

Performance Analysis of Virtual Learning System: A Case Study of ANGKASA

Stevan Del Arisandi, Nungki Selviandro, Kusuma Ayu Laksitowening, Dana Sulistyio Kusumo

School of Computing, Telkom University, Bandung, Indonesia

delarisandi@student.telkomuniversity.ac.id, nselviandro@telkomuniversity.ac.id, ayu@telkomuniversity.ac.id, danakusumo@telkomuniversity.ac.id

Abstract— Virtual e-learning system is a learning activity using electronic media such as computers and the internet as main tool. ANGKASA is a provider platform of e-learning system using cloud computing technology. E-learning system provides many benefits such as flexibility and efficiency for learning activity. However, there are some disadvantages of the usage of e-learning system, i.e., there are still many institutions especially educational institutions that still implementing conventional-based of e-learning system which considered ineffective both in terms of cost and time for infrastructure. These disadvantages can be resolved by using cloud computing system. In this study, we are implementing cloud computing system using virtualization technology. There are two methods of virtualization used in this study, full virtualization or VM and container virtualization. In this study, we are comparing the performance of these two methods of virtualization using Apache Benchmarking tool. The study shows that the container virtualization method is better than the VM virtualization method in terms of serving requests for virtual e-learning system.

Keywords— Performance Analysis, E-Learning System, Apache Benchmarking, Virtualization

I. INTRODUCTION

The implementation of virtual e-learning system nowadays has been an excellent solutions since the pandemic came. E-learning system provides benefits like flexibility and efficiency thanks to its system where lecturers and students use computers and internet to do learning activities and don't have to meet face to face. Also students are free to study at any time by accessing learning materials through the e-learning system [1].

However, to be able to develop a virtual e-learning system, educational institutions such as school, university, etc. they must have at least specialized technicians and expertise not just to develop but also operate and manage the server of virtual e-learning system itself. This method of implementation of virtual e-learning system is called conventional e-learning. From this situations, problems began to arise in the implementation of conventional e-learning such as infrastructure, human resources, and maintenance [2]. And also all of these problems of course requires a big amount of budget, which is not all of educational institutions have it.

There were some previous studies in order to solve this problems by implementing cloud computing technology to e-

learning platforms such as Chuang, Chang, and Sung [3], Pocatilu [4], and Nungki Selviandro et al. [1][2]. A research conducted by Nungki Selviandro et al. [1][2] proposed a cloud-based e-learning architecture which consists of six layers, namely: infrastructure layer, platform layer, application layer, server layer, access layer, and user layer [2]. This cloud-based architecture is based on principle of acting as a provider of e-learning services for educational institutions in an effective and at least reducing the cost and time required for developing an e-learning system infrastructure.

Based on that study by Nungki Selviandro et al. [1][2], there was a project that was initiated to create a solution to the problems described earlier. This project is called ANGKASA. ANGKASA project was initiated to help educational institutions to have their own cloud-based e-learning system with less cost, time and skills to develop it by implementing cloud computing system using virtualization technology.

In this study, we aim to do a performance analysis research of cloud-based virtual e-learning system used in ANGKASA project. We are using two methods of virtualization, full virtualization or VM and container virtualization. We are comparing these two methods of virtualizations in terms of performance using Apache Benchmarking tool to determine which method of virtualization is better for cloud-based virtual e-learning system.

This paper is organised as the following. In this section 2 of this paper, we discuss about related works and literature needed for this study such as about virtual e-learning system, cloud-based e-learning implementation, virtualization technology, hypervisor and apache benchmark. In section 3, we explain the design of system of this study from the start to end. In section 4 we present and also explain about the both VM and container virtualization performance analysis experiment and its result. Finally in section 5, we explain the final conclusions of the experiment result and we also discuss about future works.

II. RELATED WORKS

A. Virtual E-Learning System

Virtual e-learning system is a learning activity that uses technology as a medium of learning and interaction between teachers and students [5]. Virtual e-learning system aims to

support conventional learning by providing more benefits for both teachers and students. With virtual e-learning system, students can access learning materials and resources uploaded by teachers anytime and anywhere using electronic media such as computers and similar devices [1].

B. Cloud-based E-Learning Implementation

According to the National Institute of Standards and Technology (NIST), cloud-based e-learning applies cloud computing model that allows access to the e-learning system as needed or on-demand network access to shared computing resources provided by e-learning service providers [2][6]. By using cloud computing model, one or more educational institutions can use an e-learning system service that provided by e-learning service provider and so these institutions can save costs in terms of infrastructure and maintenance because all of this has been provided by the e-learning service provider while the institution only needs to pay to rent e-learning services in accordance with the use of the service (pay-on-demand).

Based on the cloud-based e-learning architecture proposed on study by Nungki Selviandro et al. [2], there is six layers, namely: infrastructure layer, platform layer, application layer, server layer, access layer, and user layer as shown in Figure 1 [2].

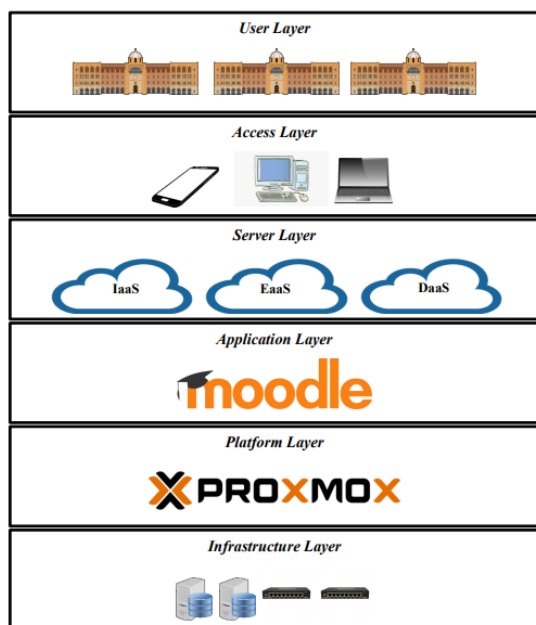


Figure 1. The cloud-based e-learning architecture proposed by Nungki Selviandro et al. [2].

The following is an explanation of each layer:

- Infrastructure layer: consists of physical infrastructure or hardware such as servers, network devices, storage, etc.
- Platform layer: consists of hypervisor that controls virtualization of virtual machines for e-learning platforms.

- Application layer: consists of the e-learning platform itself that can be used by teachers and students to interact with each other, for example Moodle.
- Server layer: consists of services provided by the e-learning platform, namely EaaS (E-learning as a Service), DaaS (Data as a Service), IaaS (Infrastructure as a Service).
- Access layer: consists of devices that can access e-learning services, such as computers, smartphones, notebooks, etc.
- User layer: consists of educational institutions as users of cloud-based e-learning services themselves.

C. Virtualization Technology

Virtualization technology is an enabling technology allowing the creation of multiple operating systems to run on the same physical platform [7]. There are two methods of virtualization: VM and container virtualization.

First of is VM or full virtualization, VM allows a host device to simulate more than one guest operating system called virtual machines at the same time controlled by a hypervisor [8]. The host device will also share the use of resources and hardware such as the use of memory, CPU, network, etc. on each simulated virtual machine. The following is the architecture of VM shown in Figure 2.

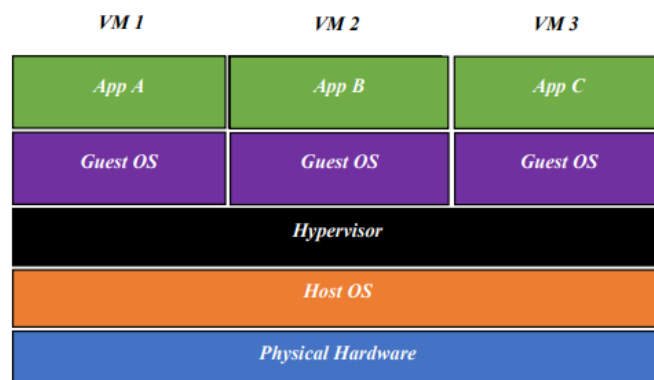


Figure 2. The architecture of VM virtualization

Unlike full virtualization, container virtualization does not simulate a full operating system. Container virtualization only isolates applications with their dependencies and resource usage such as memory usage, CPU, network, etc. by utilizing kernel features to perform OS-level equivalent virtualization into a container. In container virtualization, all virtualized containers can only run on the same OS as the host OS and each container can share an OS kernel on the same machine. In contrast to full virtualization, which shares each of its resources on each VM / guest OS as shown in Figure 3 [9][10].

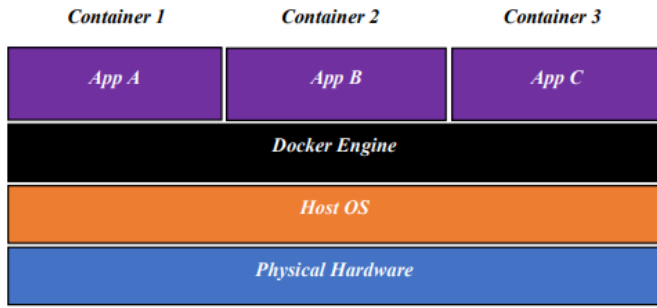


Figure 3. The architecture of container virtualization

D. Hypervisor

Hypervisor is a software that allows host devices to share resource and hardware usage such as memory usage, CPU, network, etc. on each simulated virtual machine [11]. Hypervisor is categorized into 2 types, namely hypervisor type 1 which is known as native/bare-metal and hypervisor type 2.

Bare metal hypervisor or type 1 is a type of hypervisor that is applied directly to a piece of hardware by using the entire storage capacity of that hardware [12]. This type of hypervisor is usually installed directly on the server device. Examples of type 1 hypervisors such as VMware ESXi, oVirt, and Proxmox Virtual Environment (PVE) [13]. As shown in Figure 4, the architecture of the hypervisor type 1, where the hypervisor is directly applied to a piece of hardware that can create and simulate multiple guest operating systems as virtual machines.

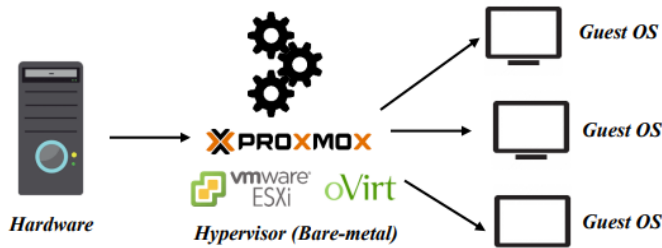


Figure 4. Hypervisor type 1 architecture (bare-metal)

Hypervisor type 2 is a desktop-oriented type of hypervisor, it means that the hypervisor is implemented in a host operating system and virtualization is carried out through the application platform on the host operating system. Examples include Oracle VM VirtualBox and VMware Workstation [13]. As shown in Figure 5, the architecture of a type 2 hypervisor, where the hypervisor runs on a host OS, and can simulate and control multiple guest OSes as virtual machines.

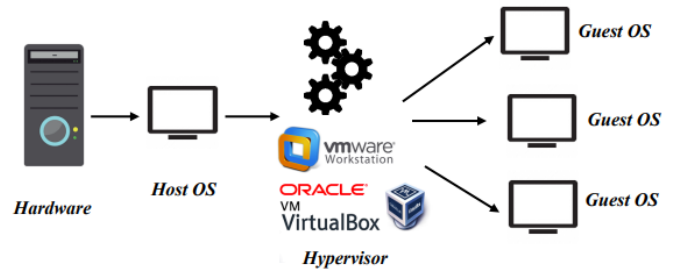


Figure 5. Hypervisor type 2 architecture

E. Apache Benchmark

Based on research on performance analysis of e-learning websites on open source virtualization by Edy Kristianto, Apache Benchmark is one indicator or benchmark that can be analyzed on e-learning performance [14].

Apache is a web server platform that can accept requests and serve all web assets and content via HTTP which is open source. So the Apache Benchmark is a benchmark for monitoring the installation of the Apache platform on a web server, and also at the same time being able to analyze how many requests are received at the same time that the web server can serve [14][15].

Cloud-based e-learning is also a web server that is able to receive and serve requests from students, teachers, or admins as users, the Apache Benchmark can be used as a benchmark or main indicator of cloud-based e-learning performance itself as web servers. To test the performance of cloud-based e-learning with the Apache Benchmark as a benchmark, tools are needed to test it and one of these tools is the Linux-based 'ab' command.

III. DESIGN OF SYSTEM

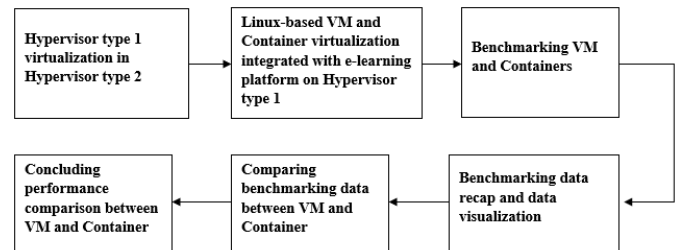


Figure 6. Design of System

In this study, due to limited availability of infrastructure, the type 1 hypervisor which will control the virtualization of the cloud-based e-learning platform will be virtualized on the type 2 hypervisor. When hypervisor type 1 is already virtualized on hypervisor type 2, Linux-based operating system that already integrated with an e-learning platform will be installed on hypervisor type 1. These Linux-based OS will be installed by using two methods of virtualization, VM and container virtualization.

After everything is set, the next step is benchmarking the performance of these two methods of virtualization. After that, the benchmarking data will be recapped and will be visualized

in the form of a graph so that it is easy to understand and analyze.

The next step is to compare the benchmarking data that already recapped and visualized between VM and container virtualization. Last step is to make a final conclusion from the benchmarking data to determine which method of virtualization is better in terms of their performance.

IV. EXPERIMENT AND RESULT

A. Experimental Setup

The location of research is using personal computer with the specifications of processor Intel(R) Core(TM) i7-4720HQ CPU x64-based processor with 2.60GHz, 12 GB RAM, and 64-bit Microsoft Windows 10 Pro operating system.

Hypervisors used in this study are Proxmox VE as hypervisor type 1 and VMware Workstation as hypervisor type 2. The Linux-based operating system used in this study is Turnkey Linux that already integrated with Moodle e-learning platform. TurnKey Linux will be installed on Proxmox VE by using two methods of virtualization, VM and container virtualization. The following is the architecture design of this experiment shown in Figure 7.

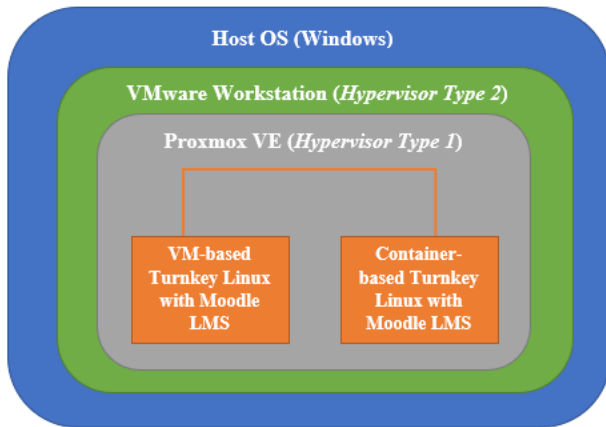


Figure 7. The experiment architecture design

B. Benchmarking Tools

Benchmarking tool used in this study is Apache Benchmark tool. Apache Benchmark is a tool for monitoring the installation of the Apache platform on a web server, and also at the same time being able to analyze how many requests are received at the same time that the web server can serve.

Apache Benchmark is used in the form of a Linux-based command called 'ab'. Apache Benchmark has two main parameters in its use, the first is the total requests that need to be served in the benchmarking session marked with the '-n' command. The second is the concurrency, denotes the number of multiple requests to perform at a time, marked with the command '-c'.

An example of its use is if we want to test the performance of the site <https://lmslorem.angkasalms.id:8080> with a total parameter of 100 requests that need to be served and the concurrency or number of requests that need to be served at

one time is 10 requests, then the command will be written like this 'ab -n 100 -c 10 <https://lmslorem.angkasalms.id:8080/>'.

C. Data Collection

In this study, we are using total 100 request that need to be served with 10, 25, 50, 75 and 100 concurrency or number of requests that need to be served at one time as parameter. The variables used in this study are time taken for tests, request per second, time per request for each concurrency, time per request for across all requests and transfer rate.

First variable is time taken for tests, the following is the data recap of time taken for tests shown in Table 1.

TABLE 1. TIME TAKEN FOR TESTS DATA RECAP

Time Taken for Tests		
Concurrency	Container	VM
10	2.450 sec	3.008 sec
25	2.510 sec	3.099 sec
50	2.577 sec	3.321 sec
75	2.649 sec	3.814 sec
100	2.798 sec	4.203 sec

The following is the data visualization of time taken for tests comparison shown in Figure 8.

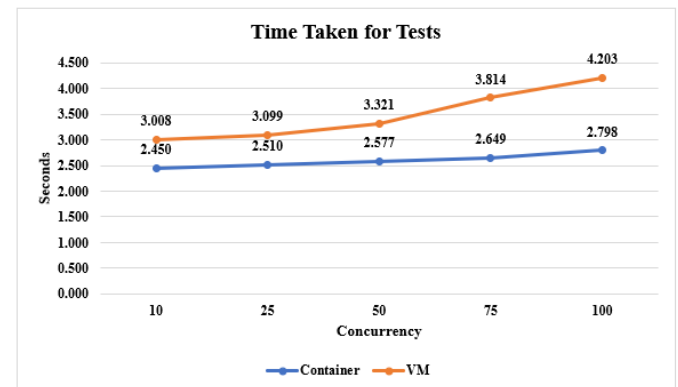


Figure 8. Time Taken for Tests Comparison

As shown in Figure 8, VM virtualization is taking more time to complete the benchmarking tests compared with container virtualization.

The following is the data recap of total request per second to be served shown in Table 2.

TABLE 2. REQUEST PER SECOND DATA RECAP

Request per Second		
Concurrency	Container	VM
10	40.82 /sec	33.24 /sec
25	39.84 /sec	32.27 /sec
50	38.80 /sec	30.11 /sec
75	37.74 /sec	26.22 /sec
100	35.75 /sec	23.79 /sec

The following is the data visualization of total request per second to be served shown in Figure 9.

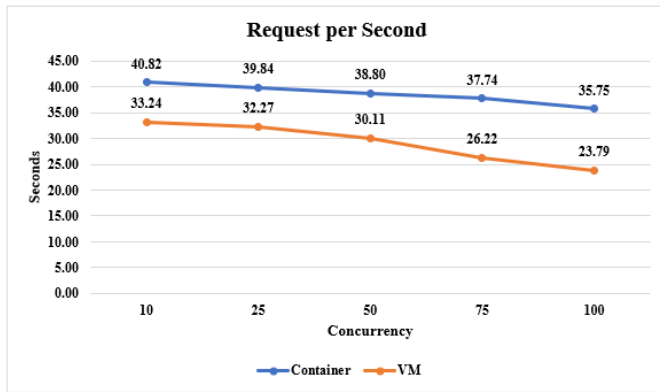


Figure 9. Request per Second Comparison

As shown in Figure 9, container virtualization served more requests per second compared with VM virtualization.

The following is data recap of the average time spent per request for each concurrency calculated with formula: (concurrency * timetaken * 1000 / done) shown in Table 3.

TABLE 3. TIME PER REQUEST FOR EACH CONCURRENCY DATA RECAP

Time per Request for Each Concurrency		
Concurrency	Container	VM
10	244.973 ms	300.803 ms
25	627.585 ms	774.809 ms
50	1288.667 ms	1660.526 ms
75	1987.114 ms	2860.342 ms
100	2797.585 ms	4202.643 ms

The following is the data visualization of the average time spent per request for each concurrency shown in Figure 10.

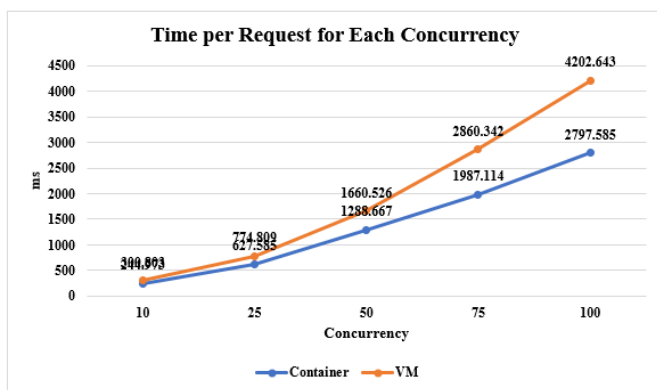


Figure 10. Time per Request for Each Concurrency Comparison

As shown in Figure 10, VM virtualization is taking more time to serve requests for each concurrency compared with container virtualization.

The following is data recap of the average time spent per request for across all requests calculated with formula: (timetaken * 1000 / done) shown in Table 4.

TABLE 4. TIME PER REQUEST FOR ACROSS ALL REQUESTS DATA RECAP

Time per Request for Across All Requests		
Concurrency	Container	VM
10	24.497 ms	30.080 ms
25	25.103 ms	30.992 ms
50	25.773 ms	33.211 ms
75	26.495 ms	38.138 ms
100	27.976 ms	42.026 ms

The following is the data visualization of the average time spent per request for across all requests shown in Figure 11.

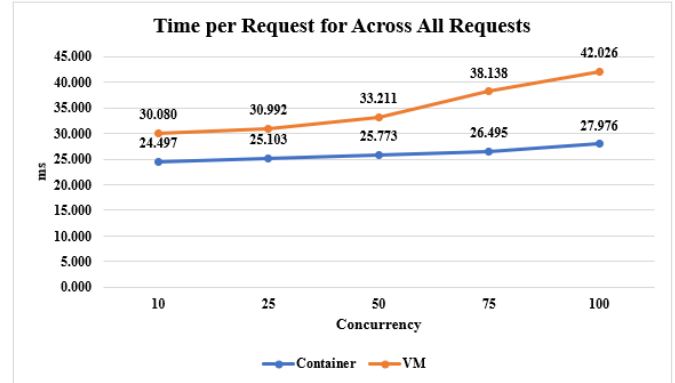


Figure 11. Time per Request for Across All Requests Comparison

As shown in Figure 11, VM virtualization is also taking more time to serve requests for across all requests compared with container virtualization.

In this study, we also use the variable of transfer rate of data with the total size of data transferred is 2916100 bytes and with total of 2855900 bytes of HTML data transferred.

The following is data recap of transfer rate of data calculated by the formula: (totalread / 1024 / timetaken) shown in Table 5.

TABLE 5. TRANSFER RATE DATA RECAP

Transfer Rate		
Concurrency	Container	VM
10	1162.48 Kbytes/sec	946.72 Kbytes/sec
25	1134.41 Kbytes/sec	918.86 Kbytes/sec
50	1104.92 Kbytes/sec	857.49 Kbytes/sec
75	1074.83 Kbytes/sec	746.70 Kbytes/sec
100	1017.93 Kbytes/sec	677.61 Kbytes/sec

The following is the data visualization of transfer rate of data shown in Figure 11.

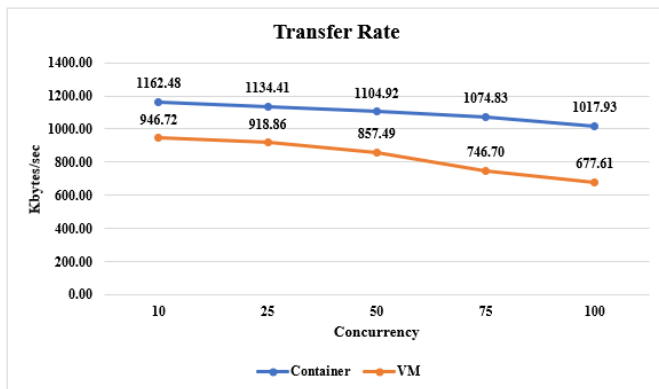


Figure 12. Transfer Rate Comparison

As shown in Figure 11, container virtualization has higher transfer rate of data compared with VM virtualization.

V. CONCLUSION AND FUTURE WORKS

Based on the experiment and comparing analysis results of benchmarking performance between VM virtualization and container virtualization in terms of serving requests for virtual e-learning system and by using five different variables, we get the results that container virtualization is more effective than VM virtualization in terms of serving request per second and its transfer rate. Also in terms of time taken, container virtualization is taking less time to complete the tests, complete the request for each concurrency and complete the request for across all requests rather than VM virtualization and we consider it is timely very efficient.

We assume that this happens because container virtualization does not simulate a full operating system and only isolates applications with their dependencies so that resource usage becomes more effective such as memory usage, CPU, network, etc. unlike VM virtualization, which does the opposite as mentioned in related works and literature in section 2.

So we can conclude that container virtualization is better than VM virtualization in terms of serving request for virtual e-learning system.

In future works, we hope to be able to implement this research more deeply in the ANGKASA project by using a

more sophisticated setup and tools that have more variables that can be analyzed so that we can get more optimal and in-depth analysis results.

REFERENCES

- [1] Selviandro, N., & Hasibuan, Z. A. (2013, March). Cloud-based e-learning: A proposed model and benefits by using e-learning based on cloud computing for educational institution. In *Information and Communication Technology-EurAsia Conference* (pp. 192-201). Springer, Berlin, Heidelberg.
- [2] Selviandro, N., Suryani, M., & Hasibuan, Z. A. (2014, February). Open learning optimization based on cloud technology: case study implementation in personalization E-learning. In *16th International Conference on Advanced Communication Technology* (pp. 541-546). IEEE.
- [3] Chuang, S. M., Chang, K. E., & Sung, Y. T. (2011, September). The cost effective structure for designing hybrid cloud based enterprise E-learning platform. In *2011 IEEE International Conference on Cloud Computing and Intelligence Systems* (pp. 523-525). IEEE.
- [4] Pocatilu, P. (2010). Cloud computing benefits for e-learning solutions. *Oeconomics of Knowledge*, 2(1), 9.
- [5] Sangrà, A., Vlachopoulos, D., & Cabrera, N. (2012). Building an inclusive definition of e-learning: An approach to the conceptual framework. *International Review of Research in Open and Distributed Learning*, 13(2), 145-159.
- [6] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
- [7] Kumar, R., & Charu, S. (2015). *An importance of using virtualization technology in cloud computing*. *Global Journal of Computers & Technology*, 1(2).
- [8] Pousa, D., & Rufino, J. (2017). Evaluation of type-1 hypervisors on desktop-class virtualization hosts. *IADIS-JOURNAL ON COMPUTER SCIENCE AND INFORMATION SYSTEMS*, 12(2), 86-101.
- [9] da Silva, V. G., Kirikova, M., & Alksnis, G. (2018). Containers for Virtualization: An Overview. *Appl. Comput. Syst.*, 23(1), 21-27.
- [10] Zhang, Q., Liu, L., Pu, C., Dou, Q., Wu, L., & Zhou, W. (2018, July). A comparative study of containers and virtual machines in big data environment. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)* (pp. 178-185). IEEE.
- [11] VMware, Inc. (2021). What is a Hypervisor? | VMware Glossary. [Online] Available at: <https://www.vmware.com/topics/glossary/content/hypervisor>.
- [12] Addaffi, A. G. (2016). Implementasi Dan Analisa Perbandingan Kinerja Virtualisasi Server Menggunakan Proxmox Dan Openstack (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).
- [13] Pousa, D., & Rufino, J. (2017). Evaluation of type-1 hypervisors on desktop-class virtualization hosts. *IADIS-JOURNAL ON COMPUTER SCIENCE AND INFORMATION SYSTEMS*, 12(2), 86-101.
- [14] Kristianto, E. (2014, August). The performance of e-learning website on open source virtualization. In *2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA)* (pp. 109-112). IEEE.
- [15] Hernandez, J. (2019). Introduction to Apache Web Server. [Online] Available at: <https://jovansh.medium.com/introduction-to-apache-web-server-2138af643039>.