



Implementation of proxmox server monitoring system with laravel and vue.js

Ariel Thymoti Irmawan¹, Raymond Maulany²

^{1,2}Faculty of Information Technology, Universitas Advent Indonesia

ARTICLE INFO

Article history:

Received Feb 21, 2024
Revised Feb 23, 2024
Accepted Feb 28s, 2024

Keywords:

Availability;
Proxmox;
Performance;
Server Monitoring;
Virtualization.

ABSTRACT

Large organizations now heavily rely on Information Technology and Telecommunications (ICT) for their operations. Server availability is crucial in managing data centers and applications, but the increase in physical servers often involves high costs. Therefore, virtualization technology has emerged as an efficient solution. The research aims to develop the "Proxmox Server Monitoring System" to automatically monitor the availability and performance of Proxmox VE servers with real-time notifications in case of failures, enhancing efficiency and availability. This research is expected to improve the utilization of Proxmox VE server resources, provide automated monitoring, detect failures faster, and reduce downtime, thereby better supporting organizational operations. The potential contribution of this research lies in optimizing server resource usage with virtualization technology, improving server infrastructure efficiency and availability, and reducing the risk of failures and downtime.

This is an open access article under the [CC BY-NC](https://creativecommons.org/licenses/by-nc/4.0/) license.



Corresponding Author:

Ariel Thymoti Irmawan¹
Faculty of Information Technology,
Universitas Advent Indonesia,
Jl. Kolonel Masturi No.288, Cihanjuang Rahayu, Kec. Parongpong, Kabupaten Bandung Barat,
Jawa Barat 40559, Indonesia,
e-mail: 2081001@unai.edu, 2raymond@unai.edu

1. INTRODUCTION

The development of information and communication technology (ICT) has had a significant impact on various organizations, including in server management (Serrat, 2023). A server is a key component in the IT infrastructure that supports the operations and services of an organization (Marin et al., 2022). One increasingly popular virtualization platform in the business world is Proxmox Virtual Environment (Proxmox VE), an open-source solution widely used by organizations to manage their servers (Harfadzi et al., 2016). Proxmox VE has various advantages, such as scalability and efficiency (Ally, 2018). However, the challenge lies in server monitoring and management (Harijanto et al., 2015). The availability and performance of Proxmox VE servers are critical factors affecting the operations and quality of services hosted on them. Amid the rapid development of the digital era, effective server monitoring becomes a prerequisite for maintaining reliable services and organizational productivity (Persaud Nadinand Dagher, 2021). Proxmox Virtual Environment (Proxmox VE) is an open-source virtualization platform that combines container virtualization with LXC and virtual

machine (VM) virtualization with KVM. LXC allows the creation of lightweight Linux containers, sharing the kernel with the host, while KVM allows running virtual machines with stronger isolation (Prasandy et al., 2015). Proxmox VE provides a web interface for management and supervision, including cluster features for central management and efficient resource allocation across multiple nodes (Yudha et al., 2019). With this combination of technologies, Proxmox VE provides high flexibility in a scalable and reliable virtualization environment, as well as optimizing resource usage (Algarni et al., 2018). In the context of this research, the main goal of the researcher is to discuss the issue of availability and performance of Proxmox VE servers in the implementation of a server monitoring system. This research focuses on the development of a monitoring system capable of providing real-time notifications when server failures occur, analyzing the level of failure, and optimizing server resource usage. The purpose of this application development is to increase the efficiency and reliability of server infrastructure by minimizing downtime and maximizing productivity. Thus, this application is expected to provide significant benefits for organizations in terms of performance improvement and operational cost reduction. In addition, this application also helps in detecting and preventing potential problems before they impact services, thus enabling organizations to respond more quickly and effectively to these issues. As the complexity of server infrastructure increases, so does the risk of server failure (S. M. Ali et al., 2022). Failures that are not detected quickly can have serious impacts on service availability and organizational productivity. Therefore, this research is directed to provide a solution by developing the "Proxmox Server Monitoring System." This system is expected to automatically monitor the availability and performance of Proxmox VE servers, provide real-time notifications when failures occur, and overall improve the efficiency and availability of Proxmox VE server infrastructure. This research aims to reduce the risk of server failures that can disrupt organizational operations. It builds upon previous work on Proxmox server monitoring solutions, GPL v2 licensed virtualization products, tested on the UPT-TKP virtual server of the Riau Province Education Department (E. Ali et al., 2015). However, that research did not yet have a comprehensive and efficient monitoring system to monitor a complex server structure, which can increase the risk of server failure levels. Therefore, the aim of this research is to develop a more comprehensive and effective monitoring system for Proxmox VE servers using modern technologies such as the Laravel framework for the backend and Vue.js for the frontend.

2. RESEARCH METHOD

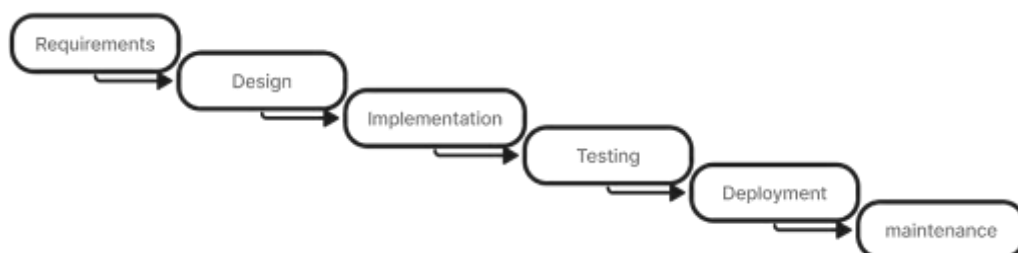


Figure 1. Waterfall model (Sasmito et al., 2017)

This research uses the Waterfall model, a sequential software development model (Silvi Purnia et al., 2019; Yadav et al., 2019). The main stages include: Analysis, Design, Implementation, Testing, Deployment, and Maintenance. This model is chosen because of its structured and sequential nature, suitable for this project (Aroral, 2021; Ciccozzi et al., 2019). This research will focus until the Testing stage. This model allows strict control at each stage (Kute et al., 2014). In addition, the development of the "Proxmox Server Monitoring System" application will use modern technology (Koç et al., 2021),

such as the Laravel framework for the backend (Richey et al., 2014). This will accelerate development and ensure the application conforms to best practices (Van, 2017). The research methodology is designed to implement the "Proxmox Server Monitoring System," aiming to automatically monitor the availability and performance of Proxmox VE servers (Elvis Pawan et al., 2021; Mishra et al., 2023). Inspiration comes from a previous study (E. Ali et al., 2015). Data collection is done through the monitoring system at certain time intervals. Data analysis will focus on measuring server availability and performance. The ethical aspects of research are guaranteed through the approval of the server owner. However, the limitations of the research involve a focus on the scope of Proxmox VE servers.

3. RESULT AND DISCUSSIONS

3.1 Use Case Diagram And Activity Diagram

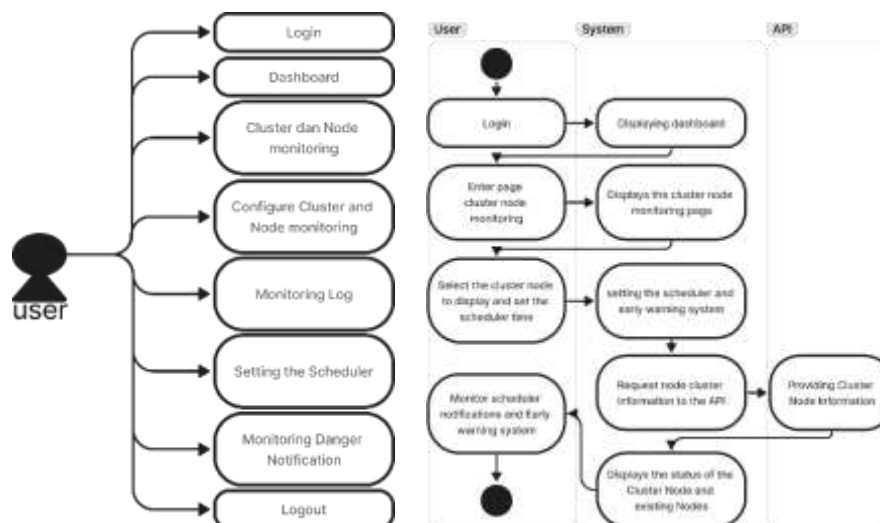


Figure 2. Use Case Diagram and Activity Diagram

A use case diagram is a UML diagram that depicts the communication between the system and the involved parties (E. Ali et al., 2015). This diagram shows the roles within the system and the interaction of the related parties (Kausar Bagwan et al., 2019). The system to be developed only has one user. The use case diagram in Figure. 2 shows the user's interaction with the Proxmox Server Monitoring System. It includes use cases like Login, Dashboard, Monitor Cluster, Configure Cluster, View Monitoring Log, Set Server Check Schedule and EWS, and View Error Notifications. These use cases form a workflow for efficient server monitoring and management

An Activity Diagram is a visual tool used to illustrate the workflow or activities in a system (Hamza et al., 2019). In the context of this research, the Activity Diagram is used to visualize the workflow in the "Proxmox Server Monitoring System" application. The Activity Diagram starts with "Login," continues to "Dashboard," and "Cluster and Node Monitoring." Users can set monitoring settings in "Configure Cluster and Monitoring," check activity records in "Log Monitoring," set schedules in "Set Scheduler," and view error notifications in "Error Notification Monitoring." This diagram depicts the workflow of the application and user interaction.

3.2 Scheduler

The Windows Task scheduler manages the execution of scheduler scripts that run every minute when the application is launched. This process aims to monitor and process data automatically, reducing infrastructure costs and enhancing server efficiency.

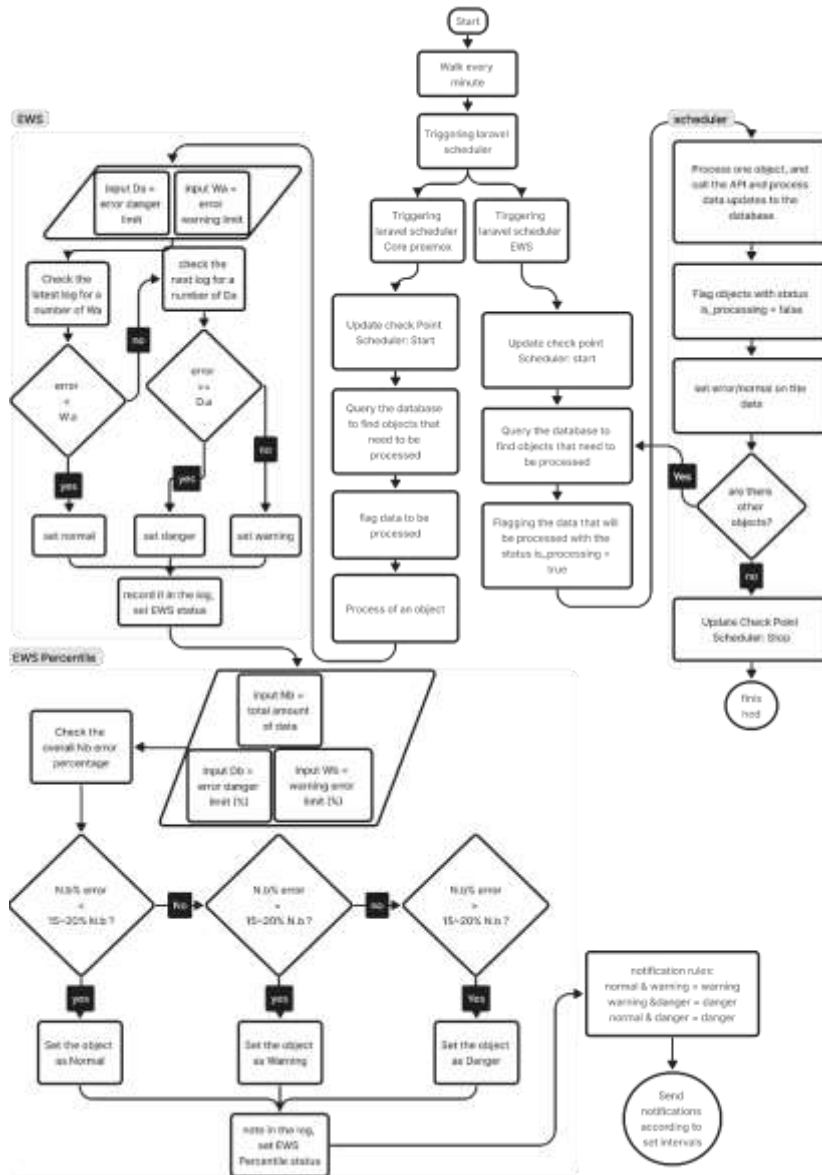


Figure 4. Scheduler

3.3 Checking Scheduler

This process begins by updating the scheduler checkpoint to "start" when a scheduler that needs to be run is found. The application queries the database to find data that requires updating or processing. Data with the status "is_processing = true" indicates that the processing is ongoing. Through the API, the data is updated and its status is changed back to "is_processing = false". The data status (warning or normal) is determined based on information from the API. If no other objects require processing, the scheduler is updated to "stop".

3.7 Discussions

a. Inteface Implmentation

The author has successfully implemented all the designed functions into the application. The interface includes main features such as login, dashboard, monitoring and configuration of Proxmox cluster, monitoring log, server check scheduling, EWS and scheduler, as well as error notifications. The author is confident that this application is functional, easy to use, and intuitive.

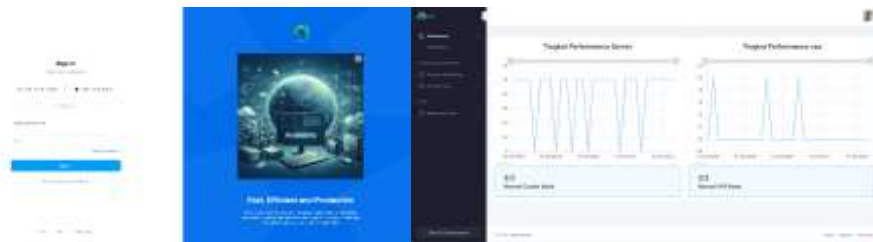


Figure 6. Login Page and dashboard

Image 6 is the login page and dashboard page. The login page serves as the entrance to the application, where users must enter valid credentials to access the features and information in the application. Meanwhile, the dashboard page now displays server status graphs, providing a visual overview of availability, performance, and system status quickly..



Figure 7. Proxmox Monitoring and Proxmox Login Pop-up

Image 7 is the Proxmox Monitoring Page and Proxmox login pop-up. The Proxmox Monitoring Page now serves as a control center for monitoring cluster nodes, where users can add monitoring for containers (CT) and virtual machines (VM) through the sidebar menu. Meanwhile, the Proxmox login pop-up now allows users to add new nodes..



Figure 8. Add CT Pop-Up and Add VM Pop-Up

Image 8 is the 'add ct' pop-up and the 'add vm' pop-up. The 'add ct' pop-up now displays a list of available containers in the cluster node that can be selected by the user. Meanwhile, the 'add vm' pop-up now displays a list of available virtual machines in the cluster node that can be selected by the user..



Figure 9. Set Scheduler And EWS and VM Monitoring Page

Image 9 is the EWS configuration pop-up and VM Monitoring Page. The EWS configuration pop-up now allows users to adjust the monitoring frequency and notification sensitivity level for each node. Meanwhile, the VM Monitoring Page now provides users with direct access to a dedicated page to monitor and manage VMs on each cluster node.



Figure 10. CT Monitoring Page and Node Edit Pop-up

Image 10 is the CT Monitoring Page and node edit pop-up. The CT Monitoring Page now provides users with direct access to a dedicated page to monitor and manage containers on each cluster node. Meanwhile, the node edit pop-up now allows users to log into the selected Proxmox to edit the node before they can access Edit VM/CT Visibility..



Figure 11. VM Edit Pop-up and CT Edit Pop-up

Image 11 is the Edit VM pop-up and Edit CT pop-up. Both of these pop-ups show the Edit VM/CT Visibility feature that allows users to manage the display visibility of virtual machines or containers..



Figure 12. EWS Edit Pop-up and Notification Detail Page

Image 12 is the pop-up to edit the EWS Interval and the notification detail page. This pop-up now allows users to edit the EWS interval according to their needs and preferences. Meanwhile, the notification detail page now allows users to see each indication generated from EWS processing and set the notification interval.

b. Testing Stage

This application was tested using the Black Box Testing method, which focuses on system input and output and aligns with the Use Case Diagram. The goal was to ensure each feature functions according to user needs. The advantages of this method include efficiency, balanced testing, and allowing testers and developers to work independently (Nur Cholifah et al., 2018). This testing process was carried out by a QA from company X.

Table 1. Blackbox testing

No	Test Case Descriptoin	Expected Result	Actual Result
1	Login	Ensured the login feature works well by testing various username and password combinations	The result was as expected.
2	Dashboard	Checked if the dashboard provides accurate graphs about the server status and latest notifications	The result was as expected.
3	Cluster and Node monitoring	Ensured users can monitor clusters and nodes by checking if the displayed information matches the actual status	The result was as expected.
4	Managing Cluster and Node Monitoring	Ensured users can configure clusters and set monitoring settings by trying various settings.	The result was as expected.
5	Viewing Monitoring Log	Ensured users can view the monitoring log by checking if all important events are recorded in the log	The result was as expected.
6	Setting Scheduler	Ensured users can set the server check schedule and EWS and Scheduler by trying to set the check schedule. The result was as expected.	The result was as expected.
7	Viewing Warnings from Notifications	Ensured users can view error notifications by checking if error notifications are correctly sent when an error occurs	The result was as expected.
8	Logout	Ensured users can log out of their accounts and return to the login stage.	The result was as expected.

4 CONCLUSION

This study has effectively. Tried the Proxmox Server Monitoring System. Each aspect, such, as Login, Dashboard and Monitoring has testing. Is operating smoothly. The system has been linked with the Proxmox VE server. Put through test scenarios. This study contributes to the literature on server management and virtualization in relation to Proxmox VE and server monitoring systems. It can serve as a point of reference for research in related fields. The developed server monitoring system offers organizations a tool to monitor their servers availability and performance in time. By receiving notifications of server failures organizations can respond promptly and efficiently to these issues. This proactive approach helps prevent disruptions before they affect services thereby reducing server downtime and enhancing productivity. Moreover this study aids in optimizing the utilization of server resources leading to cost savings on operations. However it is important to recognize that this research is confined to Proxmox servers at present and does not encompass Pluggable Authentication Module (PAM) users on Proxmox. Despite these constraints the author intends to refine this system in work by addressing these limitations and enhancing the monitoring system with integration for PAM users, on Proxmox servers. Enhancing the grasp, on server performance and security will improve the monitoring systems efficiency in supervising and enhancing the infrastructure of the Proxmox VE server.

REFERENCES

- Algarni, S. A., Ikbali, M. R., Alroobaea, R., Ghiduk, A. S., & Nadeem, F. (2018). Performance evaluation of Xen, KVM, and proxmox hypervisors. *International Journal of Open Source Software and Processes*, 9(2), 39–54. doi: 10.4018/IJOSSP.2018040103
- Ali, E., Susandri, & Rahmaddeni. (2015). Optimizing Server Resource by Using Virtualization Technology. *Procedia Computer Science*, 59, 320–325. doi: 10.1016/j.procs.2015.07.572
- Ali, S. M., Hoq, S. M. N., Bari, A. B. M. M., Kabir, G., & Paul, S. K. (2022). Evaluating factors contributing to the failure of information system in the banking industry. *PLoS ONE*, 17(3 March). doi: 10.1371/journal.pone.0265674
- Ally, S. (2018). Comparative Analysis Of Proxmox VE And Xenserver As Type 1 Open Source Based Hypervisors. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, 7. Retrieved from www.ijstr.org
- Aroral, H. K. (2021). Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis. *International Journal of Applied Business and Management Studies*, 6(1), 2021.
- Ciccozzi, F., Malavolta, I., & Selic, B. (2019). Execution of UML models: a systematic review of research and practice. *Software and Systems Modeling*, 18(3), 2313–2360. doi: 10.1007/s10270-018-0675-4
- Elvis Pawan, st, Rosiyati Thamrin, nd H., Patmawati Hasan, rd, Sariyati Bei, th H., & Paulisen Matu, th. (2021). Using Waterfall Method to Design Information System of SPMI STIMIK Sepuluh Nopember Jayapura. *International Journal of Computer and Information System (IJCIS) Peer Reviewed-International Journal*, 02, 2745–9659. Retrieved from <https://ijcis.net/index.php/ijcis/index>
- Hamza, Z. A., & Hammad, M. (2019). Generating UML Use Case Models from Software Requirements Using Natural Language Processing. *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, 1–6. doi: 10.1109/ICMSAO.2019.8880431
- Harfadzi, & Dadan, I. (2016). PERANCANGAN DAN IMPLEMENTASI VIRTUALISASI SERVER. *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic*, 4(2), 89–97.
- Hariyanto, B., & Ariyanto, Y. (2015). DESAIN DAN ANALISIS KINERJA VIRTUALISASI SERVER MENGGUNAKAN PROXMOX VIRTUAL ENVIRONMENT. *Jurnal Simantec*, 5(1).
- Kausar Bagwan, M. I., & Swati Ghule, P. D. (2019). A Modern Review on Laravel-PHP Framework. *IRE Journals*, 2(12).

- Koç, H., Erdoğan, A. M., Barjakly, Y., & Peker, S. (2021). UML Diagrams in Software Engineering Research: A Systematic Literature Review. *The 7th International Management Information Systems Conference*, 13. doi: 10.3390/proceedings2021074013
- Kute, S., & Thorat, A. S. (2014). A Review on Various Software Development Life Cycle (SDLC) Models. *International Journal of Research in Computer and Communication Technology*, 3. Retrieved from <https://www.researchgate.net/publication/312473242>
- Marin, E., Perino, D., & Di Pietro, R. (2022). Serverless computing: a security perspective. *Journal of Cloud Computing*, 11(1). doi: 10.1186/s13677-022-00347-w
- Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: a comparative analysis. *International Journal of System Assurance Engineering and Management*, 14(4), 1504–1522. doi: 10.1007/s13198-023-01958-5
- Nur Cholifah, W., Yulianingsih, & Melati Sagita, S. (2018). PENGUJIAN BLACK BOX TESTING PADA APLIKASI ACTION & STRATEGY BERBASIS ANDROID DENGAN TEKNOLOGI PHONEGAP. *Jurnal String*, 3(2). doi: 10.30998/string.v3i2.3048
- Persaud Nadini and Dagher, R. (2021). The Importance of Monitoring and Evaluation for Decision-Making. In *The Role of Monitoring and Evaluation in the UN 2030 SDGs Agenda* (pp. 89–117). Cham: Springer International Publishing. doi: 10.1007/978-3-030-70213-7_4
- Prasandy, T., & Whisnumurti. (2015). Virtualisasi Server Sederhana Menggunakan Proxmox. *JURNAL TRANSFORMATIKA*, 12(2), 2.
- Richey, R., Klein, & James. (2014). Design and Development Research. In M. D. and E. J. and B. M. J. Spector J. Michael and Merrill (Ed.), *Handbook of Research on Educational Communications and Technology* (pp. 141–150). New York, NY: Springer New York. doi: 10.1007/978-1-4614-3185-5_12
- Sasmito, G. W., Informatika, J. T., Bersama, H., Mataram, J., 09, N., & Lor, P. (2017). *Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal*. 2(1). Retrieved from <http://www.tegalkab.go.id>,
- Serrat, O. (2023). Information and Communication Technology in Organizations: Impacts and Implications. In *Digital Solutions: Reframing Leadership* (pp. 13–28). Singapore: Springer Nature Singapore. doi: 10.1007/978-981-19-7253-9_2
- Silvi Purnia, D., Rifai, A., & Rahmatullah, S. (2019). Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android. *SEMNASSTEK*, 16.
- Van, C. W. (2017). The Waterfall Model and the Agile Methodologies: A comparison by project characteristics. In *researchgate*. doi: 10.13140/RG.2.2.36825.72805
- Yadav, N., Rajpoot, D. S., & Dhakad, S. K. (2019). LARAVEL: A PHP Framework for E-Commerce Website. *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 503–508. doi: 10.1109/ICIIP47207.2019.8985771
- Yudha, C. F., Wing, W. W., & Eko, P. (2019). Analisis Teknologi Virtual Mesin Proxmox Dalam Rangka Persiapan Infrastruktur Server (Studi Kasus: Universitas Nahdlatul Ulama Yogyakarta). *Jurnal INFORMA Politeknik Indonusa Surakarta*, 5, 2442–7942.