimkan respons kembali kepada *client* lewat browser.

```
api_server = "https://'URL - API'"  #url backend Framework falcon
...
@app.route("/produk", methods=['GET'])
def produk():
   session.pop('cart', None)
   session.pop('harga', 0)
   data = requests.get(f'{api_server}/produk')
   return render_template("produk.html", user=session["user"], products=data.json())
@app.route("/detail_produk", methods=['GET', 'POST'])
def detail_produk():
   pid = request.form['id']
   data = requests.get(f'{api_server}/detail_transaksi/{pid}')
   return render_template("detail_produk.html", user=session["user"], transaksi=data.json())
.....
if __name__ == "__main__":
   app.run(host="0.0.0.0", port=8080, debug=True)
```

Seperti yang di tunjukan Kode Program 2 adalah potongan kode pada Perancangan *Frontend* dengan *framework flask* yang merupakan bagian *server-side*

Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 9, Nomor 1, Februari 2024, pp 491-503
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

dari aplikasi web. Kode ini digunakan untuk mengatur permintaan yang diterima dari *frontend* atau sisi klien aplikasi ke *API backend* untuk menjalankan beberapa proses aplikasi.

3.3. Analisis Perbandingan

Pada tahap Analisis perbandingan akan di analisis perbedaan dari segi Persiapan *Deploy*, Biaya dan Performa Aplikasi pada saat sudah di-*deploy* pada *Compute Engine* dan *Cloud Run*. Berikut perbandingan dari segi persiapan *deploy* dan biaya aplikasi pada layanan *Compute Engine* dan *Cloud Run* disajikan pada Tabel 1.

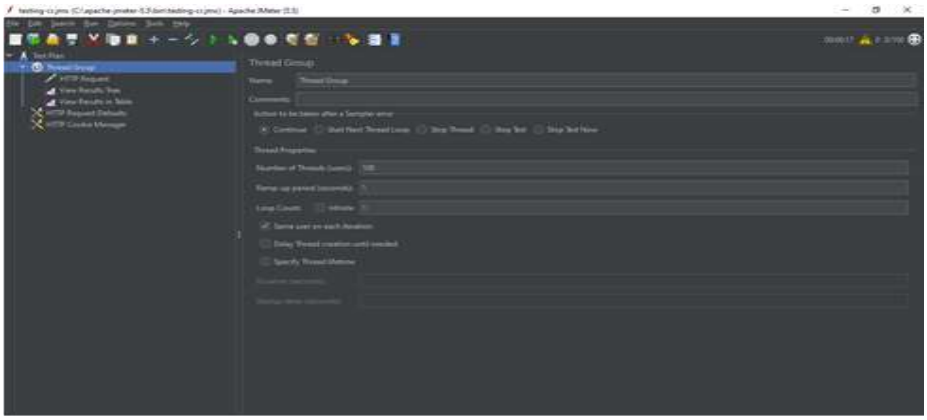
Tabel 1. Perbandingan persiapan dan biaya deploy

Persiapan	***Compute Engine***	***Cloud Run***
Persiapan Virtual Machine	Konfigurasi *Virtual Machine*.	Tidak diperlukan (bekerja dengan *container*).
Deploy Aplikasi	Atur server web *runtime*, dan depedensi didalam *Virtual Machine*.	*Deploy Container* ke *Cloud Run* melalui *Cloud Console* dan *Source Repositories* GCP.
Konfigurasi lingkungan	*Install* dan Konfigurasi *Runtime*, server web dan depedensi ke *Virtual Machine*.	Konfigurasi kedalam *Dockerfile* dan spesifikasi ke dalam *deployment* di *cloud console*.
Konfigurasi *Firewall* dan Jaringan	Mengatur *firewall* untuk Virtual konfigurasi alamat *IP* dan *domain*.	Tidak perlu, diatur otomatis oleh *Cloud Run*.
Keamanan	Terapkan keamanan dan konfigurasi izin akses.	Keamanan terkelola oleh *Cloud Run*, *HTTPS* diaktifkan otomatis.
Pemeliharaan	Konfigurasi pemantauan, *log* dan *update* berkala sistem dilakukan secara manual.	Pemantauan dan *log* diatur otomatis oleh *Cloud Run*.
Model biaya layanan	Berdasarkan permintaan	Berdasarkan permintaan
Fleksibelitas *Deployment*	Lebih fleksibel dengan kontrol manual atas konfigurasi lingkungan dan sumber daya.	Lebih fokus pada *container* dan terkelola.
Biaya Skala Nol	Biaya minimum VM selama *instance* berjalan.	Tidak ada biaya saat tidak ada permintaan.
Domain Custom	Memerlukan konfigurasi manual untuk mengatur sertifikat *SSL/TLS*.	Terintegrasi dengan otomatis menggunakan sertifikat yang dikelola oleh *Google*.

Pada penelitian terdahulu yang berjudul "Analisis Perbandingan *Serverless Computing* Pada *Google Cloud Platform*" memaparkan bahwa keunggulan *cloud run*, yakni wadah untuk produksi dalam hitungan detik karena dikemas dalam *container*, pengelolaan aplikasi dikelola oleh *cloud run* dan aplikasi terintegrasi dengan *Cloud Code, Cloud Build, Cloud Monitoring,* dan *Cloud Logging* yang memudahkan *user* untuk mengelola sebuah aplikasi yang akan di-*deploy*[7]. Perbandingan untuk persiapan *deploy* dan biaya ditunjukkan pada Tabel 2, terlihat bahwa *Cloud Run* lebih mudah digunakan untuk proses *deployment* aplikasi jika

dibandingkan dengan *Compute Engine*, tanpa memerlukan persiapan lebih kompleks yang diperlukan oleh *Compute Engine*. Beberapa konfigurasi yang harus diatur secara manual oleh *Compute Engine* otomatis ditangani oleh *Cloud Run*. Selain itu, pemeliharaan *Cloud Run* juga dikelola secara otomatis. *Cloud Run* unggul dalam segi efesiensi biaya, karena biaya diterapkan pada cloud run per-*request* tidak seperti pada *compute engine* yang setiap per waktu selama mesin virtual berjalan. Meskipun begitu, *Compute Engine* memiliki keunggulan dalam hal keamanan dan *firewall* untuk akses aplikasi serta pengaturan dependensi juga dapat dikelola langsung melalui *Compute Engine*.

Pada penelitian ini akan dilakukan Analisis perbandingan dari segi performa kinerja dengan menggunkan *Tools Jmeter* dan *GTmetrix*. *Jmeter* digunakan untuk menguji performa kinerja aplikasi yang telah di-*deploy* pada *Compute Engine* dan *Cloud Run*, pengujian dilakukan dengan tujuan bagaimana untuk melihat aplikasi berperilaku ketika beberapa *user* mengakses aplikasi secara bersamaan, dengan begitu akan terlihat layanan cloud mana yang lebih unggul dalam menangani performa tersebut[15]. Rincian pengaturan beban pengujian dapat dilihat dalam Gambar 4.



Gambar 4. Thread Grup

Kemudian dari hasil diberikan *Jmeter* akan di buat Hasil dari pengujian tersebut dibandingkan untuk mengetahui layanan mana yang dapat menangani kondisi simulasi dengan lebih baik. Berikut hasil pengujian dengan menggunakan *tools jmeter*, pengujian dengan tingkat beban disimulasikan 100, 500, dan 1000 user mengakses dalam waktu bersamaan di rangkum pada Tabel 2.

Tabel 2. Hasil Pengujian Jmeter

Platform	User	Response Berhasil (Kode : 200)	Latency	Sukses (Hasil Req: TRUE)
Compute Engine	100	100	5304113	100
	500	494	33687587	494
	1000	499	50019804	499
Cloud Run	100	100	2417454	100
	500	500	17487534	500
	1000	1000	42409848	1000

Hasil pengujian dengan *jmeter* memperoleh hasil sama dengan penelitian terdahulu yaitu penelitian yang berjudul *"A Comparative Analysis of Performance and Usability on Serverless and Server-Based Google Cloud Services"* yang mengukur dengan acuan matrix performa termasuk *latency, throughput*, dan pemanfaatan *CPU*, hasil penelitian menunjukkan bahwa *Cloud Run* unggul dari *Compute Engine* dan *App Engine* dari semua *matrix* pengujian yang dilakukan[8]. Hasil pengujian dengan *jmeter* menunjukkan *Cloud Run* unggul dalam pengujian yang dilakukan. Hasil pengujian dengan *jmeter* memperoleh hasil bahwa layanan *Cloud Run* unggul dalam performa kinerja dibandingkan dengan *Compute Engine* mulai dari terhubung ke *URL Latency* dan Sukses masuk *URL*. Berikut adalah penjelasan lebih rinci.

Terhubung ke *URL Default* (Response Kode: 200), *Cloud Run* mencapai tingkat keberhasilan 100% pada semua tingkat beban (100, 500, dan 1000). *Compute Engine* juga mencapai keberhasilan 100%, tetapi pada tingkat beban 1000, terdapat penurunan menjadi 499.

Latency (Waktu Respons), *Cloud Run* menunjukkan *latency* yang lebih rendah dibandingkan dengan *Compute Engine*. *Latency* yang lebih rendah menunjukkan bahwa *Cloud Run* dapat merespons permintaan dengan lebih cepat. Pada semua tingkat beban (100, 500, dan 1000), *Cloud Run* memiliki nilai latency yang lebih rendah dibandingkan dengan *Compute Engine*. Jumlah latency milik *cloud run* 62314836 rendah dibandingkan dengan *compute engine* yakni 89011404 dalam 3 kali simulasi percobaan dengan pengingkatan jumlah user mengakses.

Sukses Masuk ke *URL*, Cloud Run mencapai tingkat keberhasilan 100% pada semua tingkat beban. Compute Engine juga mencapai keberhasilan tinggi, tetapi terdapat penurunan keberhasilan pada tingkat beban 1000.

Analisis selanjutnya perbandingan dengan menggunakan *Tools GTmetrix*. *GTmetrix* adalah alat yang digunakan untuk menguji dan menganalisis kinerja situs web. Penggunaan *GTmetrix* dapat membantu dalam mengevaluasi performa situs web dalam konteks kecepatan muatan, skor kinerja, ukuran halaman, jumlah permintaan, dan rekomendasi optimasi. Data yang diperoleh dari *GTmetrix* dapat menjadi sumber informasi yang berharga untuk mendukung penelitian atau analisis tentang kinerja situs web dalam berbagai kondisi jaringan dan perangkat[16]. Berikut merupakan hasil pengujian aplikasi web yang diuji menggunakan *Gtmetrix* dapat dilihat pada Gambar 5.

Gambar 5. Hasil Pengujian Jmeter

Berdasarkan pengujian yang telah dilakukan, Hasil memperlihatkan bahwa *Cloud Run* lebih unggul dari *Compute Engine*. Untuk performa *Cloud Run* mendapatkan *grade* nilai A *Performance* 96% dengan *Structure* 100%, sementara *Compute Engine* mendapatkan *grade* B dengan *Performance* 83% dengan *Structure* 81%. Kemudian untuk matrix yang ada pada web *vitals*, yaitu Largest *Contentful Paint* diungguli oleh web yang di-*deploy* dengan menggunakan *Cloud Run*.

Dari kedua pengujian yang telah dilakukan, menggunakan *Jmeter* dan *GTmetrix*, *Cloud Run* mengungguli *Compute Engine* dari hasil pengujian yang telah dilakukan. Kinerja *Cloud Run* lebih stabil dan unggul dari pengujian yang telah dilakukan. Penyebab perbedaan kinerja ini dapat dijelaskan dengan karakteristik masing-masing *platform*:

a) *Cloud Run*, merupakan layanan yang berbasis *container*, memungkinkan untuk menjalankan aplikasi di dalam *container* yang dapat diatur dan dihentikan sesuai kebutuhan. Ini dapat memberikan elastisitas dan *response* yang cepat terhadap permintaan.
b) *Compute Engine*, merupakan *virtual* mesin yang harus diinisialisasi dan dihentikan secara manual. *Response* terhadap beban kerja dapat dipengaruhi oleh spesifikasi mesin virtual dan waktu yang dibutuhkan untuk memulai dan menghentikan mesin virtual tersebut.

Dengan demikian, *Cloud Run* lebih unggul dalam hal elastisitas, *latency*, dan *response* terhadap permintaan karena sifat *containerized*-nya, sementara *Compute Engine* memiliki karakteristik mesin virtual yang mungkin memerlukan waktu lebih lama untuk inisialisasi dan penyesuaian dengan beban kerja yang dinamis.

Dari analisis dan pengujian yang telah dilakukan, *Cloud Run* terbukti mengungguli *Compute Engine* dalam berbagai aspek yang diuji dan dibandingkan. *Cloud Run* menonjol dalam hal biaya, persiapan, dan performa. Oleh karena itu, layanan *Cloud Run* lebih disarankan untuk digunakan oleh pengembang di PT XYZ dalam pengembangan aplikasi web. Dari segi biaya, *Cloud Run* menawarkan solusi yang lebih ekonomis daripada *Compute Engine*. Selain itu, *Cloud Run* juga memenangkan persaingan dalam hal persiapan aplikasi, karena menyediakan layanan yang lebih terkelola dan otomatis. Misalnya, pemeliharaan aplikasi di *Cloud Run* diatur secara otomatis melalui *container*. Performa juga menjadi keunggulan *Cloud Run*, terutama dalam menangani jumlah pengguna yang besar atau beban aplikasi yang meningkat. *Cloud Run* menunjukkan kestabilan yang baik dalam situasi-situasi ini, menjadikannya pilihan yang lebih handal untuk aplikasi di PT XYZ. Secara keseluruhan, *Cloud Run* menawarkan solusi yang lebih unggul dan sangat dianjurkan untuk pengembangan aplikasi di PT XYZ. Hal juga diperkuat berdasarkan penelitian yang dilakukan oleh Rahman Foyzur menyimpulkan bahwa penggunaan *Cloud Run* lebih efesien dalam pengembangan untuk aplikasi dalam skala yang tidak terlalu luas, Karena pengelolaan aplikasi serta pemeliharaan dilakukan oleh *Cloud Run* serta biaya lebih rendah karena pengenaan biaya dilakukan pada saat mengakses aplikasi yang di-*deploy* di *Cloud Run*[17]. Layanan *Cloud Run* lebih disarankan karena unggul dalam berbagai aspek untuk pengembangan aplikasi web dibandingkan *Compute Engine*.

4. SIMPULAN

Berdasarkan desain dan perancangan aplikasi *Point Of Sales* yang telah dikembangkan serta pengujian yang telah dilakukan, dapat disimpulkan bahwa aplikasi ini lebih cocok untuk di-*deploy* di *Cloud Run*. Hal ini disebabkan oleh pola penggunaan aplikasi yang bersifat *on-demand*, di mana aplikasi dijalankan hanya saat diperlukan, dan tidak memerlukan tingkat kontrol dan kompleksitas yang tinggi. Selain itu, dari segi biaya, penggunaan *Cloud Run* tidak menimbulkan beban yang lebih besar jika dibandingkan dengan menggunakan *Compute Engine*. Disisi lain dalam hasil pengujian yang telah disimulasikan menggunakan tools *Jmeter, Cloud Run* tetap unggul dalam hal penanganan kondisi yang telah disimulasikan, terhubung ke *URL* Cloud Run mencapai tingkat keberhasilan 100% pada semua tingkat beban (100, 500, dan 1000) dengan *latency* lebih rendah dari *Compute Engine* yakni 62314836 sementara *Compute Engine latency* lebih besar yakni 89011404. *Compute Engine* juga mencapai keberhasilan 100%, tetapi pada tingkat beban 1000, terdapat penurunan menjadi 499. Dan pada pengujian yang dilakukan dengan *GTmetrix*, *Cloud Run* mendapatkan *grade* nilai A *Performance* 96% dengan *Structure* 100%, sementara *Compute Engine* mendapatkan *grade* B dengan *Performance* 83% dengan *Structure* 81%, Hasil menunjukkan bahwa *Cloud Run* lebih menggungguli *Compute Engine* dalam performa untuk penanganan kondisi yang disimulasikan.

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa jika fokus utama suatu perusahaan adalah pada skalabilitas otomatis, kemudahan *deployment*, efisiensi biaya yang terkait dengan permintaan serta penanganan dalam suatu kondisi tertentu, maka *Cloud Run* menjadi pilihan yang lebih baik daripada *Compute Engine*. Dikarenakan biaya yang lebih murah, persiapan aplikasi tidak lebih kompleks dibandingkan *Compute Engine* yang memerlukan konfigurasi tambahan dan *Cloud Run* juga tetap stabil dalam penanganan suatu kondisi yang dilakukan saat pengujian. Namun, jika perusahaan memiliki kebutuhan khusus yang memerlukan kontrol penuh atau aplikasinya memiliki tingkat kompleksitas yang tinggi, maka *Compute Engine* menjadi pilihan yang lebih sesuai dibandingkan dengan *Cloud Run*. Sebagai contoh aplikasi Sistem Manajemen Toko *(In-Store Management System)*, cocok menggunakan layanan *Compute Engine* dikarenakan aplikasi memerlukan kontrol penuh atas infrastruktur, aplikasi yang kompleks dan pola lalu lintas yang stabil, Sistem ini dirancang untuk membantu pengelola toko dalam mengelola operasional harian, termasuk pengelolaan stok, pemesanan produk, dan penjadwalan karyawan. Aplikasi ini juga dapat memberikan analisis data penjualan dan membantu dalam merencanakan strategi pemasaran dan promosi di tingkat toko.

DAFTAR PUSTAKA

[1] A. Ashari, H. Setiawan, J. Ilmu, F. Mipa, and U. G. Mada, "Kata kunci : ICT," *J. Sist. Inf.*, vol. 3, no. 2, pp. 336–345, 2011.
[2] S. Farizy and E. S. Eriana, *CLOUD COMPUTING = KOMPUTASI AWAN*, no. 1.
[3] F. Aslam, "Role of Cloud Computing for Big Data," no. September, 2023, doi: 10.5281/zenodo.8311108.
[4] H. Lee and S. Kumar, "Evaluation of Production Serverless Computing

Environments," *IEEE Int. Conf. Cloud Comput. CLOUD*, 2018, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8457830

[5] S. P. T. Krishnan, *Building Your Next Big Thing with Google Cloud Platform*. 2015. [Online]. Available: https://link.springer.com/book/10.1007/978-1-4842-1004-8

[6] J. Schleier-Smith *et al.*, "What serverless computing is and should become," *Commun. ACM*, vol. 64, no. 5, pp. 76–84, 2021, doi: 10.1145/3406011.

[7] P. T. Informatika, U. M. Sukabumi, A. Engine, and C. Run, "Analisis Perbandingan Serverless Computing Pada Google Cloud Platform," vol. 7, no. 2, pp. 169–187, 2021.

[8] S. G. C. Services, A. Abraham, and J. Yang, "A Comparative Analysis of Performance and Usability on Serverless and A Comparative Analysis of Performance and Usability on Serverless and Server-Based Google Cloud Services," no. May, 2023, doi: 10.1007/978-3-031-33743-7.

[9] I. A. Board, A. Editors, and E. R. Board, "Interna tional Journal of Cloud Applications and Computing," vol. 10, no. 3, 2020.

[10] S. C. Cahyodi and R. W. Arifin, "Sistem Informasi Point Of Sales Berbasis Web Pada Colony Amaranta Bekasi," *Inf. Syst. Educ. Prof.*, vol. 1, no. 2, pp. 189–204, 2017.

[11] Falcon Contributors, "The Falcon Web Framework," *Falcon*, 2023. https://falcon.readthedocs.io/en/stable/

[12] Graciela Fausten Novindri and P. Ocsa Nugraha Saian, "Implementasi Flask Pada Sistem Penentuan Minimal Order Untuk Tiap Item Barang Di Distribution Center Pada Pt Xyz Berbasis Website," *J. Mnemon.*, vol. 5, no. 2, pp. 81–85, 2022, doi: 10.36040/mnemonic.v5i2.4670.

[13] S. Dwiyatno, E. Rachmat, A. P. Sari, and O. Gustiawan, "Implementasi Virtualisasi Server Berbasis Docker Container," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 7, no. 2, pp. 165–175, 2020, doi: 10.30656/prosisko.v7i2.2520.

[14] Google Cloud Platform, "Create a build configuration file." https://cloud.google.com/build/docs/build-config-file-schema

[15] D. I. Permatasari, "Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 1, p. 135, 2020, doi: 10.26418/justin.v8i1.34452.

[16] S. A. Arni, D. C. Mongkau, and A. Berelaku, "Analisis Performa Website Menggunakan GTMetrix," *J. Minfo Polgan*, vol. 12, no. 1, pp. 857–861, 2023, doi: 10.33395/jmp.v12i1.12518.

[17] F. Rahman, "Serverless Cloud Computing: A Comparative Analysis of Performance, Cost, and Developer Experiences in Container-Level Services," 2023.