ISSN 2797-7889

Design and Implementation of Zabbix Monitoring to Optimize MTTD and MTTR at Nusa Putra University

Wira Harsa Abiyasa Rifki Ghani[1], Somantri[2], Muhammad Ikhsan Thohir[3]
[1,2,3] Department of Engineering Computer and Design, Nusa Putra University, Sukabumi, Indonesia
Email : wira.harsa_ti22@nusaputra.ac.id

The advancement of information technology has made Internet network management increasingly complex, including in academic environments. At Nusa Putra University, the existing network monitoring system remains passive and does not yet provide an automatic notification feature, causing incident detection to rely heavily on manual reports from users. This condition can result in delays in Mean Time To Detect (MTTD) and Mean Time To Repair (MTTR), as well as difficulties in determining incident handling priorities. This study aims to design and implement an SNMP-based network monitoring system using Zabbix integrated with real-time WhatsApp notifications and the Simple Additive Weighting (SAW) method for incident priority classification. The system was developed using the Network Development Life Cycle (NDLC) methodology. Incident priority was determined based on three criteria: impact, urgency, and duration. The system was evaluated through functional testing and a comparative analysis of MTTD and MTTR before and after implementation. The results showed that the average MTTD decreased from 487 minutes to 24 seconds, while the average MTTR decreased from 13 minutes to 4 minutes 48 seconds. These results indicate that the proposed integration reduced incident detection and repair times by providing automated monitoring, priority classification, and real-time notification to network operators.
Keywords: Zabbix, SNMP, SAW, WhatsApp Notification, MTTD, MTTR.

This is an open access article under theCC BY-NClicense

Corresponding Author:
Wira Harsa Abiyasa Rifki Ghani
Perum Graha Pertiwi No.46
wira.harsa_ti22@nusaputra.ac.id

1. Introduction

The advancement of information technology has driven the widespread development of Internet networks. However, the increasing scale and complexity of network infrastructures also make Internet network management more challenging. As network infrastructures continue to develop, monitoring and managing the systems that support them require increasingly effective approaches [1]. The Internet connects millions of users around the world and enables the exchange of information in various forms, including text, images, video, and audio [2]. This capability has made the Internet an important part of everyday life. In the academic environment, the Internet also serves as an additional learning resource that complements learning materials and provides access to information related to lectures and academic activities [3]. Therefore, as an educational institution, Nusa Putra University utilizes Internet infrastructure to support the implementation of teaching and learning activities and to provide access to various academic information services[4].

To support the availability and reliability of its Internet infrastructure, Nusa Putra University's Internet monitoring system utilizes the Simple Network Management Protocol (SNMP) [5]. SNMP is a protocol that facilitates communication between network devices and monitoring systems. Its architecture consists of several components that interact with one another. One of the main components is the SNMP Manager, which functions as the central entity responsible for requesting and processing information obtained from the monitored infrastructure. Another component is the SNMP Agent, which communicates with the Manager and retrieves technical information from the network device being monitored [6]. In practical

network management, the ability to obtain monitoring data alone is not sufficient when network incidents require immediate attention. Monitoring systems also need to provide timely information regarding the occurrence of an incident so that network operators can respond appropriately. When incident information is obtained only through manual reports, the time required to recognize a problem may become longer. This condition directly affects the time required to detect and respond to network incidents.

Existing monitoring approaches generally focus on detecting network conditions and delivering alerts to network administrators, while the prioritization of incidents that require immediate attention is not sufficiently addressed. When multiple incidents occur, notification alone does not provide a systematic basis for determining which incident should be handled first. This limitation can affect the efficiency of incident response and subsequently influence measurable network management indicators, particularly Mean Time To Detect (MTTD) and Mean Time To Repair (MTTR). Therefore, a monitoring approach that combines automated detection with a measurable incident-priority mechanism is needed.

Mean Time To Detect (MTTD)[7] is an important metric for evaluating how quickly an incident can be identified after it occurs, while Mean Time To Repair (MTTR) [8]represents the time required to restore a service or system after an incident has been detected. Delays in both stages can affect the effectiveness of network operations, particularly when multiple incidents occur and operators need to determine which problem should be addressed first. The existing monitoring condition at Nusa Putra University presents two related challenges. First, the monitoring process does not yet provide an automatic notification mechanism that can immediately communicate detected incidents to network operators. Second, the absence of a structured incident priority mechanism can make it difficult to determine the order in which multiple incidents should be handled. These conditions may contribute to delays in incident detection and resolution and may result in an accumulation of unresolved problems.

To address these issues, this study proposes the design and implementation of a network monitoring system that integrates Zabbix-based SNMP monitoring with the Simple Additive Weighting (SAW) method and real-time WhatsApp notification. Zabbix is used to monitor network infrastructure and detect network incidents, while SAW is applied to determine incident priority based on impact, urgency, and duration. The resulting priority information is delivered to network operators through WhatsApp in real time. Unlike a monitoring approach that only detects and reports incidents, the proposed system combines incident detection, priority classification, and notification within a single monitoring workflow. The effectiveness of the proposed approach is evaluated by comparing Mean Time To Detect (MTTD) and Mean Time To Repair (MTTR) before and after implementation.

2. Literature Review and Problem Statement

Previous studies related to network monitoring have demonstrated the use of different monitoring platforms, data collection protocols, visualization tools, and notification mechanisms. However, their approaches differ in terms of the functions provided beyond incident detection. Zabbix-based studies have demonstrated the capability of SNMP monitoring for observing network infrastructure, while other studies have extended monitoring systems with visualization and automated notification features. The following studies are reviewed to identify the monitoring methods, notification mechanisms, and limitations that remain relevant to the development of the proposed system.

Table 1. Literature Review

No	Title	Description
1	*Design and Implementation of Computer Network Infrastructure Monitoring at the*	This study implemented Zabbix to monitor network infrastructure, including routers, switches, and firewalls,

ISSN 2797-7889

No	Title	Description
	Data Center and Informatics Facilities Using Zabbix Network Engineering, Kamaludin Rivaldi and Giri Purnama (2025)[9]	in real time. SNMP and MySQL were used, and Black Box Testing was applied to identify conditions such as device temperature, port-down status, and bandwidth traffic anomalies.
2	*Implementation of Network and Server Monitoring Systems Using Zabbix Integrated with Grafana and Telegram*, Muhammad Aliyul Husna and Perani Rosyani (2021)[10]	This study implemented Zabbix for network and server monitoring at PT Cyber Network Indonesia. The system was developed to overcome the limitations of Cacti, which did not provide notification features. Grafana was used for visualization and Telegram for automated notifications.
3	*Design of Network Monitoring System Based on LibreNMS Using Line Notify, Telegram, and Email Notification*, Nurwan Reza Fachrurrozi et al. (2023)[11]	This study designed a LibreNMS-based network monitoring system at Institut Teknologi Telkom Jakarta. The system centralized device monitoring and provided automated notifications through Line Notify, Telegram, and Email to reduce dependence on manual monitoring.

The studies presented in Table 1 demonstrate that existing network monitoring systems have successfully addressed different aspects of infrastructure monitoring and automated notification. Rivaldi and Purnama focused on real-time infrastructure monitoring using Zabbix and SNMP, with Black Box Testing used to verify monitored conditions such as device temperature, port-down status, and bandwidth anomalies. Husna and Rosyani extended Zabbix monitoring with Grafana for visualization and Telegram for automated notifications, addressing the limitation of monitoring systems that did not provide notification features. Fachrurrozi et al. implemented LibreNMS with multiple notification channels, including Line Notify, Telegram, and Email, to reduce dependence on manual monitoring.

Although these studies demonstrate the benefits of automated monitoring and notification, the reviewed approaches primarily focus on detecting incidents and delivering monitoring information to administrators. They do not describe a mechanism for calculating incident priority based on multiple criteria before the notification is delivered. This becomes important when several incidents occur at the same time, because notification alone does not establish which incident should receive attention first. Therefore, this study introduces the Simple Additive Weighting (SAW) method to classify incident priority using three criteria: impact, urgency, and duration. The resulting priority information is integrated with Zabbix monitoring and delivered through WhatsApp, providing both incident notification and structured priority information for network operators. The approach is subsequently evaluated using MTTD and MTTR to examine changes in incident detection and repair performance before and after implementation.

3. Method

This study employed the Network Development Life Cycle (NDLC)[12] as the system development methodology because its stages are aligned with the development and implementation of network infrastructure. The research was conducted at Nusa Putra University by identifying the existing network monitoring problems, collecting infrastructure and performance data, designing the monitoring architecture, implementing the proposed system, and evaluating its performance. The main system components consisted of Zabbix as the monitoring platform, SNMPv2 for collecting network and server metrics, Node.js as middleware for processing alerts, and WhatsApp as the notification medium.

The data collection process involved direct discussions with the university's IT personnel and observation of the existing network and server infrastructure. Real-time numerical performance data were collected through SNMPv2, including interface status, CPU and memory utilization, and bandwidth traffic. These data were used as inputs for the Node.js script and subsequently converted into assessment values for the Simple Additive Weighting (SAW) method[13]. Time data were also collected in minutes and seconds to evaluate system performance using Mean Time To Detect (MTTD) and Mean Time To Repair (MTTR). For performance evaluation, incident time data were collected in two phases. The pre-implementation phase recorded the time required for operators to detect and repair incidents using the existing monitoring process, while the post-implementation phase recorded the operator response after receiving automated and prioritized notifications from the proposed system.

The proposed system followed five main NDLC[14] stages requirements analysis, system design, implementation, testing, and maintenance. During the design stage, the monitoring architecture was defined to describe data acquisition, alert processing, and notification delivery. Zabbix periodically collected data from the MikroTik network infrastructure[15] and Proxmox VE server environment[16]. When a monitored parameter exceeded its defined condition, Zabbix generated an alert and forwarded its JSON data through a webhook to the Node.js middleware. The middleware processed the alert and applied SAW to determine the incident priority before delivering the resulting information to the network operator through WhatsApp. The SAW assessment used three criteria impact, urgency, and duration. Each criterion was represented on a scale from 0 to 5.[17] Impact represented the extent of service disruption, urgency represented the severity of the detected condition, and duration represented how long the incident had persisted. Each criterion is assigned a value from 0 to 5 according to the classification presented in Table 2 Problem Scale

Table 2. Problem Scale

Scale	Impact	Urgency	Duration
0	No Impact	Not Classified	< 1 Min
1	Latency / service performance degradation, Student VLAN	Informational	1 – 3 Min
2	Packet loss, Building A VLAN	Warning	3 – 5 Min
3	Internal server, Lecturer and Study Program VLAN	Average	5 – 10 Min
4	Staff VLAN (6th floor, LOU)	High Problem	10 – 15 Min
5	Core devices down, Internet Provider Down	Disaster	> 15 Min

The scale values used for the three criteria were defined based on the incident classification developed for the network infrastructure at Nusa Putra University. Impact represents the extent of service or infrastructure disruption, Urgency represents the severity level requiring immediate attention, and Duration represents the length of time an incident remains unresolved. The weights for the three criteria were determined based on interviews and discussions with the Network Engineer of Nusa Putra University as the Subject Matter Expert (SME). Impact was assigned a weight of 0.30, Urgency 0.50, and Duration 0.20, resulting in a total weight of 1.00. The weighting reflects the relative importance of each criterion in determining incident priority as shown in Table 3 Criterion Weight Table

Table 3. Criterion Weight Table

Criterion	Symbol	Weight
Impact	w1	0.30 (30%)
Urgency	w2	0.50 (50%)

Criterion	Symbol	Weight
Duration	w3	0.20 (20%)

The preference value for each incident is calculated using the following equation:

$$Vi = \Sigma(wj \times rij)$$

Where V_i is the preference value of incident i, w_j is the weight of criterion j, and r_{ij} is the rating of incident i for criterion j.

Based on the weights used in this study, the calculation can be expressed as:

$$Priorityscore = (0.3 \times Impact) + (0.5 \times Urgency) + (0.2 \times Duration)$$

The resulting preference value represents the priority score of the incident. A higher score indicates a higher incident priority and therefore a greater need for immediate attention. The calculated priority score is then included in the notification message delivered to the network operator through WhatsApp.

System evaluation consisted of three testing approaches. First, Black Box Testing was conducted to verify the functional operation of Zabbix, SNMP data acquisition, monitoring of MikroTik and Proxmox VE, and WhatsApp alert delivery. Second, SAW classification was tested by comparing the system's priority calculation with the defined assessment criteria and by simulating an incident on a WireGuard container. Third, a comparative evaluation of MTTD and MTTR was performed using two observation phases, one before implementation, when incidents were identified through the existing monitoring and manual reporting process, and one after implementation, when incidents were detected automatically and prioritized through Zabbix, SAW, and WhatsApp notifications. For the comparative evaluation, incident detection and repair times were recorded during the pre-implementation and post-implementation phases. The collected values were compared to determine changes in the average MTTD and MTTR and to evaluate the effectiveness of the proposed monitoring system in reducing incident detection and response time.

ISSN 2797-7889

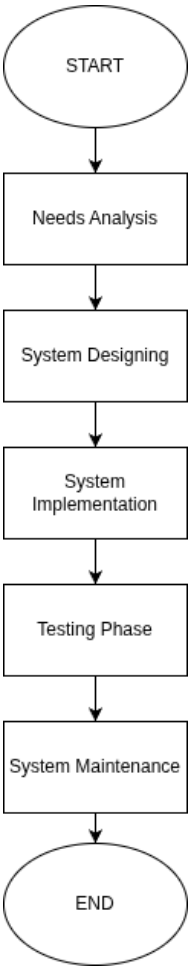


Fig. 1. NDLC Stages

4. Results And Discussion

Zabbix Monitoring Implementation

The monitoring system was implemented in a virtualized environment using Proxmox VE. Zabbix Server was installed on a Debian-based system and integrated with MariaDB as its database. The main components consisted of Zabbix Server, the web frontend, Zabbix SQL scripts, and Zabbix Agent. The MikroTik router was registered as a monitored host using the SNMPv2 protocol. The SNMP community configuration was adjusted on the router, while access was restricted to the IP address of the Zabbix Server. This configuration enabled Zabbix to collect network performance information, including interface conditions, traffic, and device-related metrics. Proxmox VE was also registered as a monitored environment to observe the performance of virtual machines and containers.

The implementation produced a monitoring dashboard containing information about the monitored network devices and server infrastructure. The communication between Zabbix and the MikroTik router was indicated by the active SNMP status, while the collected data were presented through graphical visualizations. Zabbix was also able to collect performance data from Proxmox VE, including CPU utilization of the monitored containers.

SAW-Based Alert Processing

When a monitored parameter exceeded the defined condition, Zabbix generated a trigger containing information about the detected incident. The alert was then forwarded through a webhook in JSON format

to the Node.js middleware. The middleware converted the alert information into values for three criteria: impact, urgency, and duration. The impact criterion represented the extent of the disruption to the network or service. The urgency criterion was based on the severity level generated by Zabbix, while duration represented the length of time that the incident had remained unresolved. Each criterion was assessed using the 0–5 scale and criterion weights defined in Section 3.

The preference value was calculated by multiplying the rating of each criterion by its corresponding weight and then summing the results. A higher preference value indicated a higher incident priority. The resulting priority information was included in the alert message delivered to the network operator through WhatsApp.

Functional Testing

Black Box Testing was conducted to verify the main functions of the monitoring system. The test covered data acquisition, device monitoring, alert generation, and notification delivery. The results are summarized in Table 4. Black Box Testing

Table 4. Black Box Testing

No	Tested Function	Result
1	Administrator login	Successful
2	User access restriction	Successful
3	SNMP polling from MikroTik	Successful
4	Proxmox VE data polling	Successful
5	Alert delivery to WhatsApp	Successful
6	Network link-down detection	Successful
7	Proxmox VE service-down detection	Successful

The results show that all tested functions operated according to the expected behavior. Zabbix successfully collected data from the MikroTik router and Proxmox VE environment. The system also detected network and service failures and forwarded the resulting alerts to the operator through WhatsApp. In the service-down scenario, the system detected the problem in less than one minute.

SAW Priority Classification Testing

The SAW classification was tested using a WireGuard container in the university's server environment. The container was intentionally shut down to simulate a service failure without affecting the main network infrastructure. This action generated an alert in Zabbix, which was then forwarded to the Node.js middleware for priority calculation.

During the initial detection, the incident received an impact value of 3, an urgency value of 3, and a duration value of 0. The resulting priority value was 2.40. The container was then kept offline for approximately 15 minutes. As the duration increased, the calculated priority value rose to 2.80. This result indicates that the duration criterion allowed the system to increase the priority of an unresolved incident over time. The alert information was delivered to the Zabbix Information WhatsApp group. The system sent periodic updates at approximately one-minute intervals and delivered a resolution notification after the container was restarted. Therefore, the integration between Zabbix, Node.js, SAW, and WhatsApp supported continuous incident reporting and priority updates.

MTTD and MTTR Comparison

The effectiveness of the proposed system was evaluated by comparing Mean Time To Detect (MTTD) and

Mean Time To Repair (MTTR) before and after implementation. The pre-implementation data represented the existing monitoring process, in which incidents were identified through manual reports or operator checks. The post-implementation data represented the Zabbix monitoring system integrated with SAW classification and WhatsApp notifications.

Table 5. MTTD Comparison Before and After Implementation

Incident	Before Implementation	After Implementation
Proxmox server down / MikroTik CPU overload	2,340 min	0 min
Student network or Indihome packet-loss issue	17 min	1 min
Indihome link or traffic issue	15 min	0 min
Vnet SOHO / WireGuard service issue	18 min	0 min
Building A Wi-Fi / backup container issue	45 min	1 min
Total	**2,435 min**	**2 min**
Average	**487 min**	**24 sec**

The total MTTD decreased from 2,435 minutes before implementation to 2 minutes after implementation. The average MTTD consequently decreased from 487 minutes to approximately 24 seconds. This reduction was mainly associated with the automatic detection and notification mechanism, which eliminated the need to wait for user reports or periodic manual checks. MTTR Comparisson is presented on Table 6. MTTD Comparison Before and After Implementation

Table 6. MTTR Comparison Before and After Implementation

Incident	Before Implementation	After Implementation
Incident 1	13 min	5 min
Incident 2	12 min	3 min
Incident 3	10 min	11 min
Incident 4	18 min	2 min
Incident 5	12 min	3 min
Total	65 min	24 min
Average	13 min	4 min 48 sec

The total MTTR decreased from 65 minutes to 24 minutes, while the average MTTR decreased from 13 minutes to 4 minutes and 48 seconds. Although one post-implementation incident required a longer repair time than its corresponding pre-implementation case, the overall average decreased. The result suggests that early detection and the availability of incident-priority information helped operators focus their troubleshooting activities more efficiently.

The comparison indicates that the proposed Zabbix-based monitoring system was particularly effective in reducing detection time. The integration of SNMP monitoring, SAW-based prioritization, Node.js alert processing, and WhatsApp notification provided operators with faster information about network and server incidents. The system also supplied additional context regarding incident impact, urgency, and duration, which supported the organization of incident-handling activities.

5. Conclusion

This study successfully designed and implemented a network monitoring system at Nusa Putra University using Zabbix integrated with SNMPv2, Node.js middleware, the Simple Additive Weighting (SAW) method, and WhatsApp notifications. The implemented system was able to monitor network devices and server infrastructure in real time, detect network and service disruptions automatically, and deliver incident information together with its priority level to network operators.

The functional testing showed that the main system functions operated according to the proposed design, including SNMP data collection, infrastructure monitoring, incident detection, WhatsApp notification delivery, and resolve-message notification. The SAW implementation also demonstrated that incident priority could be dynamically updated based on impact, urgency, and duration. The priority value increased as the incident duration increased, allowing operators to receive continuously updated information while an incident remained unresolved.

The comparison between the pre-implementation and post-implementation phases showed a reduction in both detection and repair times. The average MTTD decreased from 487 minutes to 24 seconds, while the average MTTR decreased from 13 minutes to 4 minutes 48 seconds. These results indicate that automated monitoring and notification reduced dependence on manual incident reporting and provided operators with more timely information for responding to network and server problems.

The proposed integration provides a more structured mechanism for network incident monitoring and response at Nusa Putra University. The combination of automated detection, dynamic priority classification, and real-time notification supports faster identification and handling of infrastructure problems.

6. References

[1] R. Bianco Huwae *et al.*, “IMPLEMENTASI SISTEM MONITORING JARINGAN MENGGUNAKAN ZABBIX BERBASIS SNMP PADA UPT. PUSAT TEKNOLOGI INFORMASI DAN KOMPUTER (PUSTIK) UNIVERSITAS MATARAM (Implementation of Network Monitoring System Using Zabbix Based on SNMP at Center for Information Technology and Computer Services of Mataram University).” [Online]. Available: http://begawe.unram.ac.id/index.php/JBTI/

[2] M. F. Mohd Fuzi, N. F. Mohammad Ashraf, and M. N. F. Jamaluddin, “Integrated Network Monitoring using Zabbix with Push Notification via Telegram,” *Journal of Computing Research and Innovation*, vol. 7, no. 1, pp. 147–155, Mar. 2022, doi: 10.24191/jcrinn.v7i1.282.

[3] K. N. Abidah, J. Ernajati, and M. Anasrulloh, “PENGARUH PENGGUNAAN INTERNET SEBAGAI SARANA PENUNJANG BELAJAR DAN KEMANDIRIAN BELAJAR TERHADAP PEMAHAMAN MATERI PELAJARAN EKONOMI SISWA KELAS XI IPS DI SMA NEGERI 1 TULUNGAGUNG TAHUN PELAJARAN 2022/2023,” *JURNAL ECONOMINA*, vol. 2, no. 7, pp. 1855–1864, Jul. 2023, doi: 10.55681/economina.v2i7.683.

[4] I. S. Yunika and I. N. Ichsan, “Quality of Service on Virtual Local Area Network (VLAN) in Campus Network,” *SISTEMASI*, vol. 14, no. 1, p. 126, Jan. 2025, doi: 10.32520/stmsi.v14i1.4725.

[5] B. Aini Muhammad Bakhit, M. Azahari Mohd Yusof, and A. Geogiana Buja, “COMPREHENSIVE REVIEW OF BANDWIDTH USAGE ANALYSIS WITH SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP) IN FSKM LABORATORY,” 2025.

[6] A. Pradana, I. R. Widiasari, R. Efendi, and T. Informatika, “Implementasi Sistem Monitoring Jaringan Menggunakan Zabbix Berbasis SNMP,” *AITI: Jurnal Teknologi Informasi*, vol. 19, no. Agustus, pp. 248–262, 2022.

[7] “What is Mean Time to Detect (MTTD)_ _ Lumifi Cybersecurity”.

[8] M. Husein Habibi, A. Jibril, and P. Kelautan dan Perikanan Sidoarjo, “Analisis Perhitungan Mean Time Between Failure (MTBF) Dan Mean Time To Repair (MTTR) Mesin Cold Storage,” *Jurnal Cendekia Ilmiah*, vol. 4, no. 4, 2025.

[9] K. Rivaldi and G. Purnama, “Perancangan dan Penerapan Monitoring Infrastruktur Perangkat Jaringan Komputer pada Pusat Data dan Sarana Informatika melalui Pengaplikasian Zabbix Network Engineering”, [Online]. Available: https://e-journal.naureendigition.com/index.php/mj

[10] M. A. Husna and P. Rosyani, “Implementasi Sistem Monitoring Jaringan dan Server Menggunakan Zabbix yang Terintegrasi dengan Grafana dan Telegram,” *JURIKOM (Jurnal Riset Komputer)*, vol. 8, no. 6, pp. 247–247, Dec. 2021, doi: 10.30865/jurikom.v8i6.3631.

[11] N. R. Fachrurrozi, A. A. Wirabudi, and S. A. Rozano, “Design of network monitoring system based on LibreNMS using Line Notify, Telegram, and Email notification,” *Sinergi (Indonesia)*, vol. 27, no. 1, pp. 111–122, 2023, doi: 10.22441/sinergi.2023.1.013.

[12] Miftahur Rahman, Moh. Dasuki, and Hardian Oktavianto, “Implementasi Manajemen Bandwidth Simple Queue Sebagai Optimalisasi Layanan Jaringan Internet Warga Menggunakan Metode NDLC,” *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 5, no. 1, pp. 27–35, Apr. 2024, doi: 10.37859/coscitech.v5i1.6899.

[13] M. H. F. Harits, M. I. Thohir, and A. Sujjada, “Implementasi logika fuzzy mamdani dan simple additive weighting (saw) pada sistem pakar berbasis web untuk deteksi dini gangguan neurologis,” *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 6, no. 3, pp. 575–584, Dec. 2025, doi: 10.37859/coscitech.v6i3.10130.

[14] Y. Kristiyanto, S. Sugiyono, C. A. Perdana, and S. Sarah, “PENGEMBANGAN KEAMANAN KOMPUTER DENGAN TEKNOLOGI THIN CLIENT DAN DOMAIN CONTROLLER DENGAN METODE NETWORK DEVELOPMENT LIFE CYCLE (NDLC),” *Jurnal Manajamen Informatika Jayakarta*, vol. 4, no. 3, pp. 258–258, Jul. 2024, doi: 10.52362/jmijayakarta.v4i3.1531.

[15] Rudi Hartono and Matsna Nurul Kholidah, “Sistem User Management Menggunakan Mikrotik Routeros untuk Mengontrol Akses Internet Siswa pada SMK Miftakhul Huda Way Areng,” *JUISIK*, vol. 4, no. 2, pp. 18–28, Jul. 2024, doi: 10.55606/juisik.v4i2.1828.

[16] S. Sheila, “EVALUASI TEKNOLOGI VIRTUALISASI MESIN PROXMOX UNTUK MEMPERSIAPKAN INFRASTRUKTUR SERVER”.

[17] S. Pendukung Keputusan Menggunakan *et al.*, “DECISION SUPPORT SYSTEM USING THE SAW METHOD FOR DETERMINING RESEARCH PRIORITIES AT UNIVERSITAS ISLAM KUANTAN SINGINGI,” vol. 10, no. 3, 2025.