Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi
https://ojs.stmik-banjarbaru.ac.id/index.php/jutisi/index
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0893

Implementation of Zabbix-Based Network Monitoring with Telegram and Web Reporting

DOI: http://dx.doi.org/10.35889/jutisi.v14i2.3040

Ahmad Yazid Isnandar[1*], Doddy Ridwandono[2], Nambi Sembilu[3]
Sistem Informasi, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia
*e-mail *Corresponding Author:* ahmad.yazid.upnjatim@gmail.com

Abstract

Effective network monitoring is essential in ensuring the stability and performance of campus IT infrastructure. This study aims to implement an open-source network monitoring system using Zabbix, integrated with Telegram alerts and a web-based visualization dashboard built with Laravel Filament. The development process follows the PPDIOO methodology, encompassing stages such as planning, installation, configuration, alert integration, and dashboard development. The system was implemented for the network infrastructure of two buildings at Universitas Pembangunan Nasional "Veteran" Jawa Timur: the Faculty of Computer Science Building (FIK2) and the Shared Lecture Building (GKB). The results show that the system effectively monitors the status of network devices, CPU and memory usage, DHCP leases, and network traffic. Additionally, it features real-time alerts via Telegram and web-based reports for visualizing network data. This solution enhances operational efficiency and offers a flexible and adaptive monitoring tool tailored to the needs of campus network management.

Keywords: *Laravel Filament; Network Monitoring; PPDIOO; Telegram Bot; Zabbix*

1. Introduction

The rapid advancement of information technology, particularly in the field of computer networking, has underscored the growing need for robust and reliable network monitoring systems to ensure optimal performance and operational stability [1]. In educational institutions such as an academic institution in Surabaya, network infrastructure plays a critical role in supporting both academic and administrative activities. However, in certain buildings like the Faculty of Computer Science 2 (Fasilkom Ilmu Komputer – FIK 2) and the Joint Lecture Building (Gedung Kuliah Bersama – GKB), the current monitoring process remains manual. This often leads to delayed responses to connectivity issues and disrupts essential services that rely on continuous internet access.

To overcome such challenges, the implementation of an automated and real-time network monitoring system is necessary. Zabbix, an open-source network monitoring solution, has gained popularity due to its support for both agent-based and agentless monitoring, automated notification systems, and API integration capabilities [2]. Compared to other monitoring tools, Zabbix offers a practical balance of advanced functionality and cost-efficiency [3]. Prometheus, while ideal for cloud-native applications, lacks direct support for traditional network devices. LibreNMS provides attractive visualizations but offers limited flexibility in customization [4]. PRTG is user-friendly but restricts usage in its free version by limiting the number of monitored devices [1]. Meanwhile, commercial-grade solutions such as SolarWinds and Datadog offer comprehensive features but are generally cost-prohibitive for institutions operating under tight budget constraints.

Prior research has effectively demonstrated the use of Telegram for real-time alerts in various monitoring and IoT applications—such as Netwatch-based network governance tools [5], formalin detection devices for food safety [6], and Telegram-based bots for image analytics [7]. These studies highlight the practicality and responsiveness enabled by Telegram integration, but there remains a usability gap for academic institutions: Zabbix's default interface

can overwhelm non-technical users like IT support staff or student interns. To address this limitation, this research proposes the development of a custom monitoring dashboard using Laravel and the Filament framework. This tailored dashboard presents essential network metrics—such as device uptime, interface status, CPU and memory usage, and bandwidth trends—in a simplified, user-friendly format. In addition, Telegram integration ensures administrators receive real-time notifications of network anomalies without needing to continuously monitor the dashboard [8].

This research adopts the PPDIOO methodology (Prepare, Plan, Design, Implement, Operate, Optimize) as a structured framework for developing a real-time network monitoring system based on Zabbix [9]. While previous studies mainly focus on basic monitoring and Telegram alerts, this work introduces a Laravel Filament-based custom dashboard integrated with multi-channel notifications (Telegram, website, and web push). The solution aims to improve incident response speed and provide an accessible visualization platform for both technical and non-technical users. It is designed as a scalable, low-cost monitoring approach suitable for an academic institution in Surabaya.

2. Literature Review

The study by M. Ishaq and Firmansyah focuses on the implementation of a network monitoring system using Zabbix on a Linux-based server [10]. One of the key contributions of this work is the integration of Telegram for real-time notifications, enabling immediate awareness of network changes. Their findings demonstrate that Zabbix significantly improves the effectiveness and responsiveness of network monitoring, aligning closely with this study's objective to enhance network visibility and incident response.

A similar approach is seen in the work by Lianda, who implemented Zabbix at SMA N 1 Kota Bengkulu [11]. Their system was equipped with Telegram notifications, providing real-time alerts whenever the network experienced downtime or disconnections. The combination of Zabbix and Telegram proved effective in ensuring administrators remained informed about network conditions at all times, which supports this study's direction in delivering timely insights through mobile alerts.

The research conducted by Rivan et al. presents a web-integrated network monitoring system deployed at PT Time Excelindo. Zabbix was utilized for data collection and monitoring, while Telegram was integrated for real-time notifications [12]. Although this study employed the NDLC methodology and focused on internal web server monitoring, it illustrates how Zabbix can be effectively paired with a web interface and messaging platforms—elements that this research also incorporates with additional enhancements using Laravel Filament.

Li designed a distributed network monitoring system for large-scale campus networks using Zabbix, TiDB, and Grafana [2]. Their study emphasized increasing the efficiency of anomaly detection while reducing the fault response time across multiple nodes. The use of Grafana for visualization and TiDB for distributed storage showcases how Zabbix can be extended for large, complex infrastructures, offering insights that could be adapted in scaling the system in this research.

Another relevant study by Sulasno et al. developed a mobile-integrated server resource monitoring system using Zabbix, Telegram, and Webhook [13]. They adopted the PPDIOO method and demonstrated how real-time server metrics can be accessed directly through smartphones, improving accessibility and decision-making. While their focus was on server monitoring, the combination of tools validates the suitability of these technologies for a broader range of network monitoring scenarios, including those addressed in this research.

Lastly, research by Fachrurrozi et al. implemented a monitoring solution based on LibreNMS, utilizing Line Notify, Telegram, and email for notifications [4]. Although the core monitoring tool differs, the use of Telegram as part of a multi-channel alert system demonstrates its continued relevance in modern network monitoring. The study supports the integration of real-time messaging platforms for administrator responsiveness, a principle that this research builds upon using Zabbix as the central engine.

Collectively, these studies demonstrate the efficacy of integrating Zabbix with Telegram for real-time network monitoring. However, they often focus on general network environments without tailoring the system for specific user needs. This research advances the field by applying the PPDIOO methodology and integrating a Laravel Filament-based custom dashboard

designed for academic infrastructures, improving usability for both technical and non-technical users.

3. Method

This study adopts the PPDIOO (Prepare, Plan, Design, Implement, Operate, Optimize) methodology developed by Cisco to ensure a structured and iterative approach to network system development. PPDIOO is selected due to its comprehensive framework that aligns well with the development of network monitoring systems, from early planning to post-deployment optimization [14]. The project also integrates modern web development tools such as Laravel Filament for dashboard creation, Zabbix for monitoring, and the Telegram API for real-time notifications [10]. The overall research flow is illustrated in Figure 1 [15](, comprising six core PPDIOO stages, followed by the formulation of conclusions and the final report documentation.

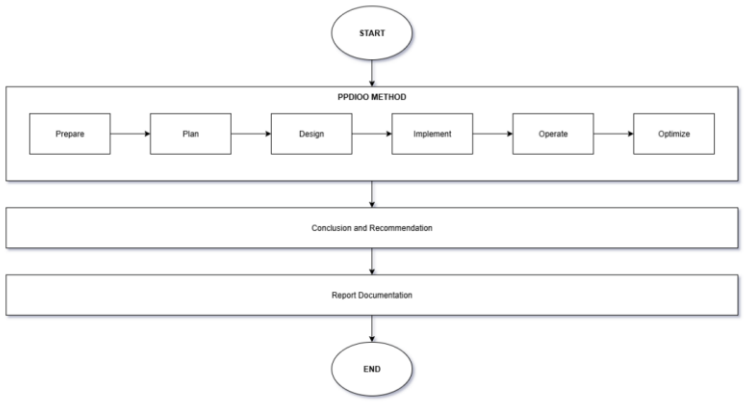

Fig. 1 Research Flow

1) PPDIOO Method

a. Prepare

In the preparation phase, a preliminary needs assessment was conducted through interviews and observational analysis of the current academic network infrastructure. This phase aimed to identify monitoring inefficiencies, especially the lack of real-time alerts and centralized visibility for administrators. Data collection included system logs, error records, and user feedback regarding existing manual or reactive monitoring practices.

b. Plan

The planning phase involved defining system goals, specifying performance expectations, and determining monitoring parameters. Functional and non-functional requirements were gathered through stakeholder consultations with IT staff, academic administrators, and technical operators. A work plan and timeline were developed to guide the development of the monitoring system, prioritizing user accessibility and real-time responsiveness.

c. Design

During the design phase, the network topology was mapped and monitoring targets were selected, including critical routers, switches, and access points within the campus infrastructure. The monitoring system architecture was planned to include the Zabbix server, a Laravel-based custom dashboard for visualization, and integration with Telegram for real-time alert notifications. Flowcharts and entity-relationship diagrams were also created to guide database and interface development.

d. Implement

This phase involved deploying the Zabbix server on a Linux-based system and integrating it with the Laravel Filament dashboard for frontend display. Custom triggers and alert rules were configured within Zabbix to detect anomalies such as device downtimes or bandwidth spikes. The Telegram Bot API was then connected to the alerting mechanism,

enabling immediate delivery of notifications to designated administrators. System components were incrementally developed and tested to ensure interoperability.

e. Operate

The system was tested in a real-world academic network environment. Monitoring operations included device status checks, log collection, and alert generation. Performance data such as average response time, alert delivery latency, and system uptime were logged. Usability testing was conducted with end-users (IT administrators and support staff) to gather feedback on dashboard clarity, alert usefulness, and accessibility. Evaluation criteria included system responsiveness, ease of use, and relevance of alerts to actual network events.

f. Optimize

Based on user feedback and performance metrics, several refinements were applied. These included adjusting alert thresholds, improving dashboard navigation, and streamlining notification formatting. Optimization focused on ensuring system scalability and minimizing false positives in alerts. Documentation and user guides were also prepared to support long-term system adoption and maintenance.

2) Conclusion and Report Documentation

After completing all stages of the PPDIOO methodology—from preparation to optimization—the process proceeds to the conclusion and reporting phase. In this stage, findings from the system implementation and testing are analyzed to draw meaningful conclusions regarding the effectiveness and performance of the network monitoring system. Recommendations for future improvements are also formulated based on the observed results and limitations. Subsequently, all development activities, configurations, test outcomes, and analysis results are compiled into a comprehensive final report. This documentation serves as a reference for system maintenance, further development, and institutional knowledge sharing.

4. Result And Duscussion

Table 1 Key Components in System Design

No.	Component	Function
1	SERMON (Zabbix Server)	Processes and stores monitoring data received from agents.
2	SERA (Zabbix Agent/SNMP)	Sends performance data from network devices and servers to the server.
3	PAM (Admin Device)	Used to access monitoring results via native or custom web dashboards.
4	Telegram Bot	Sends automatic alerts when issues or anomalies are detected.
5	Laravel Filament Dashboard	Provides simplified, user-friendly status and reporting visualization.

To illustrate the existing infrastructure being monitored, the physical network topology of the two main buildings—FIK2 and GKB—at an academic institution in Surabaya is presented. These topologies (see Figure 2 and Figure 3) include critical devices such as routers, switches, access points, and servers.

In FIK2, the topology is hierarchical, centered around a main distribution switch connecting to various devices across floors and rooms. This centralized structure enables better control and easier monitoring across the building. In contrast, GKB employs a more distributed layout. Its network includes several switches interconnected to serve classrooms, laboratories, and large lecture halls, highlighting the building's demand for high-capacity, reliable connectivity. These diagrams provide a clear overview of the infrastructure, enabling the monitoring system to be adapted to the actual environment for effective deployment.

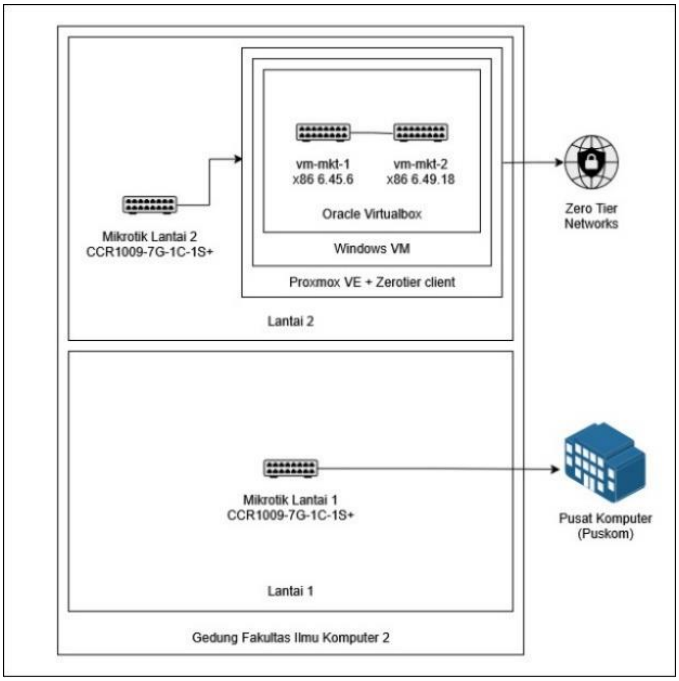

Fig. 2 Physical Network Topology FIK2

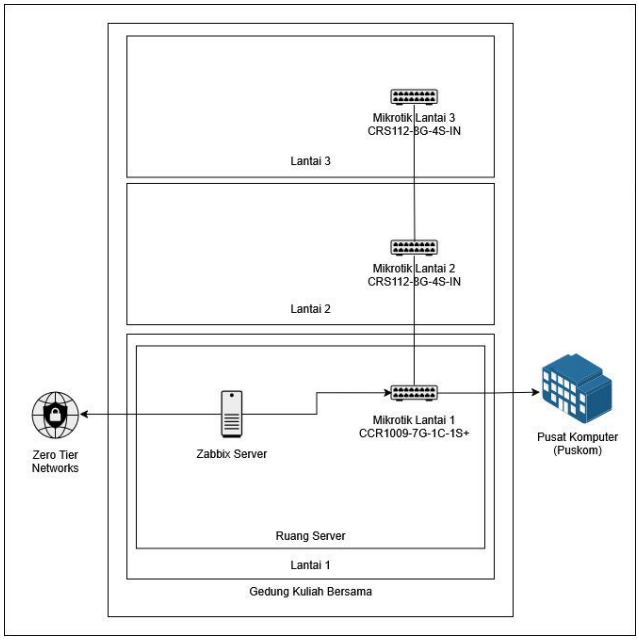

Fig. 3 Physical Network Topology GKB

The logical architecture of the monitoring system is shown in Figure 4, which reflects the centralized deployment of core components in the GKB server room. This includes:

- Zabbix Server for data collection and dashboard rendering (via Nginx),
- Laravel application for simplified reporting (via Apache),
- MariaDB for storing performance metrics,
- and communication channels using SNMP protocol with Mikrotik devices in both FIK and GKB buildings.

Laravel pulls monitoring data from Zabbix via API and presents it through a custom web interface, while Zabbix provides access through its own frontend. Both Laravel and Zabbix dashboards are accessible via Apache or Nginx as web gateways. This logical structure

supports real-time integration, reliable data retention, and centralized monitoring for technical staff.

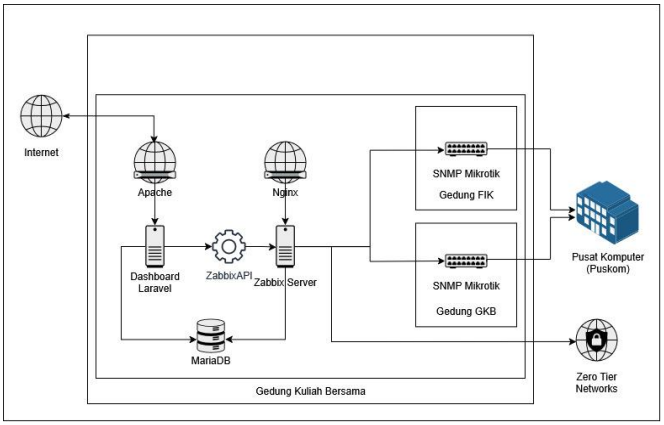

Fig. 4 Logical Topology of The Monitoring System

1) Implement

The implementation phase began with the installation and configuration of the Zabbix monitoring system on an Ubuntu 22.04 server, along with essential packages including NGINX, PHP, MariaDB, and Zabbix itself. Campus network devices—such as routers, switches, and internal service servers—were registered as hosts using SNMP and Zabbix agents. These hosts were configured with templates to collect key metrics including CPU utilization, memory usage, and network interface traffic.

To enable real-time alerting, Telegram was integrated by creating a bot, generating an access token, and configuring a webhook URL. This ensured that any critical triggers—such as service downtime or bandwidth spikes—resulted in immediate push notifications to the network administrator. In addition, a custom website-based notification mechanism was developed. This involved creating a new Zabbix media type that points to a Laravel endpoint, which processes incoming alerts and displays them on a centralized web dashboard.

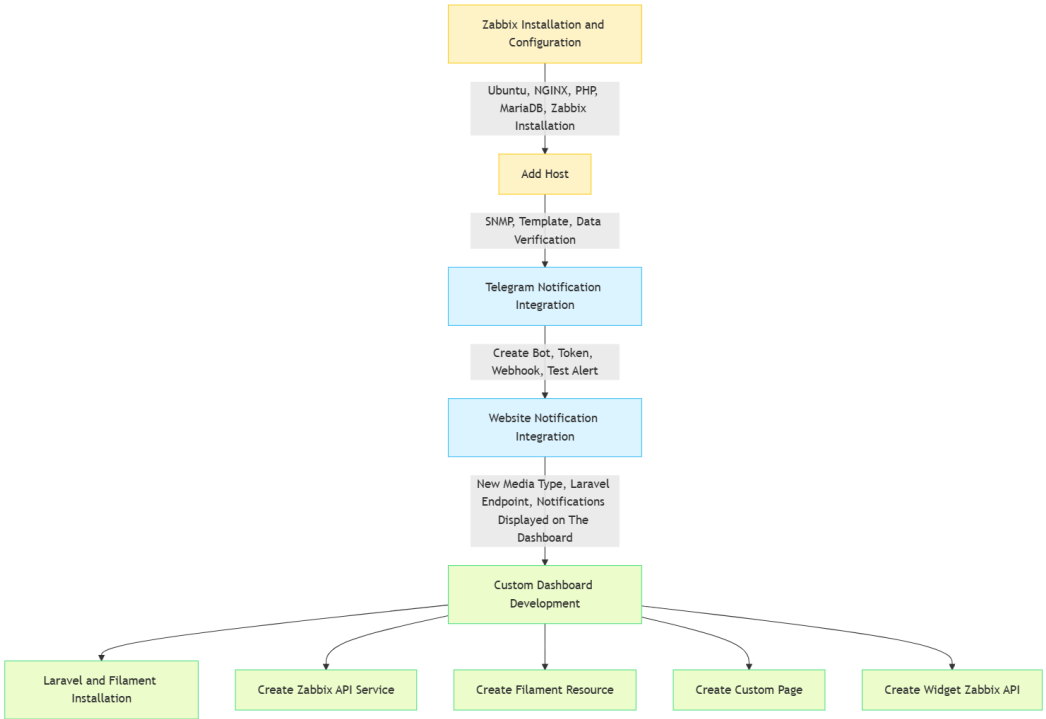

Fig. 5 Flow of Zabbix Monitoring System with Telegram and Laravel Dashboard Integration.

For better visualization and administrative control, a custom dashboard was developed using Laravel and the Filament library. The dashboard fetches data from Zabbix via its API and presents critical information such as host status, link availability, and historical trends. It also includes filtering options and PDF export functionality for reporting needs. The full implementation flow is illustrated in Figure 5.

The system was hosted on a virtual server with sufficient capacity to handle monitoring workloads, as detailed in Table 2.

Table 2 Virtual Server Specification

No.	Component	Specification
1	OS	Ubuntu Server 22.04 LTS
2	CPU	4 Core
3	RAM	8 GB
4	Storage	120 GB
5	Network	Bridge Adapter

Following the successful setup of Zabbix, its web-based frontend provided a comprehensive overview of all monitored hosts and metrics in real time, as shown in Figure 6.

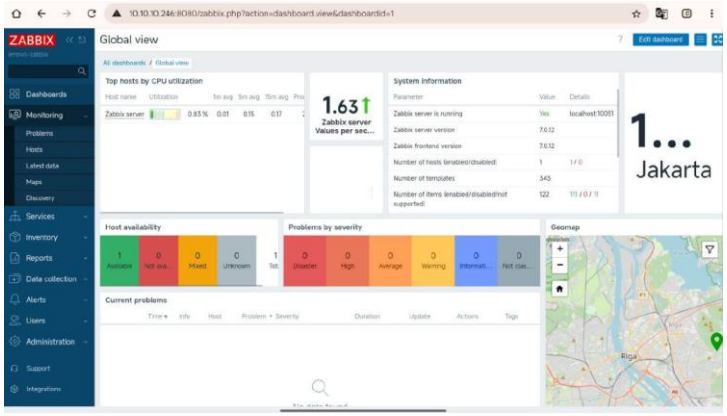

Fig. 6 Zabbix Frontend Dashboard

To enable notification delivery through Telegram, the media type had to be activated and configured. The configuration steps, including bot token setup and script parameters, are shown in Figure 7.

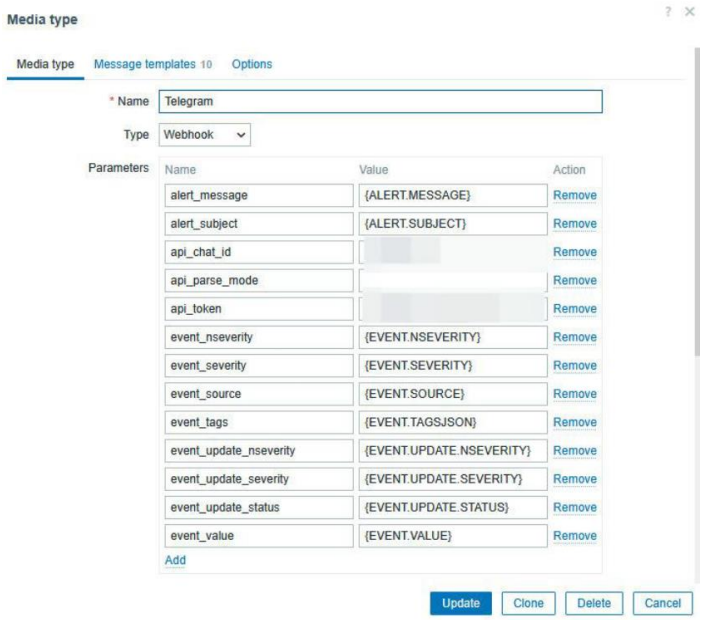

Fig. 7 Media Type Webhook Configuration of Telegram Integration

To support the web-based notification system, a webhook-based media type was created to route alerts from Zabbix to the Laravel application, as shown in Figure 8.

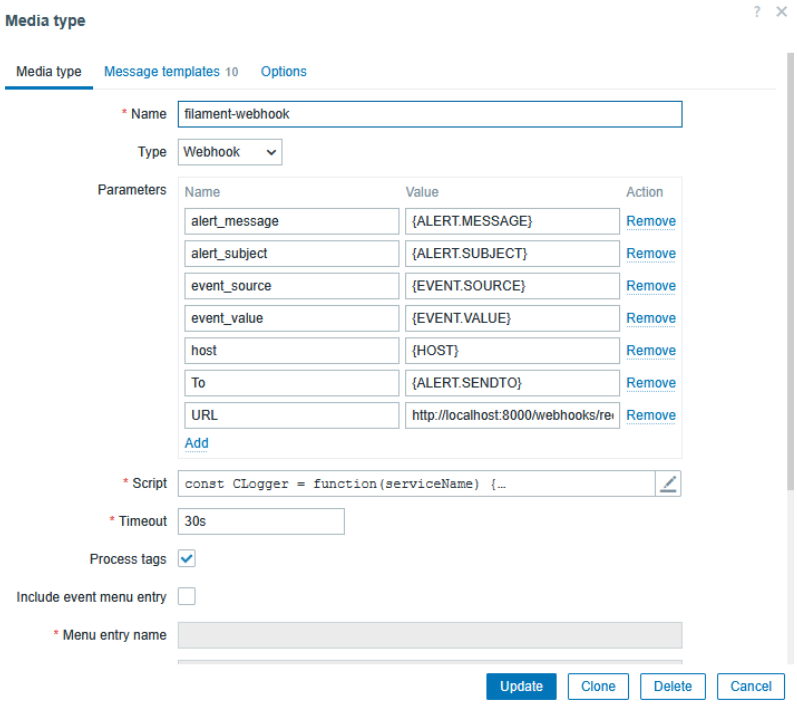

Fig. 8 Media Type Webhook Configuration for Laravel Integration

Finally, the data was visualized through a custom Laravel Filament dashboard, which allows administrators to view alerts, monitor host availability, and export reports, as illustrated in Figure 9.

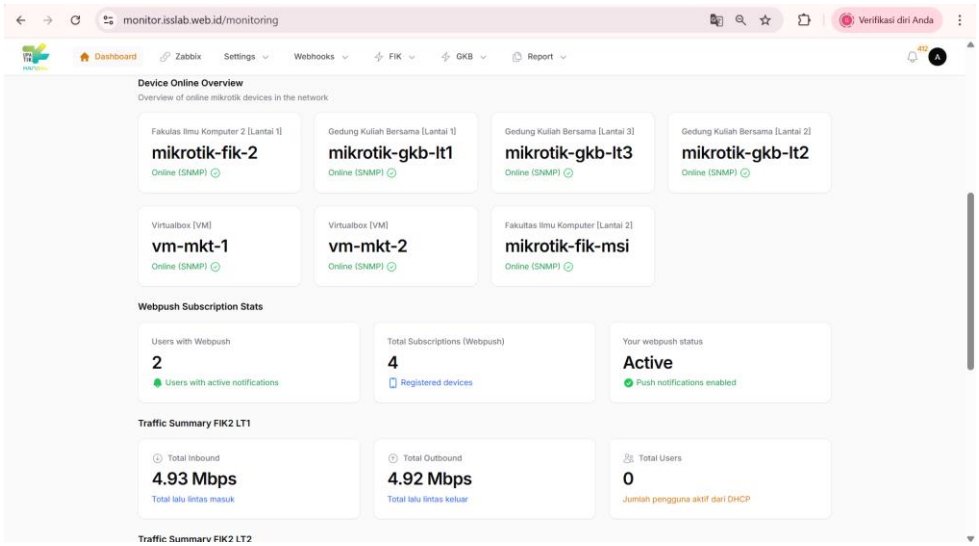

Fig. 9 Laravel Filament Monitoring Dashboard

In addition to the dashboard, the monitoring system also has a separate menu for each building, both FIK2 and GKB, which displays analytical chart widgets including CPU usage, memory usage, interface link up/down status, ICMP ping, interface combo traffic, and interface ether traffic. There is also a report menu for comparing metrics between buildings and viewing or printing network trend reports. Example of the analytical chart widgets displayed on each floor of each building can be seen in the Figure 10.

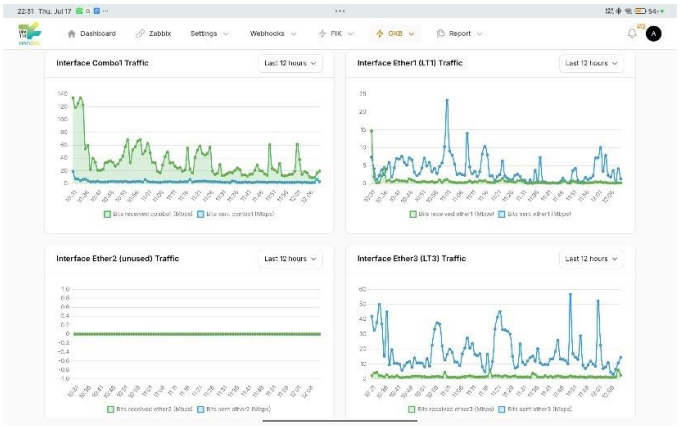

Fig. 10 Widget Chart for Each Network Monitoring Metrics

2) Operate

In this stage, the system was actively deployed to monitor the network infrastructure in real time. Zabbix was configured to detect the operational status of devices, such as routers and switches. When a device experienced a failure, Zabbix successfully identified the change and triggered an alert condition. These alerts were forwarded through a Telegram bot, which promptly delivered a structured notification message to a designated group chat, as illustrated in Figure 12. In addition to Telegram alerts, the system was also integrated with the Laravel-based dashboard to display notifications directly within the website interface for easier monitoring by non-technical users (Figure 13). Furthermore, a web push notification feature was implemented to immediately inform administrators via browser alerts even when they were not actively viewing the dashboard, as shown in Figure 14. The Laravel dashboard continued to provide an intuitive interface displaying the current status of all monitored devices, ensuring staff could easily identify issues at a glance. During this phase, special attention was paid to usability and clarity, ensuring that all critical functions performed as expected in real-time scenarios.

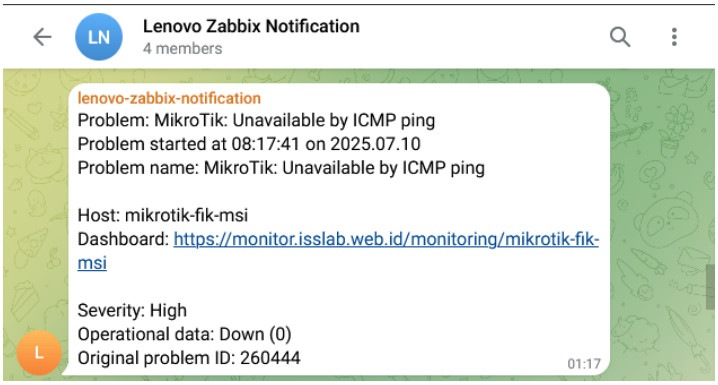

Fig. 11 Telegram Notification Example for Network Device Failure

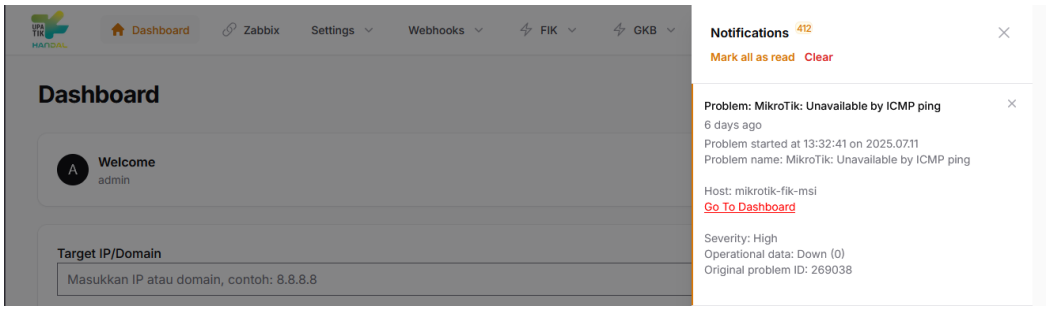

Fig. 12 Laravel Filament Notification Alert Display

Fig. 13 Laravel Filament Notification Webpush

3) Optimize

The was then validated through direct testing on campus. One real-life example was when a switch device in one of the buildings malfunctioned and became unresponsive. The system successfully detected the incident, sent a Telegram notification in less than two minutes, and changed the status of the device on the Laravel Filament dashboard in real time. The average response time of network staff to notifications was between 5–15 minutes, indicating improved efficiency in incident handling. This validation demonstrates that the system not only functions technically but also supports faster and more accurate decision-making in the context of network operations at the university.

Table 3. Effectiveness of the system

Validation test	Result	Description
Success rate of detection	100% (10/10 incidents detected)	No false negatives found.
False positives	1 incident out of 15 notifications	Caused by delayed SNMP response
Average notification delivery time	48 seconds (max. 1 minute 30 seconds)	Within the target of less than 2 minutes
Dashboard data consistency	100% synchronized with host status in Zabbix	Data is retrieved directly from the Zabbix API on a regular basis
Ease of report interpretation	Based on feedback from 3 technical administrators: “very helpful”	Concise display & PDF export available

Previous research by [13] explored Zabbix as a network monitoring tool in SMEs, but lacked integration with real-time messaging platforms. Another study by [12] proposed Telegram-based alerting using custom scripts, without a user-friendly dashboard. This study integrates both mechanisms and introduces a Laravel Filament-based web interface, making real-time monitoring more accessible for educational institutions. Thus, the proposed system bridges the gap between raw monitoring and practical usability, contributing a more integrated and scalable model to the domain.

5. Conclusion

This paper has presented the design and implementation of a network monitoring system that integrates Zabbix with Telegram notifications and a Laravel-based visual dashboard. The system was developed to address the challenges faced in traditional network monitoring setups, particularly in terms of real-time responsiveness and user accesibility.The successful deployment of the system on a virtualized environment confirmed that open-source tools can provide reliable, scalable, and customizable solutions for infrastructure monitoring without incurring high costs. The Telegram integration effectively enabled instant notifications, enhancing the administrator's ability to respond to incidents promptly. Meanwhile, the Laravel dashboard provided a user-friendly alternative interface for data visualization and reporting, supporting both technical and non-technical users in managing monitoring data.

While the system has shown promising results, it is not without limitations. Internet dependency for Telegram alerts and the need for further dashboard optimization were identified as areas for improvement. Nonetheless, the modular architecture allows future extensions, such as integration with additional platforms, predictive analytics, and personalized alert routing. The

implementation supports the conclusion that integrating Zabbix with external messaging platforms and frontend interfaces can significantly enhance the monitoring experience. It also opens opportunities for further innovation in the field of network infrastructure monitoring, particularly for educational institutions or small-to-medium enterprises that require efficient yet affordable solutions.

Daftar Referensi

[1] D. Arvianto, M. Kamisutara, and W. A. Kristiana, "Development Of Paessler Router Traffic Grapher Network Monitoring Application," *Int. J. Electr. Eng. Inf. Technol.*, vol. 07, no. 02, pp. 80–87, 2024.

[2] K. Li, "Research on Zabbix Monitoring System for Large - scale Smart Campus Network from a Distributed Perspective," pp. 631–648, 2024.

[3] S. Sulasno and R. Saleh, "Desain dan Implementasi Sistem Monitoring Sumber Daya Server Menggunakan Zabbix 4.0," *JUITA J. Inform.*, vol. 8, no. 2, pp. 187-196, 2020, doi: 10.30595/juita.v8i2.6886.

[4] 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.

[5] A. Wijaya, A. Kemala Jaya, D. Ayu Ningsih, and D. Lyanda, "Development Of Netwatch Host Using Telegram As A Strengthening Model Of Institutional Performance Quality Governance," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 5, no. 1, pp. 347–358, May 2023, doi: 10.47709/cnahpc.v5i1.2309.

[6] R. Prayogo, I. Anshory, I. Sulistiyowati, and S. Syahrorini, "Formalin detection device in Tofu food with telegram monitoring," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 5, no. 2, pp. 563–570, Jul. 2023, doi: 10.47709/cnahpc.v5i2.2393.

[7] A. Fajaryanto, F. Masykur, and M. Rizqi Rosyadi, "Implementation of Bot Telegram as Broadcasting Media Classification Results of Convolutional Neural Network (CNN) Images of Rice Plant Leaves," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 5, no. 1, pp. 1–9, Jan. 2023, doi: 10.47709/cnahpc.v5i1.1976.

[8] J. Purwanto, "Implementation of telegram bots for user management on radius servers with captive portal," *IJEEIT Int. J. Electr. Eng. Inf. Technol.*, vol. 5, no. 1, pp. 35–40, Jun. 2022, doi: 10.29138/ijeeit.v5i1.1864.

[9] M. D. S. Jayakumari, "Computer Networks," in *Artificial Neural Networks, Deep Learning and Computer Vision*, San International Scientific Publications, p. 180, 2024. doi: 10.59646/cn/283.

[10] M. Y. Ishaq and F. Firmansyah, "Implementasi Sistem Monitoring Menggunakan Zabbix Dan Notifikasi Realtime Telegram," *J. Insa. J. Inf. Syst. Manag. Innov.*, vol. 3, no. 2, pp. 72–77, 2023, doi: 10.31294/jinsan.v3i2.2432.

[11] D. Lianda, "Penerapan Zabbix Dengan Notifikasi Telegram Untuk Melakukan Monitoring Jaringan Penerapan Zabbix Dengan Notifikasi Telegram Untuk MelakukanMonitoring Jaringan," *Jalan Meranti Raya No.32 Sawah Lebar Telp*, vol. 20, no. 1, p. 341139, 2024.

[12] M. Rivan, A. Arsandi, and A. Syaripudin, "Perancangan Sistem Monitoring Jaringan berbasis Web Server Terintegrasi Zabbix dan Notifikasi Telegram Pada PT Time Excelindo," vol. 3, no. 6, pp. 1553–1561, 2024.

[13] S. Sulasno, R. Saleh, and I. Savitri, "Developing Integrated Smartphones Notification of Server Resource Monitoring System Using Zabbix, Webhook, and Telegram," *JUITA J. Inform.*, vol. 9, no. 2, pp. 191-202, 2021, doi: 10.30595/juita.v9i2.10411.

[14] A. Purwanto and B. Soewito, "Optimization Problem Of Computer Network Using PPDIOO," *ICIC Int.*, vol. 15, no. 7, pp. 769–777, 2021.

[15] A. S. Elrashdi, S. K. Alferjani, R. R. Omar, and F. M. Hasan, "The efficiency of using PPDIOO Methodology to Design Graduation Projects for Network Department Students," in *2024 IEEE 7th International Conference on Advanced Technologies, Signal and Image Processing (ATSIP)*, IEEE, Jul. 2024, pp. 438–442. doi: 10.1109/ATSIP62566.2024.10638951.