ANALISIS PERBANDINGAN PERFORMA *APP ENGINE* DAN *COMPUTE ENGINE* PADA *GOOGLE CLOUD PLATFORM* DALAM MEMPREDIKSI PENYAKIT MATA DENGAN MODEL CNN

I Putu Tosan Krisna Wiranata, A A Istri Ita Paramitha, I Putu Satwika
Sistem Informasi, Primakara University Teknik Informatika, Primakara University
Jalan Tukad Badung No.135 Denpasar, Indonesia
tosankrisna@gmail.com

ABSTRAK

Menurut data Perhimpunan Dokter Spesialis Mata Indonesia (PERDAMI) menyatakan bahwa penyebab kebutaan tertinggi di Indonesia disebabkan oleh penyakit katarak. Seiring berjalannya waktu yang ditandai dengan kemajuan teknologi, muncul suatu cara untuk memprediksi penyakit mata yang disebut dengan *machine learning*. Model *machine learning* yang telah dibuat terlebih dahulu harus melalui tahap *deployment*. Tahapan *deployment* tersebut membutuhkan teknologi lain yang disebut *cloud computing*. *Google Cloud Platform* merupakan salah satu *cloud computing platform* yang didalamnya terdapat *App Engine* dan *Compute Engine* yang dapat dimanfaatkan untuk proses *deployment* model *machine learning*. Penelitian ini menggunakan metode komparasi yaitu membandingkan performa *App Engine* dan *Compute Engine* yang sebelumnya dilakukan pengukuran dengan menggunakan teknik *load testing* untuk mendapatkan performa yang akan dibandingkan. Perbandingan performa dilakukan dengan mengukur *response time, standard deviation, error rate, throughput* dan *CPU Utilization* pada kedua layanan tersebut. Kesimpulan yang didapatkan dari penelitian ini menunjukan bahwa *Compute Engine* lebih unggul dibandingkan *App Engine* dari segi *response time, standard deviation, throughput* dan *CPU Utilization*. Sedangkan pada *error rate* performa keduanya sama yaitu 0%.

Kata kunci: *Cloud Computing, Google Cloud Platform, App Engine, Compute Engine, Load Testing*.

1. PENDAHULUAN

Menurut data Perhimpunan Dokter Spesialis Mata Indonesia (PERDAMI) menyatakan bahwa penyebab kebutaan tertinggi di Indonesia disebabkan oleh penyakit katarak sebesar 0,78%. Berdasarkan data tersebut, Indonesia menjadi negara pertama dengan jumlah penderita kebutaan tertinggi di wilayah Asia Tenggara sebesar 3 juta orang (1,5% populasi penduduk) [1]. Kota Denpasar sebagai ibu kota Provinsi Bali juga memiliki jumlah penderita penyakit katarak yang tidak sedikit. Jumlah penderita penyakit katarak di Kota Denpasar berdasarkan data Laporan Tahunan Dinas Kesehatan Kota Denpasar tahun 2020 mencapai sebanyak 1.022 kasus [2].

Perkembangan teknologi dalam beberapa tahun terakhir berjalan dengan sangat cepat, diantara yang termasuk adalah *machine learning*, yaitu teknologi dalam bidang kecerdasan buatan (AI) yang memungkinkan mesin untuk belajar secara mandiri [3]. *Machine learning* dapat digunakan untuk memecahkan berbagai permasalahan mulai dari pengenalan sistem, robotika, maupun memprediksi gambar seperti katarak [4]. Hasil akhir dari *machine learning* disebut dengan model. Model *machine learning* terlebih dahulu harus melalui tahapan *deployment* sebelum nantinya dikoneksikan dengan suatu aplikasi atau website melalui API (*application programming interface)*. *Cloud computing* merupakan teknologi yang dapat digunakan untuk *deploy* model tersebut [5]. Sebelumnya terdapat penelitian yang berjudul "Implementasi Teknologi *Cloud Computing* Pada Sistem Transportasi Angkutan Umum Kota Tasikmalaya" yang menunjukan *cloud computing* efektif dari segi biaya [6]. *Google Cloud Platform* (GCP) merupakan salah satu penyedia *cloud computing* diantara beberapa *service* provider lainnya [7].

Sebelum melakukan proses *deployment* model *machine learning*, terlebih dahulu dilakukan pemilihan layanan yang akan digunakan pada proses *deployment*. Dalam proses *deployment* pada GCP, terdapat dua tipe layanan yang dapat digunakan yaitu *highly managed service* dan *minimally managed service*. Masing-masing jenis layanan memiliki kelebihan dan kekurangannya tersendiri. *Highly managed service* memiliki kelebihan yaitu pengguna dapat fokus mengembangkan aplikasinya tanpa menghiraukan konfigurasi *server*, akan tetapi memiliki kekurangan yaitu tidak support terhadap GPU/TPU. Layanan *minimally managed service* memiliki kelebihan yaitu support terhadap GPU/TPU, akan tetapi memiliki kekurangan yaitu mengharuskan penggunanya untuk melakukan konfigurasi *server* secara manual. Layanan GCP yang termasuk *highly managed service* yaitu *App Engine* dan layanan yang termasuk *minimally managed service* yaitu *Compute Engine*.

Sebelumnya terdapat penelitian serupa yang berjudul "Analisis Perbandingan *Serverless Computing* Pada *Google Cloud Platform*" [8], akan tetapi pada penelitian tersebut hanya membandingkan layanan secara kualitatif dari segi karakteristiknya dan belum membandingkan dari segi performa layanan. Selain itu, terdapat juga penelitian yang berjudul

"*Serverless Machine Learning* sebagai Solusi *Deployment* untuk *Prototipe Sign Language Recognition App* Menggunakan *Google Cloud Platform*" [9], pada penelitian tersebut telah dilakukan perbandingan dari segi performa layanan, akan tetapi jenis layanan yang dilakukan perbandingan hanya *highly managed service* yaitu *App Engine* dan *Cloud Functions* sedangkan layanan *minimally managed service* belum dilakukan perbandingan dari segi performanya.

Berdasarkan uraian permasalahan yang telah dijelaskan, maka peneliti tertarik melakukan penelitian terhadap perbandingan performa antara *App Engine* dan *Compute Engine* dalam hal memprediksi penyakit mata. Tolak ukur dari keberhasilan penelitian ini adalah dapat memaparkan hasil perbandingan performa antara kedua layanan pada skenario yang dilakukan dan dapat memberikan kesimpulan pada hasil perbandingan tersebut.

2. TINJAUAN PUSTAKA

2.1. *Cloud Computing*

Cloud computing atau yang kerap disebut komputasi awan merupakan suatu teknik komputasi yang menggunakan *shared computer processing* atau sumber daya pemrosesan komputer bersama serta data yang menuju ke komputer maupun perangkat yang lain akan dikirimkan ke pengguna sesuai permintaan [10] Keuntungan dari penggunaan *cloud computing* yaitu berupa skalabilitas, fleksibilitas, dan keamanan. Skalabilitas *cloud computing* berupa penyimpanan, *bandwith* dan kebutuhan lainnya yang sudah disediakan serta dapat disesuaikan dengan kebutuhan pengguna sehingga dapat menghemat biaya pengeluaran. *Cloud computing* juga menawarkan fleksibilitas yaitu pengguna dapat mengakses layanan cloud dimanapun dan kapanpun. Serta yang paling penting adalah aspek keamanan yang ditawarkannya. Adapun karakteristik *cloud computing* terbagi menjadi lima yaitu *on-demand self service, broadband network access, resource pooling, rapid elasticity,* dan *measured service.*

2.2. *Performance Testing*

Performance testing adalah proses pengujian perangkat lunak untuk mengukur suatu kecepatan atau waktu respon sistem maupun penggunaan sumber daya perangkat lunak dibawah beban kerja tertentu [11]. Terdapat tiga fokus performance testing yaitu sebagai berikut [12]:

a. *Speed,* bertujuan untuk menguji kecepatan respon suatu sistem dalam menerima request yang terjadi.
b. *Scalability,* bertujuan untuk menguji beban yang bisa dikendalikan oleh suatu sistem.
c. *Stability,* bertujuan untuk menguji kestabilan suatu sistem dalam mengatasi suatu kondisi.

2.3. *Load Testing*

Performance testing memiliki beberapa teknik yang dapat digunakan untuk mengukur performa suatu sistem, salah satunya adalah *load testing*. *Load testing* merupakan suatu teknik performance testing dimana suatu respon dari sistem diuji dan diukur dalam berbagai kondisi *load* [13]. *Load testing* dapat membantu pengguna untuk mengetahui perilaku suatu sistem dalam menangani beban

2.4. Penelitian Terdahulu (*State of the Art*)

Terdapat beberapa penelitian terdahulu yang menjadi referensi dalam melakukan penelitian ini. Adapun penelitian terdahulu tersebut sebagai berikut:

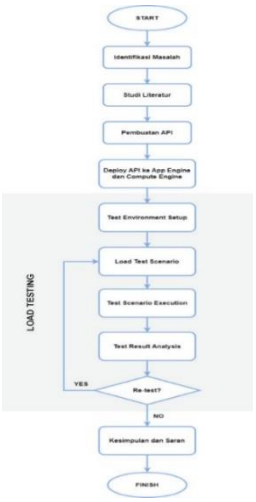
a. Penelitian berjudul "Analisis Perbandingan Performa *Web Server* Docker Swarm dengan Kubernetes Cluster" oleh Stefanus Eko Prasetyo dan Yulfan Salimin [14], pada penelitian tersebut Kubernetes Cluster memberikan performa yang lebih cepat dari Docker Swarm.
b. Penelitian berjudul "Pengukuran *Throughput Load Testing* Menggunakan *Test Case Sampling Gorilla Testing*" oleh Desy Intan Permatasari [15], Budi Santoso, Nadia Ningtias, dkk. Hasil yang didapatkan yaitu server memiliki kinerja yang baik pada login & register.
c. Penelitian berjudul "Performa *Microframework* PHP Pada REST API Menggunakan Metode *Load Testing*" oleh Indra Yatini, Wiwiek Nurwiyati, dan Khairul Anam [16]. Hasil yang didapatkan yaitu setelah dijalankan pengujian, didapatkan bahwa performa tertinggi diperoleh oleh Phalconmicro.
d. Penelitian berjudul "Serverless Machine Learning sebagai Solusi Deployment untuk Prototipe Sign Language Recognition App Menggunakan Google Cloud Platform" oleh Nur Ardli Rachmat Saputra [17] yang didapatkan hasil kualitas *latency* dari *App Engine* jauh mengungguli *Cloud Function* dengan hanya sedikit perbedaan *memory usage*.
e. Penelitian berjudul "Analisis Perbandingan Kinerja *Framework* Codeigniter Dengan Express.Js Pada Server RESTful API" oleh Luky Maulana, Kamal Prihan-dani, dan Adhi Rizal [18]. Hasil yang didapatkan yaitu r*esponse time* Express.Js lebih cepat dari Codeigniter, Codeigniter unggul di *CPU* dan *Memory Utilization*.

Berdasarkan dari referensi penelitian yang sudah dilakukan sebelumnya, belum terdapat penelitian lain yang mengukur performa dua jenis layanan *Google Cloud Platfrom* yaitu *App Engine* dan *Compute Engine* dalam melakukan prediksi suatu penyakit, sehingga terdapat keterbaruan dalam penelitian ini dibandingkan penelitian sejenis lainnya.

3. METODE PENELITIAN

Penelitian ini menggunakan metode komparasi atau biasa disebut sebagai metode perbandingan merupakan sebuah metode yang bertujuan untuk membandingkan satu atau lebih variabel pada dua atau lebih sampel yang berbeda [19]. Selain itu dilakukan juga teknik observasi dengan mengamati langsung

perbedaan nilai proses pengujian. Tahapan-tahapan yang dilakukan dapat dilihat pada gambar 1.



Gambar 1. Tahapan penelitian

Adapun tahapan-tahapan yang dilakukan pada gambar 1 adalah sebagai berikut:

a. Identifikasi masalah yaitu berdasarkan perbedaan jenis *App Engine* dan *Compute Engine* serta penelitian serupa yang hanya membandingkan performa layanan *highly managed service* dan belum membandingkan antara layanan *highly managed service* dengan *minimally managed service*. Sehingga dibutuhkan suatu perbandingan untuk mengetahui kelebihan dan kekurangan layanan dari segi performanya.
b. Studi literatur yaitu tahapan pengumpulan informasi dan juga mempelajari teori-teori yang berhubungan dengan penelitian yang akan dilakukan.
c. Pembuatan API yaitu dilakukan dengan menggunakan *framework* Flask. API tersebut akan digunakan pada proses pengujian performa nantinya. API yang dibuat memiliki dua buah *method* yaitu *POST* dan *GET*.
d. *Deploy* API ke *App Engine* dan *Compute Engine* yaitu bertujuan agar hasil pengujian bersumber dari *App Engine* dan *Compute Engine* itu sendiri. Adapun spesifikasi *server* yang digunakan sebagai berikut:

Tabel 1. Spesifikasi *server* pertama

Spesifikasi	***App Engine***	***Compute Engine***	
		Non GPU	**GPU**
CPU	2 vCPU	2 vCPU	2 vCPU
RAM	7,5 GB	7,5 GB	7,5 GB
Storage	50 GB	50 GB	50 GB
GPU	-	-	Nvidia T4

Tabel 2. Spesifikasi *server* kedua

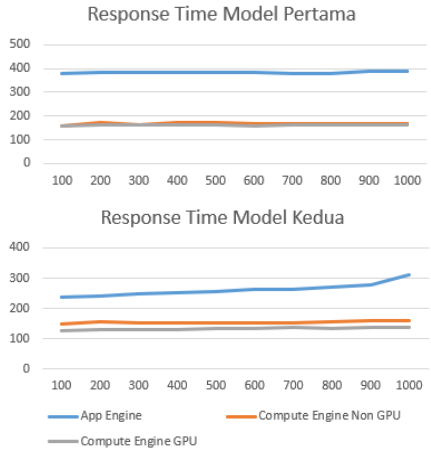
Spesifikasi	***App Engine***	***Compute Engine***	
		Non GPU	**GPU**
CPU	4 vCPU	4 vCPU	4 vCPU
RAM	15 GB	15 GB	15 GB
Storage	100 GB	100 GB	100 GB
GPU	-	-	Nvidia T4

e. *Test environtment setup* yaitu proses persiapan sebelum pengujian dilaksanakan seperti instalasi *tools testing* yaitu JMeter.
f. *Load test scenario* merupakan proses menentukan skenario-skenario pengujian yang akan dilakukan. Penentuan skenario-skenario tersebut bertujuan untuk mendapatkan hasil pengujian yang sesuai dengan harapan. Pengujian dilakukan dengan skenario mulai dari 100-1000 kali pengujian terhadap masing-masing *method*. Model yang digunakan berjumlah 2 buah model, dimana keduanya menggunakan jenis model CNN *(Convolutional Neural Network)*.
g. *Test scenario execution yaitu p*roses pengujian akan mengikuti skenario-skenario yang sudah dibuat pada tahapan sebelumnya.
h. *Test result analysis* yaitu bertujuan untuk memastikan hasil pengujian sudah sesuai dengan harapan. Pada tahap ini, dilakukan analisis pada hasil pengujian *response time, standard deviation, error rate, throughput* dan *CPU Utilization*.
i. *Re-test* yaitu tahapan untuk mengulangi *testing* apabila terdapat kesalahan pada langkah sebelumnya.
j. Kesimpulan dan saran, yaitu proses mengambil kesimpulan dan saran dari hasil pengujian yang telah dilakukan.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengujian *Response Time Server* Pertama *Method* POST

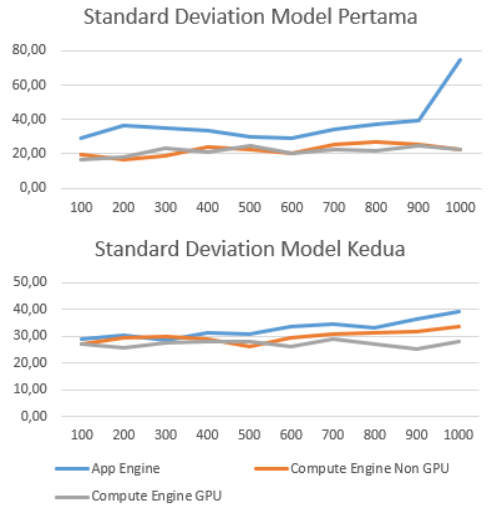
Berdasarkan hasil pengujian model pertama yang telah dilakukan, rata-rata hasil pengujian keseluruhan dari *App Engine* yaitu 383,2 *milisecond*, kemudian hasil pengujian *Compute Engine* Non GPU yaitu 167,3 *milisecond*, dan *Compute Engine* GPU sebesar 161,6 *milisecond*. Jadi pada pengujian model pertama ini *Compute Engine* GPU menjadi yang terbaik. Selanjutnya berdasarkan hasil pengujian model kedua, rata-rata hasil pengujian keseluruhan dari *App Engine* yaitu 263,6 *milisecond*, *Compute Engine* Non GPU yaitu 149,5 *milisecond*, dan *Compute Engine* GPU sebesar 135,6 *milisecond*. Jadi pada pengujian model kedua ini *Compute Engine* GPU menjadi yang terbaik.



Gambar 2. Pengujian *response time POST*

4.2. Hasil Pengujian *Standard Deviation Server* Pertama *Method* POST

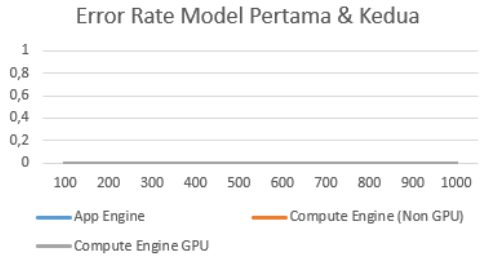
Pada hasil pengujian model pertama, rata-rata keseluruhan *standard deviation App Engine* memiliki nilai sebesar 37,897 sedangkan *Compute Engine Non GPU* memiliki nilai sebesar 22,362 dan *Compute Engine GPU* memiliki nilai sebesar 21,621. Jadi pada pengujian model pertama ini *Compute Engine GPU* menjadi yang terbaik. Selanjutnya berdasarkan hasil pengujian model kedua rata-rata keseluruhan *standard deviation App Engine* memiliki nilai sebesar 32,645 sedangkan *Compute Engine Non GPU* memiliki nilai sebesar 29,729 dan *Compute Engine GPU* memiliki nilai sebesar 26,692. Jadi pada pengujian model kedua ini *Compute Engine GPU* menjadi yang terbaik.



Gambar 3. Pengujian *standard deviation POST*

4.3. Hasil Pengujian *Error Rate Server* Pertama *Method* POST

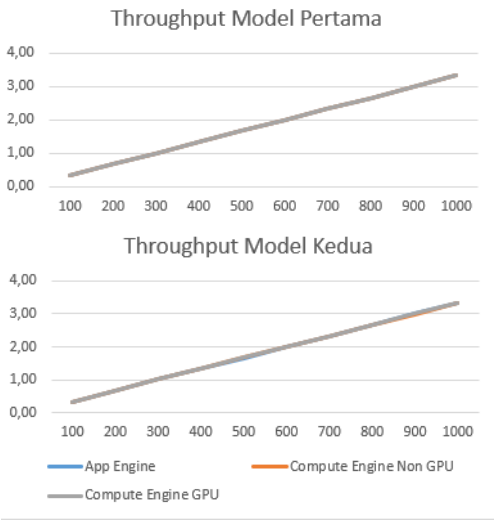
Berdasarkan hasil pengujian yang telah dilakukan kepada ketiga layanan yaitu *App Engine, Compute Engine Non GPU,* dan *Compute Engine GPU*, terlihat bahwa semua layanan tersebut dapat menjalankan seluruh permintaan (*request*) pengguna sehingga tidak terjadi *error* pada *server*. Jadi pada pengujian ini performa ketiga layanan tersebut pada kedua model dapat dikatakan setara.



Gambar 4. Pengujian *error rate POST*

4.4. Hasil Pengujian *Throughput Server* Pertama *Method* POST

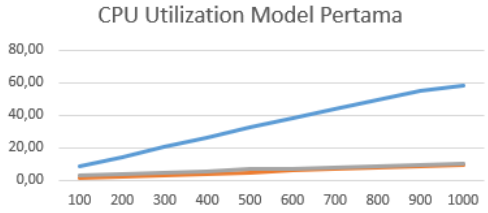
Pada pengujian model pertama, rata-rata *throughput* keseluruhan pengujian *App Engine* memiliki yaitu 1,834 sedangkan *Compute Engine Non GPU* memiliki rata-rata yaitu 1,835 dan *Compute Engine GPU* memiliki rata-rata yaitu 1,837. Jadi pada pengujian model pertama ini *Compute Engine GPU* menjadi yang terbaik. Pada pengujian model kedua rata-rata *throughput* keseluruhan pengujian *App Engine* yaitu 1,835 sedangkan *Compute Engine Non GPU* yaitu 1,834 dan *Compute Engine GPU* yaitu 1,837. Jadi pada pengujian model kedua ini *Compute Engine GPU* menjadi yang terbaik.

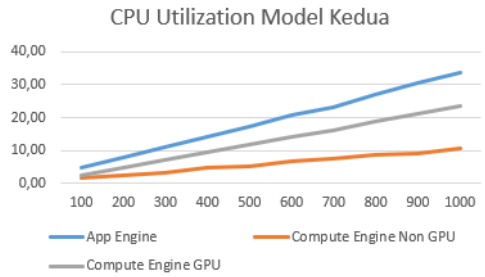


Gambar 5. Pengujian *throughput POST*

4.5. Hasil Pengujian *CPU Utilization Server* Pertama *Method* POST

Pada pengujian model pertama, rata-rata keseluruhan hasil pengujian *App Engine* yaitu 34,85% sedangkan *Compute Engine Non GPU* memiliki rata-rata yaitu 5,41% dan *Compute Engine GPU* memiliki rata-rata yaitu 6,78%. Jadi pada pengujian model pertama ini *App Engine* menjadi yang terburuk. Selanjutnya berdasarkan grafik pengujian model kedua rata-rata keseluruhan hasil pengujian *App Engine* yaitu 19,03% sedangkan *Compute Engine Non GPU* memiliki rata-rata yaitu 5,95% dan *Compute Engine GPU* memiliki rata-rata yaitu 13,04%. Jadi pada pengujian model kedua ini *App Engine* menjadi yang terburuk.

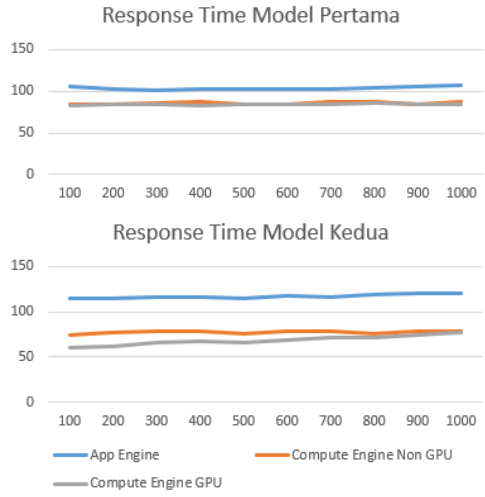




Gambar 6. Pengujian *CPU utilization POST*

4.6. Hasil Pengujian *Response Time Server* Pertama *Method GET*

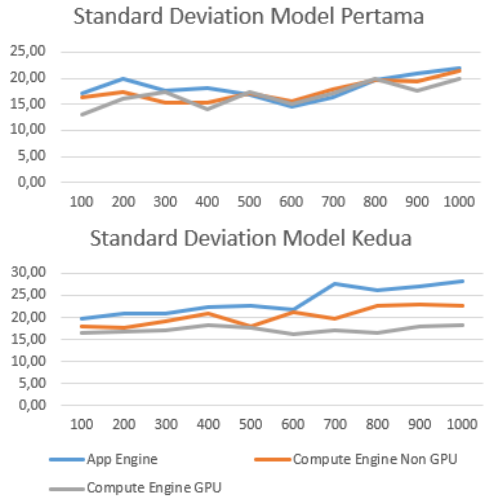
Hasil pengujian model pertama pada *App Engine* memiliki performa rata-rata yaitu sebesar 103,5 *milisecond*, *Compute Engine Non GPU* memiliki rata-rata sebesar 85,9 *milisecond* dan *Compute Engine GPU* memiliki rata-rata sebesar 84,4 *milisecond*. Jadi pada pengujian model pertama ini *Compute Engine GPU* menjadi yang terbaik. Selanjutnya untuk hasil pengujian model kedua, *App Engine* memiliki rata-rata yaitu sebesar 114,2 *milisecond*. *Compute Engine Non GPU* memiliki rata-rata sebesar 77,5 *milisecond* dan *Compute Engine GPU* memiliki rata-rata sebesar 70,5 *milisecond*. Jadi *Compute Engine GPU* menjadi yang terbaik.



Gambar 7. Pengujian *response time GET*

4.7. Hasil Pengujian *Standard Deviation Server* Pertama *Method GET*

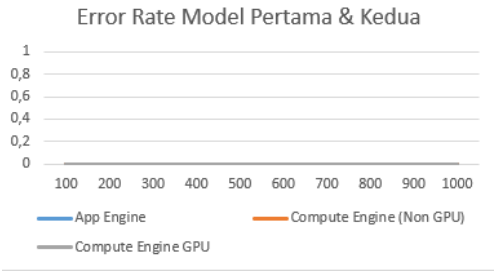
Berdasarkan hasil pengujian model pertama, *App Engine* memiliki rata-rata hasil pengujian keseluruhan sebesar 18,328. Kemudian *Compute Engine Non GPU* memiliki rata-rata hasil pengujian keseluruhan sebesar 17,583. Sedangkan *Compute Engine GPU* memiliki rata-rata hasil pengujian keseluruhan sebesar 16,768. Jadi pada pengujian ini *Compute Engine GPU* menjadi yang terbaik. Selanjutnya pada hasil pengujian model kedua, *App Engine* memiliki rata-rata hasil pengujian keseluruhan sebesar 23,781. Kemudian *Compute Engine Non GPU* memiliki rata-rata hasil pengujian keseluruhan sebesar 20,247. Sedangkan *Compute Engine GPU* memiliki rata-rata hasil pengujian keseluruhan sebesar 17,234. Jadi pada pengujian model kedua ini *Compute Engine GPU* menjadi yang terbaik.



Gambar 8. Pengujian *response time GET*

4.8. Hasil Pengujian *Error Rate Server* Pertama *Method GET*

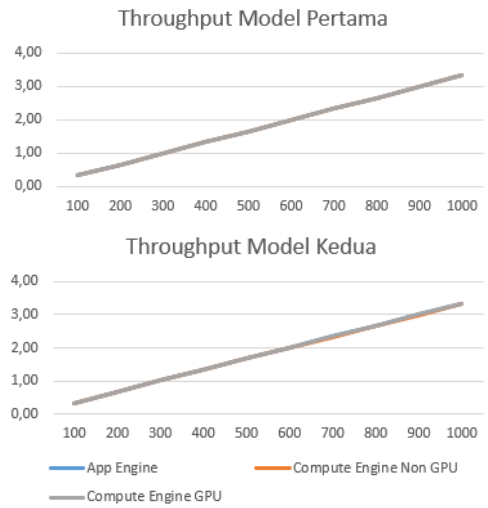
Berdasarkan hasil pengujian yang telah dilakukan kepada ketiga layanan yaitu *App Engine*, *Compute Engine Non GPU*, dan *Compute Engine GPU*, terlihat bahwa semua layanan tersebut dapat menjalankan seluruh permintaan (*request*) pengguna sehingga tidak terjadi *error* pada *server*. Jadi pada pengujian ini performa ketiga layanan tersebut pada kedua model dapat dikatakan setara.



Gambar 9. Pengujian *error rate GET*

4.9. Hasil Pengujian *Throughput Server* Pertama *Method GET*

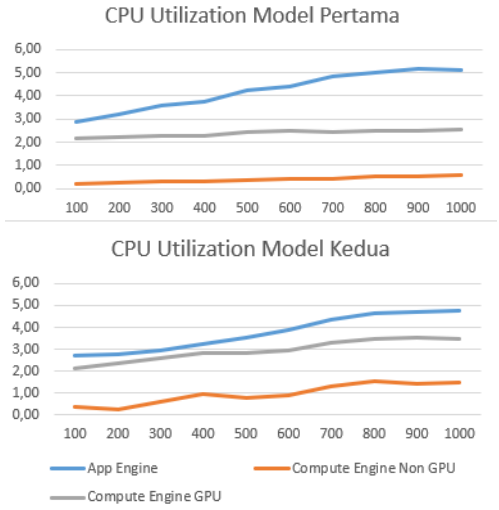
Berdasarkan hasil pengujian model pertama, *App Engine* memiliki rata-rata 1,834 *request* per detik, *Compute Engine Non GPU* memiliki rata-rata 1,836 *request* per detik dan *Compute Engine GPU* dengan rata-rata 1,836 *request* per detik. Jadi pada pengujian ini *Compute Engine Non GPU dan Compute Engine GPU* menjadi yang terbaik. Selanjutnya pada pengujian model kedua, *App Engine* dengan rata-rata 1,837 request per detik, *Compute Engine Non GPU* dengan rata-rata 1,837 request per detik dan *Compute Engine GPU* dengan rata-rata 1,837 request per detik. Jadi pada pengujian ini ketiga layanan menunjukan performa yang setara.



Gambar 10. Pengujian *throughput GET*

4.10. Hasil Pengujian *CPU Utilization Server* Pertama *Method GET*

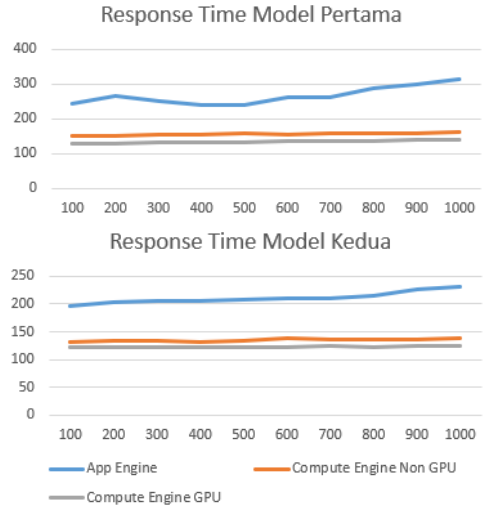
Berdasarkan hasil pengujian model pertama, *App Engine* memiliki rata-rata performa *CPU Utilization* sebesar 4,23%, sedangkan *Compute Engine GPU* sebesar 2,38% dan *Compute Engine Non GPU* sebesar 0,39%. Jadi pada pengujian model pertama, *App Engine* menjadi yang terburuk. Selanjutnya pada pengujian model kedua, rata-rata keseluruhan hasil pengujian *App Engine* yaitu 3,77%, sedangkan *Compute Engine GPU* sebesar 2,85% dan *Compute Engine Non GPU* sebesar 0,95%. Jadi pada pengujian model kedua, *App Engine* menjadi yang terburuk.



Gambar 11. Pengujian *CPU utilization GET*

4.11. Hasil Pengujian *Response Time Server* Kedua *Method POST*

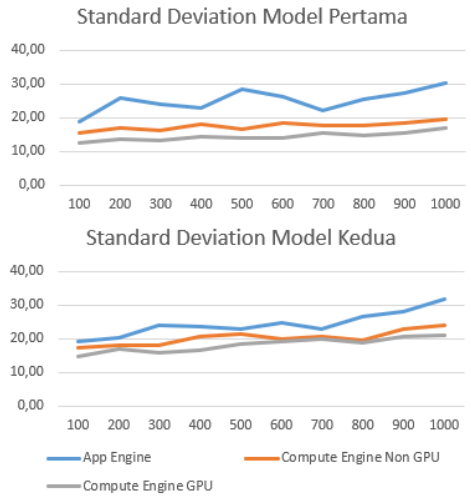
Pada pengujian menggunakan model pertama, jika dilihat dari rata-rata keseluruhan hasil pengujian *response time*, *App Engine* memiliki rata-rata 271 *milisecond*, *Compute Engine Non GPU* memiliki rata-rata 160,6 *milisecond* dan *Compute Engine GPU* memiliki rata-rata 135,1 *milisecond*. Jadi pada pengujian ini *Compute Engine GPU* menunjukan performa yang terbaik. Sedangkan saat menggunakan model kedua, *App Engine* memiliki rata-rata 220,2 *milisecond*, *Compute Engine Non GPU* memiliki rata-rata 134,5 *milisecond* dan *Compute Engine GPU* memiliki rata-rata 129 *milisecond*. Jadi pada pengujian ini *Compute Engine GPU* menunjukan performa yang terbaik.



Gambar 12. Pengujian *response time POST*

4.12. Hasil Pengujian *Standard Deviation Server* Kedua *Method* POST

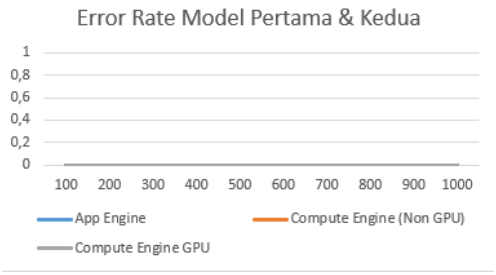
Pada pengujian menggunakan model pertama, Jika dilihat dari rata-rata keseluruhan hasil pengujian *standard deviation*, *App Engine* memiliki rata-rata 25,176, *Compute Engine Non GPU* memiliki rata-rata 17,579 dan *Compute Engine GPU* memiliki rata-rata 14,401. Jadi pada pengujian ini *Compute Engine GPU* menunjukan performa yang terbaik. Selanjutnya pada pengujian menggunakan model kedua, *App Engine* memiliki rata-rata 24,378, *Compute Engine Non GPU* memiliki rata-rata 20,103 dan *Compute Engine GPU* memiliki rata-rata 18,144. Jadi pada pengujian ini *Compute Engine GPU* menunjukan performa yang terbaik.



Gambar 13. Pengujian *standard deviation POST*

4.13. Hasil Pengujian *Error Rate Server* Kedua *Method* POST

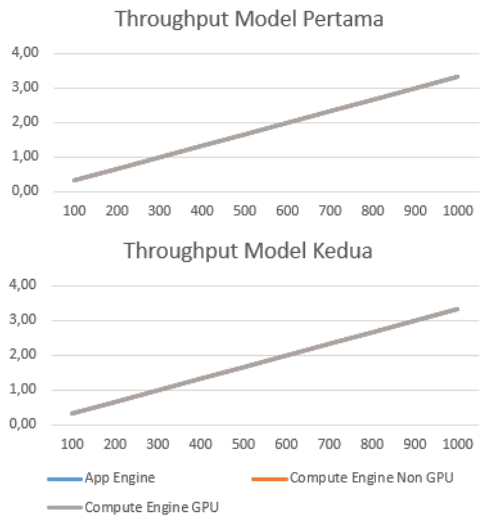
Berdasarkan hasil pengujian yang telah dilakukan kepada ketiga layanan yaitu *App Engine*, *Compute Engine Non GPU*, dan *Compute Engine GPU*, terlihat bahwa semua layanan tersebut dapat menjalankan seluruh permintaan (*request*) pengguna sehingga tidak terjadi *error* pada *server*. Jadi pada pengujian ini performa ketiga layanan tersebut pada kedua model dapat dikatakan setara.



Gambar 14. Pengujian *error rate POST*

4.14. Hasil Pengujian *Throughput Server* Kedua *Method* POST

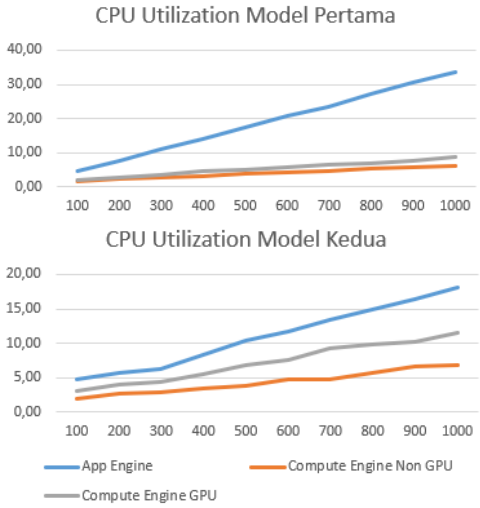
Pada saat pengujian menggunakan model pertama, jika dilihat dari rata-rata keseluruhan hasil pengujian *throughput*, *App Engine* memiliki rata-rata 1,834, *Compute Engine Non GPU* memiliki rata-rata 1,834 dan *Compute Engine GPU* memiliki rata-rata 1,834. Jadi pada pengujian ini ketiga layanan menunjukan performa yang setara. Sedangkan saat menggunakan model kedua, *App Engine* memiliki rata-rata 1,834, *Compute Engine Non GPU* memiliki rata-rata 1,834 dan *Compute Engine GPU* memiliki rata-rata 1,834. Jadi pada pengujian ini ketiga layanan kembali menunjukan performa yang setara.



Gambar 15. Pengujian *throughput POST*

4.15. Hasil Pengujian *CPU Utilization Server* Kedua *Method* POST

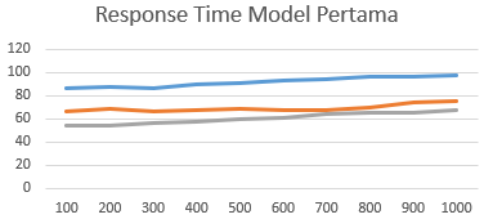
Pada saat pengujian model pertama, jika dilihat dari rata-rata keseluruhan hasil pengujian *CPU Utilization*, *App Engine* memiliki rata-rata 19,03, *Compute Engine Non GPU* memiliki rata-rata 4,01 dan *Compute Engine GPU* memiliki rata-rata 5,20. Jadi pada pengujian ini *App Engine* memiliki performa terburuk. Sedangkan saat menggunakan model kedua, *App Engine* memiliki rata-rata 10,92, *Compute Engine Non GPU* memiliki rata-rata 4,26 dan *Compute Engine GPU* memiliki rata-rata 7,15. Jadi pada pengujian ini *App Engine* kembali memiliki performa terburuk.

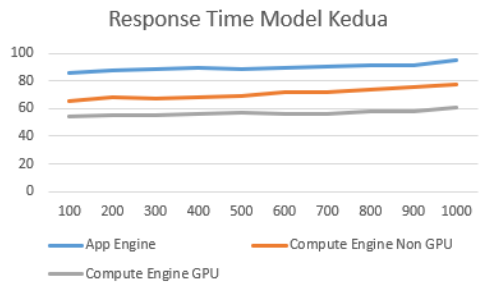


Gambar 16. Pengujian *throughput POST*

4.16. Hasil Pengujian *Response Time Server* Kedua *Method GET*

Pada pengujian model pertama, jika dilihat dari rata-rata keseluruhan hasil pengujian *response time*, *App Engine* memiliki rata-rata 91,9 *milisecond*, *Compute Engine Non GPU* memiliki rata-rata 71,2 *milisecond* dan *Compute Engine GPU* memiliki rata-rata 60,5 *milisecond*. Jadi pada pengujian ini *Compute Engine GPU* memiliki performa terbaik. Selanjutnya pada pengujian model kedua, dari rata-rata keseluruhan hasil pengujian *response time*, *App Engine* memiliki rata-rata 90,2 *milisecond*, *Compute Engine Non GPU* memiliki rata-rata 72 *milisecond* dan *Compute Engine GPU* memiliki rata-rata 56,6 *milisecond*. Jadi pada pengujian ini *Compute Engine GPU* memiliki performa terbaik.

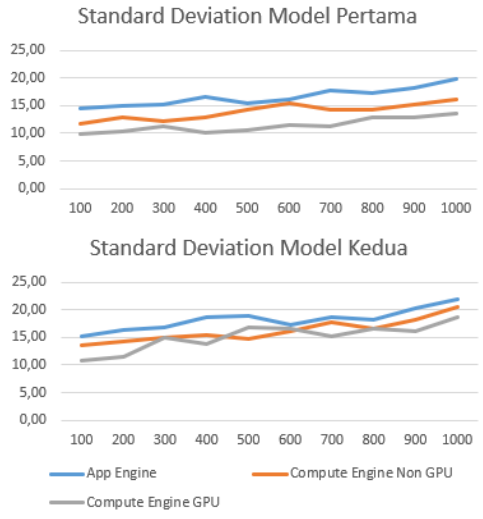




Gambar 17. Pengujian *response time GET*

4.17. Hasil Pengujian *Standard Deviation Server* Kedua *Method GET*

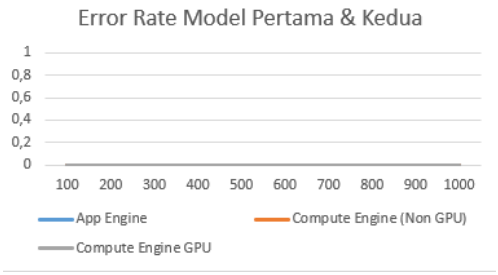
Pada pengujian model pertama, dilihat dari rata-rata keseluruhan hasil pengujian standard deviation, *App Engine* memiliki rata-rata 16,602, *Compute Engine Non GPU* memiliki rata-rata 13,92 dan *Compute Engine GPU* memiliki rata-rata 11,46. Jadi pada pengujian ini *Compute Engine GPU* memiliki performa terbaik. Selanjutnya pada pengujian model kedua, *App Engine* memiliki rata-rata 18,27, *Compute Engine Non GPU* memiliki rata-rata 16,292 dan *Compute Engine GPU* memiliki rata-rata 15,133. Jadi pada pengujian model kedua ini *Compute Engine GPU* memiliki performa terbaik.



Gambar 18. Pengujian *standard deviation GET*

4.18. Hasil Pengujian *Error Rate Server* Kedua *Method GET*

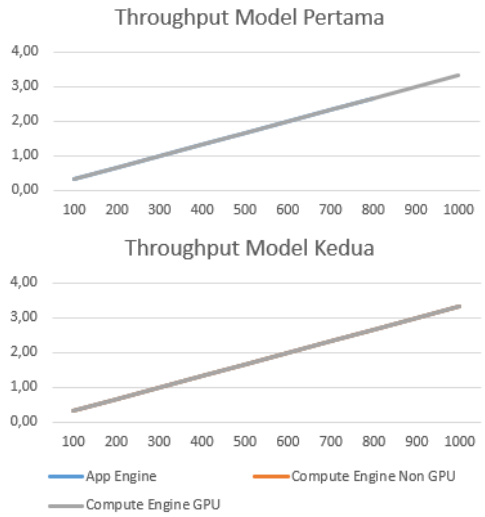
Berdasarkan hasil pengujian yang telah dilakukan kepada ketiga layanan yaitu *App Engine*, *Compute Engine Non GPU*, dan *Compute Engine GPU*, terlihat bahwa semua layanan tersebut dapat menjalankan seluruh permintaan (*request*) pengguna sehingga tidak terjadi *error* pada *server*. Jadi pada pengujian ini performa ketiga layanan tersebut pada kedua model dapat dikatakan setara.



Gambar 19. Pengujian *error rate GET*

4.19. Hasil Pengujian *Throughput Server* Kedua *Method GET*

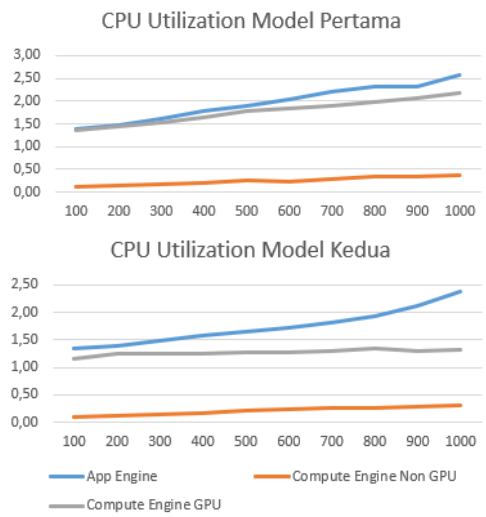
Pengujian untuk model pertama menunjukan rata-rata keseluruhan hasil pengujian *App Engine* memiliki rata-rata 1,837, *Compute Engine Non GPU* memiliki rata-rata 1,837 dan *Compute Engine GPU* memiliki rata-rata 1,837. Jadi pada pengujian ini ketiga layanan menunjukan performa yang setara. Selanjutnya pada pengujian model kedua, *App Engine* memiliki rata-rata 1,837, *Compute Engine Non GPU* memiliki rata-rata 1,837 dan *Compute Engine GPU* memiliki rata-rata 1,837. Jadi pada pengujian ini ketiga layanan menunjukan performa yang setara.



Gambar 20. Pengujian *throughput GET*

4.20. Hasil Pengujian *CPU Utilization Server* Kedua *Method GET*

Pada pengujian model pertama, jika dilihat dari rata-rata keseluruhan hasil pengujian *CPU Utilization*, *App Engine* memiliki rata-rata 1,96%, *Compute Engine Non GPU* memiliki rata-rata 0,25% dan *Compute Engine GPU* memiliki rata-rata 1,83%. Jadi pada pengujian menggunakan model pertama ini *App Engine* memiliki performa terburuk. Selanjutnya pada pengujian model kedua, *App Engine* memiliki rata-rata 1,68%, *Compute Engine Non GPU* memiliki rata-rata 0,22% dan *Compute Engine GPU* memiliki rata-rata 1,36%. Jadi pada pengujian model kedua, *App Engine* memiliki performa terburuk.



Gambar 21. Pengujian *CPU utilization GET*

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dari penelitian yang telah dilakukan, maka dapat ditarik kesimpulan yaitu saat pengujian *response time* dan *standard deviation*, *Compute Engine GPU* memiliki performa *response time* dan *standard deviation* terbaik. Penggunaan GPU *(Graphical Processing Unit)* dapat membantu mempercepat proses *request* data dan juga membutuhkan sumber daya yang lebih besar untuk mengoptimalkan performa. Ketika melakukan pengujian *error rate* menggunakan *method POST* dan *GET*, tidak terlihat terjadinya *error* pada kedua layanan. Pada pengujian *throughput*, secara keseluruhan *Compute Engine (GPU* dan *Non GPU)* memiliki performa terbaik. Lalu saat pengujian *CPU Utilization*, *App Engine* memiliki performa yang terburuk dibandingkan *Compute Engine (Non GPU* dan *GPU)*. Selain itu, disarankan juga untuk memilih layanan yang sesuai dengan kebutuhan yang diperlukan. Apabila membutuhkan layanan *minimally managed service* (layanan yang mengharuskan pengguna mengelola banyak aspek sistem seperti instalasi *web server*) yang dapat menangani request pengguna dalam melakukan prediksi gambar dengan cepat dan memiliki penggunaan CPU yang rendah maka *Compute Engine* menjadi pilihan layanan yang terbaik karena unggul dari segi *response time*, *standard deviation*, thorughput dan *CPU Utilization*. Apabila ingin memaksimalkan performa *Compute Engine* maka disarankan untuk menambahkan GPU karena terbukti dapat meningkatkan performa dalam memprediksi gambar. Sedangkan jika membutuhkan layanan *highly managed service* (layanan yang tidak mengharuskan mengelola banyak aspek sistem) dan tidak terlalu mementingkan performa *response time*, *standard deviation*, *throughput* dan *CPU Utilization* maka *App Engine* dapat menjadi pilihan yang lebih baik.

DAFTAR PUSTAKA

[1] PERDAMI, "Vision 2020 di Indonesia," https://perdami.or.id/vision-2020-di-indonesia.

[2] Dinas Kesehatan Kota Denpasar, "Laporan Tahunan Dinas Kesehatan Kota Denpasar 2021," https://www.dinkes.denpasarkota.go.id/public/uploads/download/download_222107120748_laporan-tahunan-dinas-kesehatan-kota-denpasar-2022.pdf .

[3] M. Idris, R. I. Adam, Y. Brianorman, R. Munir, and D. Mahayana, "Kebenaran dalam Perspektif Filsafat Ilmu Pengetahuan dan Implementasi dalam Data Science dan Machine Leaning," *Jurnal Filsafat Indonesia*, vol. 5, no. 2, pp. 173–181, Jul. 2022, doi: 10.23887/jfi.v5i2.42207.

[4] Y. I. Sulistya, "Analisis perbandingan Reduction Technique dengan metode Dimentional Reduction dan Cross Validation pada dataset Breast Cancer," *Indonesian Journal of Data and Science*, vol. 3, no. 2, pp. 82–88, Sep. 2022, doi: 10.56705/ijodas.v3i2.41.

[5] T. Khan, W. Tian, G. Zhou, S. Ilager, M. Gong, and R. Buyya, "Machine learning (ML)-centric resource management in cloud computing: A review and future directions," *Journal of Network and Computer Applications*, vol. 204, p. 103405, Aug. 2022, doi: 10.1016/j.jnca.2022.103405.

[6] A. B. Hikmah, Y. Sumaryana, M. Kusmira, T. Alawiyah, and Y. Apriyani, "Implementasi Teknologi Cloud Computing Pada Sistem Transportasi Angkutan Umum Kota Tasikmalaya," *IJCIT (Indonesian Journal on Computer and Information Technology)*, vol. 4, no. 2, Nov. 2019, doi: 10.31294/ijcit.v4i2.6191.

[7] N. Ramsari and A. Ginanjar, "Implementasi Infrastruktur Server Berbasis Cloud Computing Untuk Web Service Berbasis Teknologi Google Cloud Platform," *Conference SENATIK STT Adisutjipto Yogyakarta*, vol. 7, Mar. 2022, doi: 10.28989/senatik.v7i0.472.

[8] Ilmi Barokah and Asriyanik Asriyanik, "Analisis Perbandingan Serverless Computing pada Google Cloud Platform," *Jurnal Teknologi Informatika dan Komputer*, vol. 7, no. 2, pp. 169–187, 2021.

[9] Nur Ardli Rachmat Saputra, "Serverless Machine Learning sebagai Solusi Deployment untuk Prototipe Sign Language Recognition App Menggunakan Google Cloud Platform," Universitas Jendral Soedirman, Jakarta, 2022. Accessed: Aug. 31, 2023. [Online]. Available: http://repository.unsoed.ac.id/id/eprint/13977

[10] Shilpashree. S, R. R. Patil, and P. C, "'Cloud computing an overview,'" *International Journal of Engineering & Technology*, vol. 7, no. 4, p. 2743, Oct. 2018, doi: 10.14419/ijet.v7i4.10904.

[11] A. Ismail, Ahmadi Yuli Ananta, Sofyan Noor Arief, and Elok Nur Hamdana, "Performance Testing Sistem Ujian Online Menggunakan

Jmeter Pada Lingkungan Virtual," *Jurnal Informatika Polinema*, vol. 9, no. 2, pp. 159–164, Feb. 2023, doi: 10.33795/jip.v9i2.1190.

[12] Arlinta Christy Barus, Johannes Harungguan, and Efren Manulu, "Pengujian API website Untuk Perbaikan performansi APLIKASI DITENUN," *Journal of Applied Technology and Informatics Indonesia*, vol. 1, no. 2, pp. 14–21, 2022.

[13] D. I. Permatasari, "Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 8, no. 1, p. 135, Jan. 2020, doi: 10.26418/justin.v8i1.34452.

[14] S. E. Prasetyo and Y. Salimin, "Analisis Perbandingan Performa Web Server Docker Swarm dengan Kubernetes Cluster," 2021. [Online]. Available: https://journal.uib.ac.id/index.php/combines

[15] Desy Intan Permatasari *et al.*, "Pengukuran Throughput Load Testing Menggunakan Test Case Sampling Gorilla Testing," *Seminar Nasional Sistem Informasi (SENASIF)*, vol. 3, no. 1, 2019.

[16] I. Yatini, F. W. Nurwiyati, and K. Anam, "PERFORMA MICROFRAMEWORK PHP PADA REST API MENGGUNAKAN METODE LOAD TESTING," *FAHMA*, vol. 19, no. 2, pp. 12–20, 2021.

[17] N. A. R. Saputra, "Serverless Machine Learning sebagai Solusi Deployment untuk Prototipe Sign Language Recognition App Menggunakan Google Cloud Platform," Universitas Jenderal Soedirman, Jakarta, 2022.

[18] L. Mulana, K. Prihandani, and A. Rizal, "Analisis Perbandingan Kinerja Framework Codeigniter Dengan Express. Js Pada Server RESTful Api," *Jurnal Ilmiah Wahana Pendidikan*, vol. 8, no. 16, pp. 316–326, 2022.

[19] Dias Astisa, "Perbandingan Hasil Belajar Siswa Antara Model Pembelajaran Kooperatif Group Investigation Dengan Two Stay Two Stray Pada Kelas IX MTS Madani Pao-pao," Universitas Islam Negeri (UIN) Alauddin Makassar, Makassar, 2016