Journal of Information Systems and Informatics

Vol. 6, No. 2, Juni 2024 e-ISSN: **2656-4882** p-ISSN: **2656-5935**

DOI: 10.51519/journalisi.v6i2.773 Published By DRPM-UBD

Comparative Analysis of Server-Based and Serverless Service Performance on Google Cloud Platform (GCP) (Case Study: Machine Learning Model Deployment)

Vina Fujiyanti[1], Galura Muhammad Suranegara[2], Ichwan Nul Ichsan[3]

[1,2,3]Telecommunication System, Universitas Pendidikan Indonesia, Purwakarta, Indonesia
Email: [1]vinafujiyanti@upi.edu, [2]galurams@upi.edu, [3]ichwannul.ichsan90@upi.edu

Abstract

Cloud infrastructure providers such as GCP provide various computing services to deploy applications such as machine learning models, namely server-based and serverless. However, the two services each have different characteristics and advantages so that this becomes a difficulty factor for users in choosing cloud services. This research was conducted to compare server-based and serverless services with the aim of knowing the best service resulting from the analysis of performance measurements, namely CPU and memory utilization, latency, pricing, and developer experiences. The application of machine learning models is carried out on Compute Engine and Vertex AI services and will be tested for performance through requests to endpoints 100 times using JMeter for 30 minutes. The findings show that Vertex AI performance is better than Compute Engine with CPU utilization of 0.10%, memory utilization of 0.94%, and latency of 17.34ms but the cost efficiency is owned by the Compute Engine.

Keywords: GCP, Serverless, Server-Based, Machine Learning Deployment

1. INTRODUCTION

In the era of rapidly growing digital transformation, the use of cloud computing technology is a consideration for users, both an enterprise and an application developer due to cost efficiency, high flexibility in managing and monitoring data centrally and the ability to handle data problems quickly [1]. Some of the things that users consider to take advantage of cloud technology compared to on-premise servers are expenses that can be minimized such as electricity costs, software purchase costs and server procurement costs whose failure rate is estimated to reach 55-75% [2]. This failure can occur due to poor server infrastructure development so that server usage is not optimal, such as lack of server capacity in handling overload and delays or inaccuracies in regular server maintenance [1]. In addition, cloud technology provides a variety of server services to support the deployment of applications, one of which is computing services. Google Cloud Platform (GCP) is one of the cloud service providers that provides various

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

computing services consisting of server-based and serverless services. Server-based services refer to the IaaS (Infrastructure as a Service) cloud computing model where users have control over computing, storage, networking, and other services through managing operating systems and applications in virtual machines on a pay-per-use basis [3]. Meanwhile, serverless services allow users to only develop applications through application code stored in cloud containers without having to manage the development infrastructure because it is managed directly by the cloud service provider [4]. Both computing services are used for the deployment of an application, one of which is a machine learning model.

Machine learning is a part of artificial intelligence (AI) that can learn patterns from data without the need to define them directly [5]. Like other applications, machine learning can also be implemented through local sever or cloud. However, there are some drawbacks of using local servers to deploy machine learning, one of which is limited data access and processing [6]. Cloud technology has the potential to overcome these drawbacks with available services so that access to applications becomes easy and data processing time and implementation of machine learning can be done more quickly [7], [8]. GCP has a specialized service for managing AI or machine learning, namely Vertex AI.

Some previous researches that utilize Cloud technology to deploy applications include two researches conducted by Abraham & Yang in deploying real-time bus location tracking web applications. The first research compared Cloud Run and App Engine services while the second research compared Cloud Run, AWS App Runner, and Azure Container Apps services. Both researches only focused on comparing services without a server and the system configuration specified was different for each resource. The results show Cloud Run has better performance [3], [4]. Then research conducted by Rahman in developing online store web applications and deployed on Cloud Run, GKE AutoPilot, and AWS EKS with AWS Fargate services. This research also only compares serverless services and states that Cloud Run has high performance [9]. Another research was conducted by Wiranata who compared App Engine services with Compute Engine to implement a machine learning model that can predict eye diseases. The CNN model is used in machine learning and the results show that Compute Engine has better performance [10]. From these researches, serverless services are most widely used compared to server-based services.

Reference [11] said that serverless computing has the potential to provide services that are high performance, low cost, and easy to manage. However, server-based services also provide high performance and high flexibility through user management of a combination of computing devices such as CPU and GPU, cloud storage, and other features as needed to implement high-performance data science or machine learning [3], [12]. With the performance advantages provided by each

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

service, users must have considerations to choose a service that suits their needs for machine learning deployment. Reference [13] said that the cloud service selection process cannot be taken lightly and the difficult consumer factors in choosing cloud services include the level of knowledge and understanding of cloud service requirements that vary greatly and service providers that provide various services based on variations in performance, prices, and others. Therefore, research is needed that can be a reference for users to consider choosing the appropriate cloud computing service for deploying applications, especially in the context of machine learning models.

From the problems that have been described, this research will compare server-based and serverless services to find out the best service used in deploying machine learning models based on service performance measurements. In this research, the services used to deploy machine learning models are Compute Engine (server-based) and Vertex AI (serverless) because Vertex AI services are still rarely used as a comparator in deploying applications. The machine learning model deployed is smart agriculture based on machine learning to find out recommendations for crop seeds.

2. METHODS

The type of research used in this research is quantitative with experimental research methods to deploy machine learning models on server-based and serverless services at GCP and measure performance using several parameters, including CPU and memory utilization, latency, and pricing. CPU and memory utilization are used to predict future host performance. This is very important because workload prediction is a crucial aspect in managing cloud infrastructure [14], [15].

For additional comparison parameters are developer experiences which include machine learning framework compatibility, ease of implementation, and availability of documentation. After the measurement, a comparison of server-based (Compute Engine) and serverless (Vertex AI) service performance is conducted to determine the best service for machine learning deployment model. Figure 1 presents the research flowchart [16].

1) Literature Review

The initial stage aims to explore literature sources related to the use of cloud technology such as cloud services and collect information relevant to the use of cloud services and analyze research needs or gaps in these literature sources.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2) Problem Identification

This research identifies problems that exist in the use of cloud services after exploring existing literature sources. Then focus the problem on the literature regarding the use of cloud computing services for application deployment, especially machine learning model applications.

3) Problem Formulation

At the problem formulation stage, problems regarding application deployment on cloud services are formulated specifically to achieve the research objectives, namely, to find out the best cloud service by deploying a machine learning model at GCP.

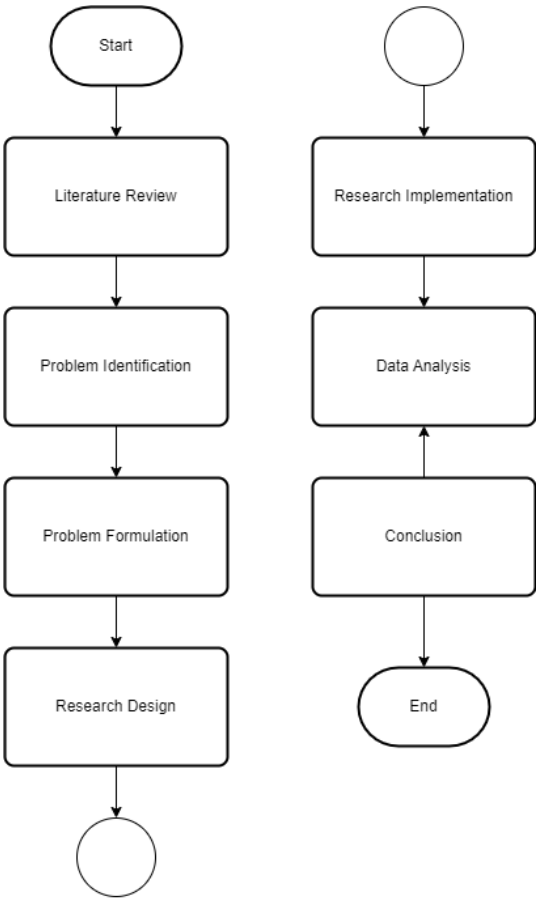

Figure 1. Research Method

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

4) Research Design

The research design is determined as a reference for creating an application deployment system on server-based and serverless services. This research uses a browser to access Google Cloud Platform, and Apache JMeter (v5.6.3) software to test through endpoint requests from the results of the deployment of machine learning models on Compute Engine and Vertex AI. For deployment needs, machine learning models, API files to handle HTTP requests and responses, and dependencies consisting of Flask (v2.3.2), Gunicorn (v.21.0.0), and scikit-learn (v1.3.0).

The system configuration of each server-based and serverless service is presented in Table 1. The machine learning model to be deployed has been trained locally using the Decision Tree algorithm in the scikit-learn framework version 1.3.0 with the output file in .joblib format and the best prediction accuracy rate of 98.41%. The model contains training code. Then, the API to handle HTTP requests and responses uses python 3.11. Table 1 shows the system configuration of this research.

Table 1. Configuration System

System Configuration	Server-based	Serverless
	Compute Engine	**Vertex AI**
Instance Type	e2-standard-4	e2-standard-4
Number of vCPU	4	4
Memory	16 GB	16 GB

The system configuration in Table 1 is the resource specification of each service for the deployment of the machine learning model in this research. Referring to the GCP documentation, instance type E2 provides a balanced service between price and performance (price-to-performance) and is suitable for application testing.

5) Research Implementation

The research implementation includes deployment models and testing to measure the performance of Compute Engine and Vertex AI. Performance testing can be done to assess applications and systems before production, compare the performance characteristics of several systems under test, and analyze the sources that hinder the performance of the systems under test [17]. The implementation varies depending on the requirements for each service and the process being carried out. After the deployment model is carried out on each service, there are endpoint deployment results that will be tested through JMeter as many as 10 sets

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

of tests with each set of 10 requests for 30 minutes to the application through the endpoint. Thus, the total test will consist of 100 requests. After that, the request process will be monitored using Cloud Monitoring and each performance parameter data will be collected.

6) Data Analysis

After the data is collected, the data is analyzed using descriptive statistical analysis, namely the calculation of the average (Equation 1) [18] and comparative analysis to compare the performance of serverless and server-based services. The network performance measurement results will be validated for quality using TIPHON QoS standardization presented in Table 2 [19].

$$Average\ (\bar{x}) = \frac{total\ reps\ result}{number\ of\ reps} \quad (1)$$

Table 2. TIPHON latency standard

Degradation Category	**Latency (ms)**
Very Good	< 150
Good	150 s/d 300
Medium	300 s/d 450
Bad	> 450

7) Conclusion

After analyzing the data obtained through performance measurement of each server-based and serverless service, it will be concluded which service is the best in deploying machine learning models.

3. RESULTS AND DISCUSSION

The results of serverless and server-based service performances testing are presented and analyzed based on predetermined parameters, namely CPU and memory utilization, latency, pricing, and developer experiences as additional comparison parameters. The presentation of costs in rupiah in this study is a conversion when the nominal exchange rate from USD to rupiah is IDR 16,209.99.

3.1. Server-based Service Performances

Server-based service performance only describes the results of Compute Engine service performance testing as follows:

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

1) CPU Utilization

The results of CPU utilization performance testing are shown in Table 3 and graph in Figure 2. The average utilization reaches 0.79% with the highest utilization reaching 2.93% and the lowest utilization is 0.23%.

Table 3. CPU Utilization Measurement Result of Compute Engine Service

Time	CPU Utilization	Time	CPU Utilization
17:43:00	0.22%	17:58:00	2.49%
17:44:00	0.23%	17:59:00	0.32%
17:45:00	0.27%	18:00:00	0.29%
17:46:00	0.28%	18:01:00	0.30%
17:47:00	2.89%	18:02:00	0.32%
17:48:00	2.46%	18:03:00	0.31%
17:49:00	0.30%	18:04:00	0.31%
17:50:00	0.30%	18:05:00	0.33%
17:51:00	0.30%	18:06:00	0.34%
17:52:00	0.30%	18:07:00	2.93%
17:53:00	0.30%	18:08:00	2.49%
17:54:00	0.30%	18:09:00	0.37%
17:55:00	0.30%	18:10:00	0.40%
17:56:00	0.31%	18:11:00	0.40%
17:57:00	2.92%	18:14:00	0.37%
Average			**0.79%**

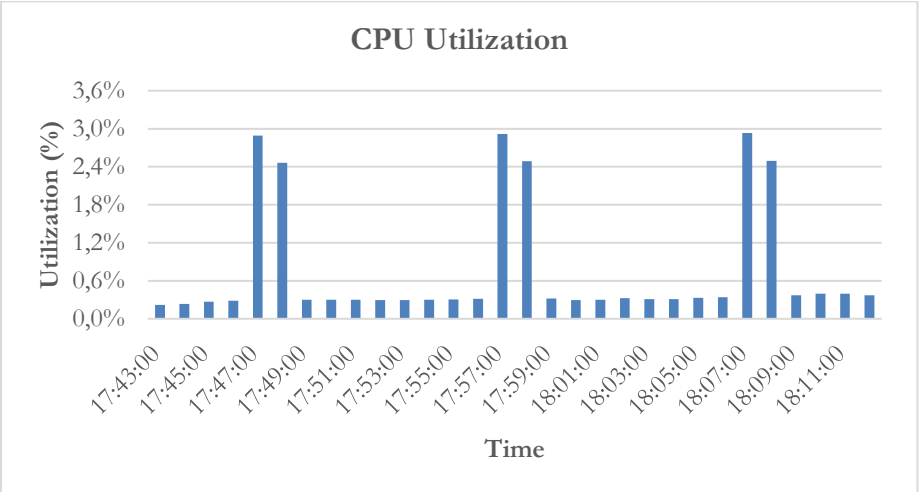

Figure 2. Compute Engine Service CPU Utilization

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Based on Figure 2 and Figure 3, CPU performance spikes at the 5th (17:47:00), 15th (17:57:00), and 25th (18:07:00) minutes every 10 minutes which simultaneously have errors in logging, namely accessing the Metadata Server (MDS) and failure to obtain the certificate workload configuration status of the MDS in the same minute. Therefore, it can be concluded that the spike was caused by an error in accessing the metadata server. Certificate workload is a credential that each VM uses to establish secure communication and is updated every 10 minutes on the active instance. Based on GCP documentation regarding the guest environment and VM Metadata, these credentials are managed by a guest agent that is useful for reading server metadata so that the VM runs properly on the Compute Engine. Metadata is owned by each instance to provide instance-specific information stored on the Metadata Server. The error is shown in the log in Figure 3.

Figure 3. Compute Engine Service Log

Then for the time range 17.43.00 to 18.14.00 in addition to the spike value, the utilization percentage graph is quite stable in the range of 0.22% to 0.40%. In addition to spikes caused by errors, internal installation tasks or packages running in the virtual machine background also affect the spike. Even so, the CPU utilization value can be said to be good because the resulting value and the difference between the maximum and minimum values are small, which is 2.70%.

2) Memory Utilization

The results of the memory utilization performance test are shown in Table 4 and the graph in Figure 4. The average utilization is 3.43% with the highest utilization reaching 3.7% and the lowest utilization is 3.37%.

Table 4. Memory Utilization Measurement Result of Compute Engine Service

Time	Memory Utilization	Time	Memory Utilization
17:43:00	3.37%	17:59:00	3.44%
17:44:00	3.37%	18:01:00	3.44%
17:45:00	3.37%	18:02:00	3.46%

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Time	Memory Utilization	Time	Memory Utilization
17:46:00	3.37%	18:03:00	3.45%
17:47:00	3.68%	18:04:00	3.45%
17:48:00	3.38%	18:05:00	3.45%
17:49:00	3.38%	18:06:00	3.44%
17:50:00	3.38%	18:07:00	3.67%
17:51:00	3.38%	18:08:00	3.37%
17:52:00	3.38%	18:09:00	3.38%
17:53:00	3.38%	18:10:00	3.39%
17:54:00	3.38%	18:11:00	3.40%
17:55:00	3.38%	18:12:00	3.41%
17:56:00	3.38%	18:13:00	3.37%
17:57:00	3.70%	18:14:00	3.37%
Average			**3.43%**

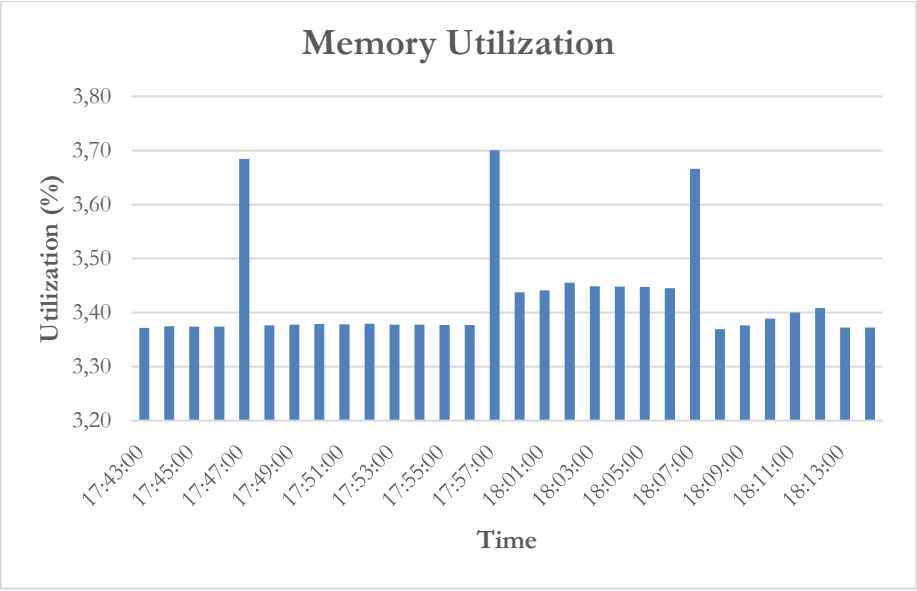

Figure 4. Compute Engine Service Memory Utilization

In Figure 4, it can be seen that the graph shows the highest spike of 3.70% at 17:57:00. The second highest utilization percentage spike of 3.68% at 17:47:00 and a spike of 3.67% at 18:07:00 are also caused by access to the Metadata Server (MDS) and failure to obtain the certificate workload configuration status of the MDS such as CPU utilization. Then, for the time range of 17:43:00 to 18:14:00 in addition to the spike value, the utilization percentage graph shows stability with values in the range of 0.22% to 0.40%.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3) Latency

The results of latency performance testing are shown in Table 5 and graph in Figure 5. The average latency obtained is 128.62ms with the highest latency of 267.42ms and the lowest latency of 7.31ms.

Table 5. Compute Engine Service Latency Measurement Results

Time	Latency (ms)
17:47:00	7.78
17:55:00	267.42
17:56:00	161.58
17:59:00	7.80
18:02:00	7.31
18:06:00	223.26
18:10:00	225.17
Average	**128.62**

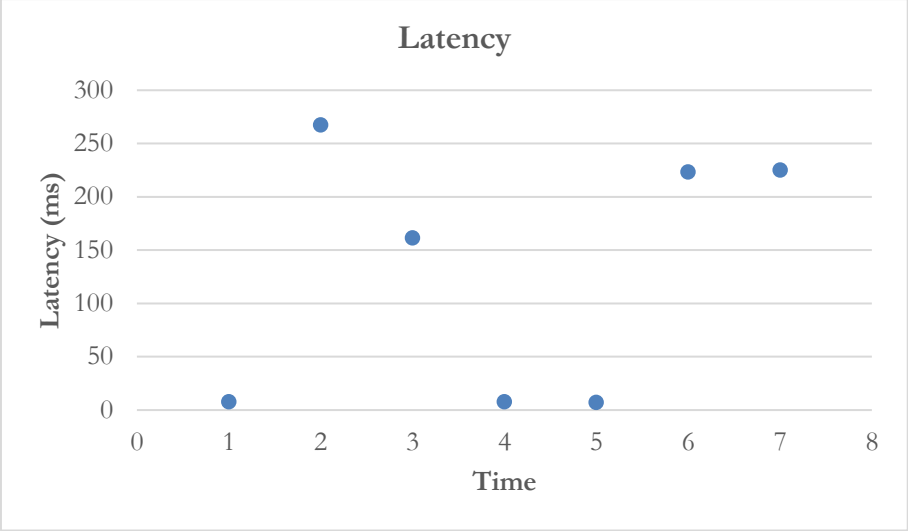

Figure 5. Compute Engine Service Latency

Based on Figure 5, latency on the Compute Engine service is less stable because there are significant spikes in minutes 2, 6, and 7 with the highest spike of 267.2ms. This can be caused by high transaction loads that cannot be managed by the server at that time. Then, geographic location can also affect network latency. Data transfers that occur between regions or between zones on a VM can cause delays due to the distance between the location and the user. In addition, the latency data

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

generated was delayed during real time testing so that less data was generated because the GCP metrics required additional processing time in displaying the data. The location of processing infrastructure is key in network latency [20].

4) Pricing

The cost incurred if all resources on the Compute Engine service are activated for one month to perform deployment and testing is estimated at $138.79 or around Rp 2,249,785 based on cost calculations with details as presented in Table 6. Then, the total cost during testing for 30 minutes the instance is activated is about $0.095 or about Rp 1,540. The cost of data transfer depends on the destination location where the VM data is transmitted from the source.

Table 6. Compute Engine Usage Cost

Resource	Cost per hour	Estimated Cost per month
Machine type: e2-standard-4		
4 vCPU 16 GB memory	$0.18	$131.55
10 GB Balanced Persistent Disk	-	$1.30
Storage Persistent Disk	-	$0.052
External IP (Standard VM)	$0.005	$3.65
Data transfer	-	$0.1
VM manager	$0.003	$2.19
Total (one month)		**$138.79**
Total (during the test)		**$0.095**

Compute Engine costs are charged based on the resources and network used. However, the cost for data transfer will increase if there is data transmission within the Google Cloud region or zone. Overall, the Compute Engine performance measurement value has a small difference despite the fluctuations caused by errors on the Server. In terms of cost, Compute Engine does have a fairly expensive cost. However, because the server can be managed directly by the user, the resources can be arranged so that the costs incurred are also optimized according to needs.

3.2. Serverless Service Performances

Serverless service performance only describes the results of testing the performance of Vertex AI service as follows.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

1) CPU Utilization

The results of testing CPU utilization performance on the Vertex AI service are shown in Table 7 and the graph in Figure 6. The average utilization reaches 0.098% with the highest utilization reaching 0.104% and the lowest utilization is 0.095%.

Table 7. Vertex AI Service CPU Utilization Measurement Results

Time	CPU Utilization	Time	CPU Utilization
19:25:00	0.100%	19:40:00	0.097%
19:26:00	0.097%	19:41:00	0.095%
19:27:00	0.097%	19:42:00	0.097%
19:28:00	0.095%	19:43:00	0.100%
19:29:00	0.098%	19:44:00	0.099%
19:30:00	0.104%	19:45:00	0.096%
19:31:00	0.100%	19:46:00	0.098%
19:32:00	0.097%	19:47:00	0.099%
19:33:00	0.099%	19:48:00	0.099%
19:34:00	0.101%	19:49:00	0.096%
19:35:00	0.097%	19:50:00	0.098%
19:36:00	0.096%	19:51:00	0.097%
19:37:00	0.097%	19:52:00	0.100%
19:38:00	0.100%	19:53:00	0.097%
19:39:00	0.099%	19:54:00	0.096%
Average		**0.098%**	

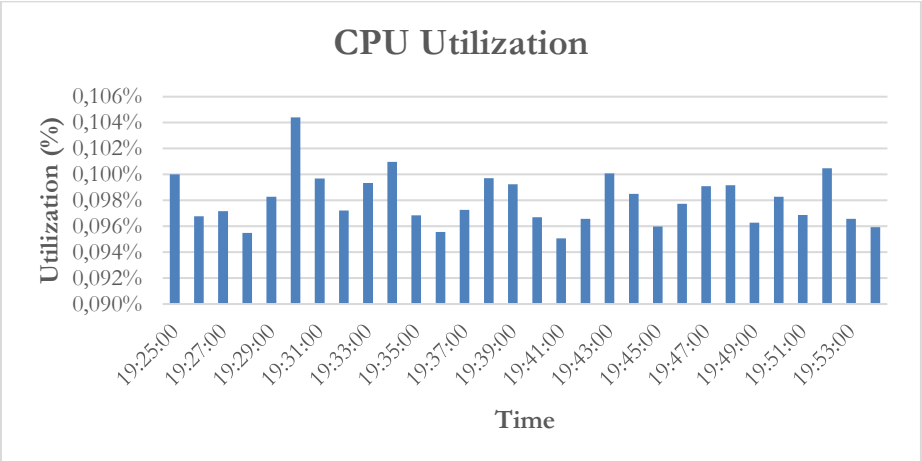

Figure 6. Vertex AI Service CPU Utilization

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

In Figure 6, the graph produces fluctuating values during the test in the time range 19:25:00 to 19:54:00 with a range of values of 0.095% to 0.104%. There is a spike in the 6th minute (19:30:00) with a utilization of 0.104% which can be caused by the use of servers on internal tasks of service infrastructure providers. However, the graph can be said to be stable because the difference between the maximum and minimum values is 0.009%. Overall, the CPU utilization test results on serverless services have a more stable graph than server-based services (Compute Engine) even though the value fluctuates.

2) Memory Utilization

The results of testing the memory utilization performance of the Vertex AI service are shown in Table 8 and the graph in Figure 7. The resulting utilization is consistent throughout the test, which is 0.94%.

Table 8. Vertex AI Service Memory Utilization Measurement Results

Time	Memory Utilization	Time	Memory Utilization
19:25:00	0.94%	19:40:00	0.94%
19:26:00	0.94%	19:41:00	0.94%
19:27:00	0.94%	19:42:00	0.94%
19:28:00	0.94%	19:43:00	0.94%
19:29:00	0.94%	19:44:00	0.94%
19:30:00	0.94%	19:45:00	0.94%
19:31:00	0.94%	19:46:00	0.94%
19:32:00	0.94%	19:47:00	0.94%
19:33:00	0.94%	19:48:00	0.94%
19:34:00	0.94%	19:49:00	0.94%
19:35:00	0.94%	19:50:00	0.94%
19:36:00	0.94%	19:51:00	0.94%
19:37:00	0.94%	19:52:00	0.94%
19:38:00	0.94%	19:53:00	0.94%
19:39:00	0.94%	19:54:00	0.94%
Average		**0.94%**	

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

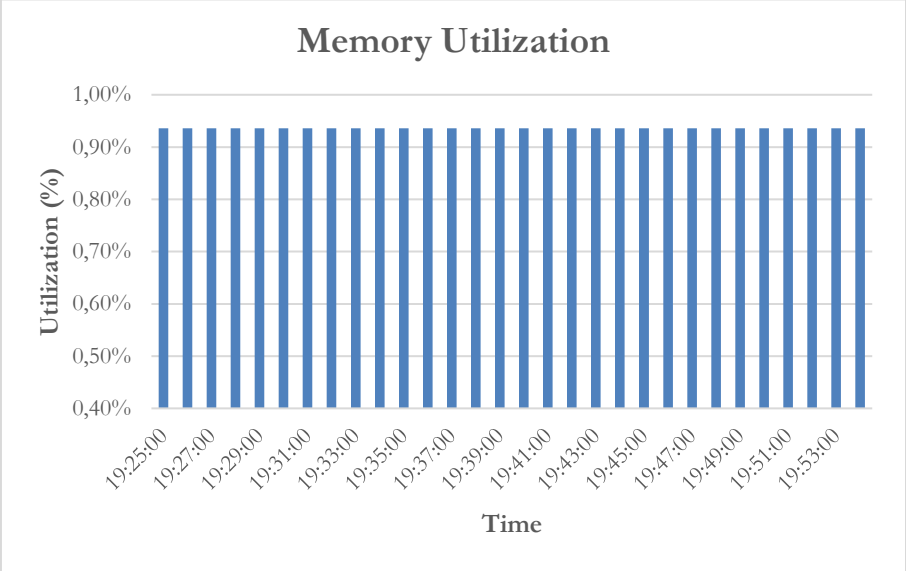

Figure 7. Vertex AI Service Memory Utilization

In Figure 7, it can be seen that the graph produces a consistent value during the test in the time range 19:25:00 to 19:54:00 with a value of 0.94%. The stable memory utilization indicates that the paging process where the empty page is used well for the processing that will run. Paging is the process of allocating memory in fixed-sized units called pages that are used to avoid fragmentation problems [21].

3) Latency

The results of testing the latency performance of the Vertex AI service are shown in Table 9 and the graph in Figure 8. The average latency obtained is 17.07ms with the highest latency of 32.30ms and the lowest latency of 9.31ms.

Table 9. Vertex AI Service Memory Utilization Measurement Results

Time	Latency (ms)	Time	Latency (ms)
19:26:00	19.96	19:43:00	18.60
19:27:00	14.77	19:44:00	11.96
19:28:00	32.30	19:46:00	22.26
19:29:00	10.08	19:47:00	14.45
19:30:00	12.90	19:48:00	21.16
19:31:00	20.16	19:49:00	12.11
19:33:00	21.75	19:50:00	15.50
19:34:00	13.98	19:51:00	24.21

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Time	Latency (ms)	Time	Latency (ms)
19:36:00	19.04	19:53:00	14.48
19:37:00	12.24	19:54:00	9.31
19:39:00	10.92	19:56:00	23.62
19:40:00	13.98	19:57:00	15.18
19:41:00	21.85		
Average		**17.07**	

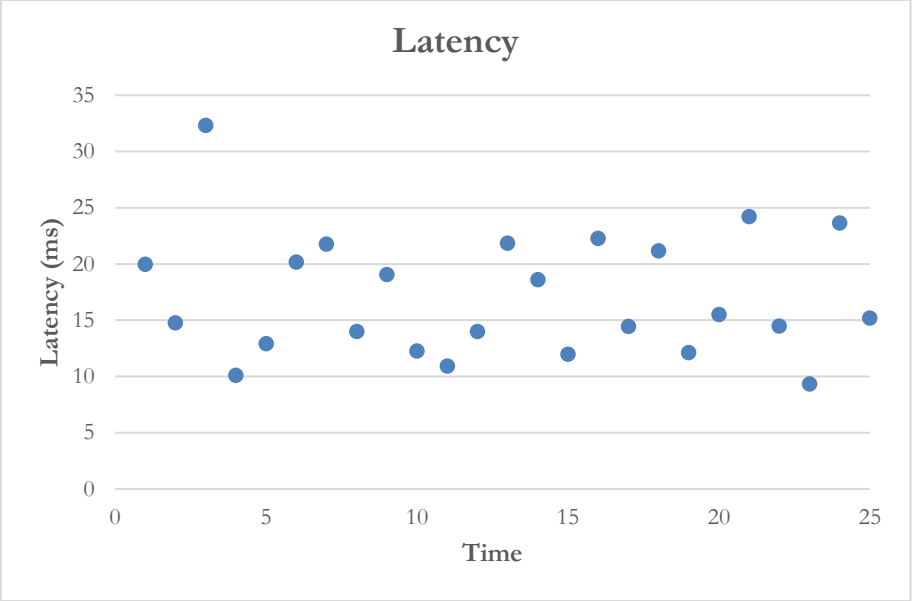

Figure 8. Vertex AI Service Latency

In Figure 8, the Vertex AI service latency test results are quite complete in the time range 19:26:00 to 19:57:00 although it is still not fulfilling the test time. The latency results on the Vertex AI service fluctuate with the highest value of 32.30ms at 19:28:00. The fluctuating graph can be caused by geographical factors resulting in delays through data transfer between regions in the GCP or communication between servers in the zone and user endpoints.

4) Pricing

The cost incurred if all resources on the Vertex AI service are activated for one month to perform deployment on Vertex AI endpoints and testing is estimated at USD 157.882296 or around Rp 2,559,270 with a total cost during testing of

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

$0.1081 or Rp 1,752 based on hourly cost calculations with details as presented in Table 10.

Table 10. Vertex AI Pricing

Resource	Cost per Hour	Total Unit Cost	Estimated Cost per Month
4 vCPU	$0.0047166	$0.0754656	$55.089888
16 GB memory	$0.0351974	$0.1407896	$102.776408
Nearline Storage	-	-	$0.016
Total (one month)			**$157.882296**
Total (during the test)			**$0.1081**

The Vertex AI service costs are quite expensive, similar to the Compute Engine service. The resource cost charged is also separated between vCPU and memory, unlike the Compute Engine service which is charged by machine type. If the resources used in the Vertex AI service are larger, the service cost will increase.

3.3. Performance Comparison of Server-Based and Serverless Services

The results of performance measurements on each service are compared and the network performance is analyzed based on TIPHON standardization. Performance comparisons will be added with considerations based on developer experiences.

1) CPU Utilization

The performance comparison based on CPU utilization of server-based and serverless services is presented in Figure 9 and described in Table 11.

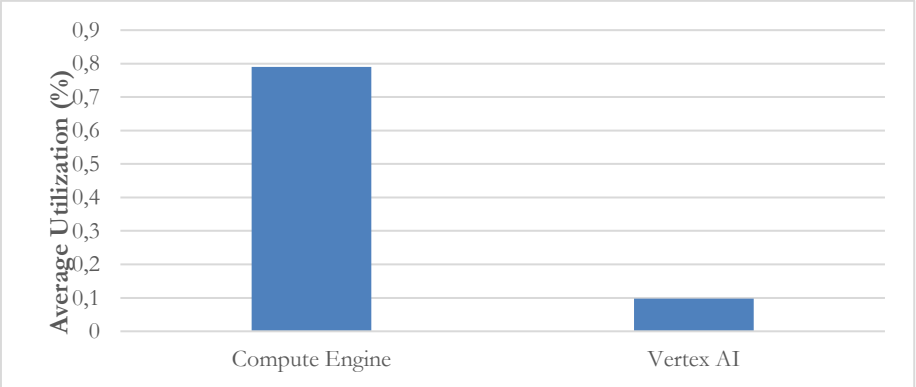

Figure 9. CPU Utilization Performance Comparison

Table 11. CPU Utilization Performance Comparison

Services	Average Utilization (%)
Compute Engine	0.79
Vertex AI	0.10

CPU utilization in both services can be said to be very good although less stable because the number of CPUs used is small and leaves a lot of free space from a total of 4 vCPU cores. Utilization on Vertex AI is smaller than Compute Engine with a utilization result of 0.1%. The smaller the utilization performance, the better the service because there is efficiency and optimization of resources in the deployment so that free space will be greater. Therefore, CPU utilization in Vertex AI service is better than Compute Engine.

2) Memory Utilization

Performance comparison based on memory utilization of server-based and serverless services is presented in Figure 10 and described in Table 12.

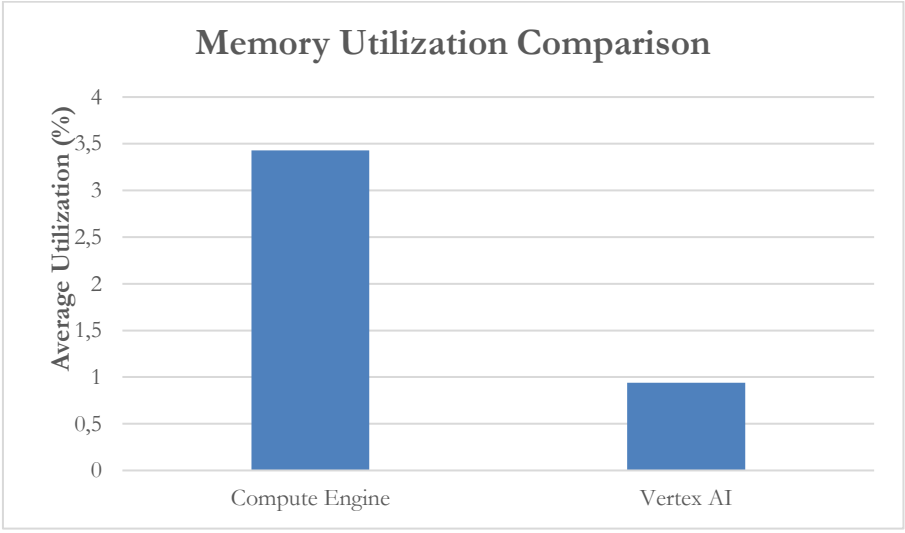

Figure 10. Memory Utilization Performance Comparison

Table 12. Memory Utilization Performance Comparison

Services	Average Utilization (%)
Compute Engine	3.43
Vertex AI	0.94

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Similar to CPU utilization, memory utilization in both services can be said to be very good because the amount of CPU used is small and leaves a lot of free space from a total of 16 GB (100%). Utilization on Vertex AI is smaller than Compute Engine with a utilization result of 0.94%. The smaller the utilization performance, the better the service because there is efficiency and resource optimization so that the free space will be larger. Therefore, memory utilization in the Vertex AI service is better than Compute Engine service.

3) Latency

Performance comparison based on network latency of server-based and serverless services is presented in Figure 11 and described in Table 13.

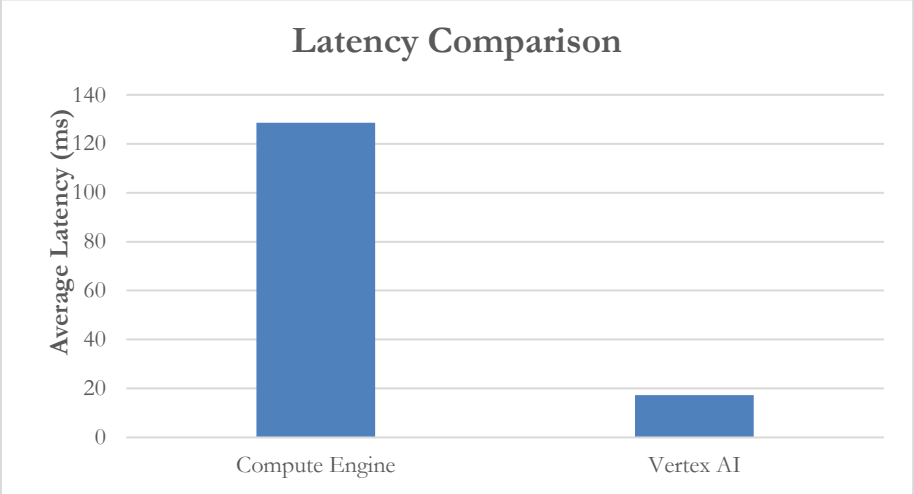

Figure 11. Latency Performance Comparison

Table 13. Latency Performance Comparison

Services	Average Latency (ms)
Compute Engine	128.62
Vertex AI	17.34

The latency value for each service based on TIPHON standardization has good quality because the latency value ranges less than 150ms. The service with the smallest latency level as well as the service with the best performance based on latency is the Vertex AI service. The service with the highest latency level is Compute Engine at 128.62ms due to unstable data results. Even so, the amount of latency is still said to be small based on TIPHON standardization.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

4) Pricing

The price comparison of GCP computing services is presented in Figure 12 and described in Table 14.

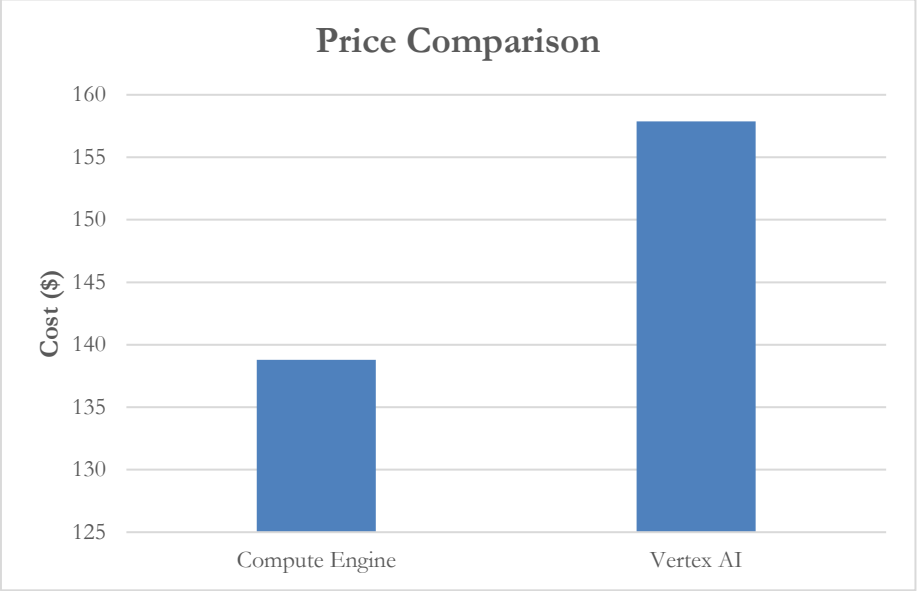

Figure 12. Pricing Comparison

Table 14. Pricing Comparison

Services	Cost for one month		Cost during testing	
	USD ($)	IDR (Rp)	USD ($)	IDR (Rp)
Compute Engine	138.79	2,249,785	0.095	1,540
Vertex AI	157.882296	2,559,270	0.1081	1,752

In terms of cost, users with a free trial account will be given a credit of $300 or Rp 4,757,401 for 90 days of use to access various services on Google Cloud Platform, including computing services. Both services, Compute Engine and Vertex AI are charged if resources are used (pay-per-use). Vertex AI is quite expensive compared to Compute Engine services as the resource price is charged per unit. It is different from Compute Engine which charges resources based on the type of machine. Thus, it can be concluded that the Vertex AI service has good performance but costs more. Meanwhile, the Compute Engine service is cheaper but has a performance that is not better than Vertex AI but not too bad either.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

5) Developer Experiences

Performance evaluation based on developer experiences includes the compatibility of machine learning frameworks in each service, the ease of implementation and the availability of documentation. In terms of machine learning framework compatibility, scikit-learn can be applied to both services assisted by creating a REST API using python to handle HTTP requests and responses in displaying recommendations and dependency requirements needed so that models and APIs can run properly.

Both services can be said to be easy to implement or deploy but do have to pay attention to the requirements specified by GCP such as permissions, firewalls, APIs created, and so on. Server-based services are self-service where the server is configured and managed by the user so that the implementation is more "extra" than serverless services which only need to create source code or deployment function code. The service that must be more careful when deploying is the Vertex AI service because it is very sensitive to the permissions used, one of which is the role "**aiplatform.endpoints.predict**" which plays an important role in displaying model prediction results. The Compute Engine service must also be considered when deploying because the server is created and managed by the user. If the requirements and deployment are not done correctly, such as there are dependencies that are not installed or the wrong web server is configured, the server cannot run properly so that it does not produce the desired response. Both services are easy to deploy because their documentation is available in the GCP. For the basic configuration needed to deploy the ML model, everything is available in the GCP documentation. However, to see the different steps, other sources are needed in order to see which steps are more suitable for deployment. If there are any problems in performing the deployment as well, the documentation and some other sources, such as GitHub, community forums, and others can help. Users should read carefully and read the documentation on the topic thoroughly so that the information can be understood in detail.

3.4. Discussion

The machine learning application was successfully deployed on server-based services, namely Compute Engine and serverless services, namely Vertex AI and produced the expected application response, namely crop seed recommendations based on soil conditions. In conducting deployment and performance monitoring, there are monitoring results with fluctuating values caused by several factors, such as errors and internal server network factors. However, these factors do not affect the application response results. From all the performance comparisons that have been described, a summary of both services performance comparison results is outlined in Table 15.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Table 15. Comparison Summary

Parameters	Best Service	Value
CPU utilization	Vertex AI	0.10%
Memory utilization	Vertex AI	0.94%
Latency	Vertex AI	17.34ms
Pricing	Compute Engine	Rp 2,249,785

From Table 15, the best service for implementing machine learning models based on performance is Vertex AI, as it outperforms all the measured performance parameters. Vertex AI has a very good performance but has a more expensive price than Compute Engine. The Compute Engine service has a price of Rp 2,249,785 while Vertex AI has a price of Rp 2,559,270. Based on ease of implementation, the Vertex AI service is easier to implement because it only needs to deploy the model to the Model Registry and deploy through the Vertex AI endpoint. If users want to use a computing service based on its performance and easy implementation, Vertex AI is the choice. If users want to use computing services based on cost efficiency and flexible servers, then Compute Engine services are suitable.

4. CONCLUSION

This research successfully deployed machine learning-based application through server-based and serverless services. The performance measurement results between the two services show that Vertex AI performance measurement results are better than Compute Engine with CPU utilization of 0.10%, memory utilization of 0.94%, and latency of 17.34ms. However, it is more expensive than the Compute Engine service. Therefore, the Vertex AI service is recommended if users want a service that prioritizes performance. Whereas the Compute Engine service is recommended if users have a limited budget and need flexibility in server management.

REFERENCES

[1] N. Ramsari and A. Ginanjar, 'Implementasi Infrastruktur Server Berbasis Cloud Computing Untuk Web Service Berbasis Teknologi Google Cloud Platform', *Conf. SENATIK STT Adisutjipto Yogyak.*, vol. 7, pp. 169–182, 2022, doi: 10.28989/senatik.v7i0.472.

[2] A. Fadil, 'Strategi Efisiensi Energi dan Penyeimbangan Beban Kerja Layanan Cloud Computing Melalui Konsolidasi Mesin Virtual Dinamis', *Appl. Technol. Comput. Sci. J.*, vol. 3, no. 1, Art. no. 1, 2020, doi: 10.33086/atcsj.v3i1.1680.

[3] 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 on Advances in Computing Research (ACR'23), vol. 700, K. Daimi and A. Al Sadoon, Eds., in Lecture Notes in Networks and Systems, no. ACR 2023, vol. 700. , Cham: Springer Nature Switzerland, 2023, pp. 408–422. doi: 10.1007/978-3-031-33743-7_33.

[4] A. Abraham and J. Yang, 'Analyzing the System Features, Usability, and Performance of a Containerized Application on Serverless Cloud Computing Systems'. Research Square, 2023. doi: 10.21203/rs.3.rs-3167840/v1.

[5] I. Lee and Y. J. Shin, 'Machine learning for enterprises: Applications, algorithm selection, and challenges', *Bus. Horiz.*, vol. 63, no. 2, Art. no. 2, 2020, doi: 10.1016/j.bushor.2019.10.005.

[6] A. Arbain, M. A. Muhammad, T. Septiana, and H. D. Septama, 'Komparasi Implementasi Model Machine Learning Hoax News pada Local dan Cloud Computing Deployment Menggunakan Google App Engine', *J. Inform. Dan Tek. Elektro Terap.*, vol. 10, no. 3, Art. no. 3, 2022, doi: 10.23960/jitet.v10i3.2646.

[7] R. Xu, 'A Design Pattern for Deploying Machine Learning Models to Production'. Computer Science and Information Systems California State University San Marcus, 2020.

[8] W. Zhu *et al.*, 'QuakeFlow: a scalable machine-learning-based earthquake monitoring workflow with cloud computing', *Geophys. J. Int.*, vol. 232, no. 1, Art. no. 1, 2022, doi: 10.1093/gji/ggac355.

[9] F. Rahman, 'Serverless Cloud Computing: A Comparative Analysis of Performance, Cost, and Developer Experiences in Container-Level Services', Thesis, Aalto University School of Science, 2023.

[10] I. P. T. K. Wiranata, A. A. I. I. Paramitha, and I. P. Satwika, 'Analisis Perbandingan Performa App Engine dan Compute Engine Pada Google Cloud Platform dalam Memprediksi Penyakit Mata dengan Model CNN', *JATI J. Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3968–3977, 2023.

[11] Y. Wu, T. T. A. Dinh, G. Hu, M. Zhang, Y. M. Chee, and B. C. Ooi, 'Serverless Data Science - Are We There Yet? A Case Study of Model Serving', in *Proceedings of the 2022 International Conference on Management of Data*, Philadelphia, PA, USA: ACM, 2022, pp. 1866–1875. doi: 10.1145/3514221.3517905.

[12] H. B. Barua, 'Data science and Machine learning in the Clouds: A Perspective for the Future', no. arXiv:2109.01661. arXiv, 2021. doi: 10.48550/arXiv.2109.01661.

[13] M. Eisa, M. Younas, K. Basu, and I. Awan, 'Modelling and Simulation of QoS-Aware Service Selection in Cloud Computing', *Simul. Model. Pract. Theory*, vol. 103, p. 102108, 2020, doi: 10.1016/j.simpat.2020.102108.

[14] A. K. C, S. B, and N. R, 'Resource Utilization Prediction in Cloud Computing using Hybrid Model', *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 4, 2021, doi: 10.14569/IJACSA.2021.0120447.

[15] S.-Y. Hsieh, C.-S. Liu, R. Buyya, and A. Y. Zomaya, 'Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers', *J. Parallel Distrib. Comput.*, vol. 139, pp. 99–109, 2020, doi: 10.1016/j.jpdc.2019.12.014.
[16] A. Ibrahim, A. H. Alang, Madi, Baharuddin, M. A. Ahmad, and Darmawati, *Metodologi Penelitian*. Jakarta: Gunadarma Ilmu, 2018.
[17] B. Erinle, *Performance Testing with JMeter 3*, Third. Birmingham: Packt Publishing, 2017.
[18] Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta, 2013.
[19] B. Arifwidodo, V. Metayasha, and S. Ikhwan, 'Analisis Kinerja Load Balancing pada Server Web Menggunakan Algoritma Weighted Round Robin pada Proxmox VE', *J. Telekomun. Dan Komput.*, vol. 11, no. 3, Art. no. 3, 2021, doi: 10.22441/incomtech.v11i3.11775.
[20] A. Abouaomar, S. Cherkaoui, Z. Mlika, and A. Kobbane, 'Resource Provisioning in Edge Computing for Latency Sensitive Applications'. arXiv, 2022. Accessed: Jun. 22, 2024.
[21] C. J. Theaker and G. R. Brookes, 'Memory Management — Paging Algorithms and Performance', in *Concepts of Operating Systems*, C. J. Theaker and G. R. Brookes, Eds., London: Macmillan Education UK, 1993, pp. 77–102. doi: 10.1007/978-1-349-11511-2_6.