JISTecS: Journal of Information Systems, Technology and Security
ISSN 3125-1099 (online)
Vol. 1 Iss. 2 Pg. 148-164
https://ejournal.scmdigi.id/index.php/jistecs

Real-time network monitoring and automated whatsapp alerting for mikrotik infrastructure using PRTG

Putra Buana, Serly Oktarina*, Hadi Syaputra
Universitas Sumatera Selatan, Palembang, Sumatera Selatan, Indonesia
*Correspondence author: serlyoktarina@uss.ac.id
DOI: https://doi.org/10.65881/jistecs.v1i2.193

ARTICLE INFO

History:
Received: 09-01-2026
Revised: 09-06-2026
Accepted: 09-07-2026
Published: 09-15-2026

Keywords:
prtg network monitor;
mikrotik;
network monitoring;
whatsapp gateway api;
real-time alerting.

ABSTRACT

Purpose: to implement a real-time network monitoring and automated whatsapp alerting system for mikrotik infrastructure using PRTG network monitor, and to evaluate its fault detection capability and response time.
Method: this study employed a research and development (R&D) method involving observation, interviews, system implementation, and functional testing. The system was implemented in a testing network using PRTG network monitor to monitor mikrotik infrastructure and integrated with a whatsapp gateway API for automated alerts. System performance was evaluated through fault simulation and response-time measurement.
Findings: the system successfully monitored key mikrotik parameters, detected simulated network faults, and delivered automated whatsapp notifications. The PRTG detection time was 5–10 seconds, whatsapp notification delivery took 3–5 seconds, and the total system response time was 8–15 seconds.
Implications: the proposed system can improve the efficiency and responsiveness of network monitoring by providing centralized monitoring and automated real-time notifications, thereby supporting faster identification and response to network faults.
Originality: lies in the integration of PRTG-based mikrotik infrastructure monitoring with a whatsapp gateway API to provide automated real-time fault notifications and evaluate end-to-end detection and notification response time.

Introduction

The rapid development of information and communication technology has made computer networks a fundamental infrastructure for supporting business operations, particularly in internet service provider (ISP) environments (Aara et al., 2019; Mahmood & Yuksel, 2025). Network availability, reliability, and performance are critical

factors because disruptions can directly affect service quality, increase downtime, and ultimately reduce customer satisfaction (Khawar et al., 2024; Nurcahyanto et al., 2025). In an ISP environment, network conditions may change dynamically due to device failures, link interruptions, excessive traffic, packet loss, high latency, or other technical problems (Shahid et al., 2024). Therefore, continuous monitoring of network infrastructure is required to provide timely information regarding device availability and network performance (Aldea et al., 2023; Hegde & Varughese, 2022). Real-time monitoring is particularly important because early detection of abnormalities enables network administrators and technicians to respond to incidents before they develop into more serious service disruptions (Averineni, 2025; Kampourakis et al., 2025).

In practice, however, network monitoring is not always implemented in a proactive and integrated manner. Based on observations and interviews conducted at PT Nextmedia Teknologi Indonesia, Sukomoro Branch, network monitoring activities prior to this research were still performed predominantly through manual checking. Administrators were required to check mikrotik devices periodically or became aware of network problems only after receiving complaints from customers. This condition creates a reactive monitoring pattern in which network problems may remain undetected between manual inspection intervals. Such a mechanism is less efficient, depends heavily on administrator availability, and can increase the time required to identify and respond to network incidents. Consequently, the absence of centralized and automated monitoring may contribute to delayed fault detection, prolonged troubleshooting processes, and potentially longer service downtime.

The use of network monitoring systems provides an opportunity to overcome these limitations. PRTG network monitor is a network monitoring platform that can be used to centrally monitor network devices, services, and infrastructure through sensors and dashboards (Fathima & Devi, 2024). Previous research has demonstrated the applicability of PRTG for monitoring network and server conditions, indicating that centralized monitoring can improve the visibility of infrastructure status and facilitate the identification of network problems (Daud et al., 2024). Similarly, Arvianto et al. (2024) discussed the development of a network monitoring application based on Paessler Router Traffic Grapher, demonstrating the continued relevance of Paessler-based monitoring approaches for observing network infrastructure. In addition, Daud et al. (2024) reported the use of PRTG network monitor for enterprise infrastructure monitoring, further supporting the role of centralized monitoring platforms in network management.

Mikrotik routerOS provides additional capabilities that can support automated network monitoring and incident detection (Christanto et al., 2026). Its support for protocols and mechanisms such as SNMP, API, scripting, scheduler, and Netwatch enables administrators to obtain device information, monitor connectivity, and execute automated actions when specific network conditions occur (Bojović & Bojović, 2025). Ardiansyah et al. (2026) examined the utilization of mikrotik RouterOS in network monitoring, while their subsequent work discussed network monitoring using mikrotik RouterOS through an API-based approach. These studies indicate that mikrotik infrastructure can be integrated with monitoring mechanisms to provide more systematic information regarding network conditions. From a broader perspective, real-time monitoring has also been recognized as an important component in modern network environments because continuous observation can provide timely information about abnormal conditions and support faster operational responses (Kebande et al., 2021). Furthermore, intelligent and real-time alert mechanisms have been investigated

as a means of improving the responsiveness of network monitoring systems (Zhang et al., 2025).

Although centralized monitoring can provide information through dashboards, monitoring alone does not necessarily guarantee that technicians will immediately become aware of a detected incident. An effective monitoring system therefore needs to be complemented by an automated notification mechanism that can deliver alerts directly to operational personnel. Previous research has investigated automated notification mechanisms for computer network monitoring, demonstrating the potential of automated alerts to improve the delivery of network incident information (Safinah, 2026; Tanimu et al., 2026). WhatsApp is particularly relevant as a notification medium because it is widely used for real-time communication and can provide direct access to operational personnel (Lee et al., 2024). Kavitha et al. (2024) examined whatsapp as a real-time notification tool, while Febriani et al. (2025) investigated the implementation of a whatsapp gateway API for network monitoring systems. These studies provide evidence that whatsapp-based notifications can be utilized to distribute monitoring information automatically rather than relying exclusively on a monitoring dashboard.

Nevertheless, the existing literature generally addresses network monitoring, mikrotik-based monitoring, PRTG implementation, or automated messaging as separate components. Several studies focus on the implementation of PRTG as a monitoring platform (Daud et al., 2024; Fuzi et al., 2023), while other studies emphasize mikrotik routerOS and API-based monitoring (Ardiansyah et al., 2026). Other research investigates automated network notification or whatsapp gateway implementation (Febriani et al., 2025; Safinah, 2026). In addition, research on network downtime highlights that delays in detecting network disturbances can negatively affect service availability and operational performance, particularly in ISP environments (Murphy et al., 2024; Troia et al., 2022). However, there remains a practical and research gap in the integration of centralized PRTG-based monitoring of mikrotik infrastructure with automated, real-time whatsapp alerting, particularly in the operational context of an ISP branch. The gap is not merely the absence of another monitoring platform, but the need to establish and evaluate an end-to-end mechanism that connects network condition detection, automated event triggering, message delivery, and technician response within a single monitoring workflow.

Based on this gap, the novelty of this research lies in the implementation and evaluation of an integrated monitoring and alerting architecture in which PRTG network monitor functions as the centralized monitoring platform for mikrotik infrastructure. At the same time, a whatsapp gateway API serves as the automated notification channel for detected network incidents. The proposed approach is designed to transform the existing monitoring process from a predominantly manual and reactive mechanism into a more centralized, automated, and proactive workflow. In addition to implementing the integration, this research evaluates the system based on its ability to detect selected network disturbances, deliver automated notifications, and measure response time between the occurrence or detection of a network event and the delivery of the corresponding whatsapp alert. Thus, the research extends previous studies by not only implementing monitoring and notification technologies individually, but also examining their operational integration and performance as an end-to-end network incident alerting mechanism.

Accordingly, the main objective of this research is to design and implement a real-time network monitoring and automated whatsapp alerting system for mikrotik infrastructure using PRTG network monitor at PT Nextmedia Teknologi Indonesia,

Sukomoro Branch. More specifically, the research aims to: (1) develop a centralized monitoring mechanism for mikrotik network infrastructure using PRTG; (2) integrate PRTG monitoring events with a whatsapp gateway API to automatically distribute network alerts; (3) evaluate the capability of the proposed system to detect selected network conditions, including host status, latency, packet loss, throughput, and uptime; and (4) measure the response time and assess the effectiveness of the proposed mechanism in supporting faster network monitoring and incident identification.

This research is important because the effectiveness of network management is not determined solely by the availability of monitoring information, but also by how quickly and reliably that information reaches personnel responsible for network operations. For an ISP, even a relatively short delay in identifying and communicating a network problem can affect troubleshooting time and service continuity. By integrating monitoring and automated notification, network administrators can receive incident information without continuously observing the monitoring dashboard. The resulting mechanism is therefore expected to reduce dependence on manual checking, improve the timeliness of incident awareness, and support a more proactive approach to network maintenance. The research also provides an empirical basis for determining whether the proposed integration can produce measurable improvements in the speed and effectiveness of network incident notification.

The expected contribution of this research is both practical and academic. From a practical perspective, the proposed system is expected to provide PT Nextmedia Teknologi Indonesia with a more systematic mechanism for monitoring mikrotik infrastructure and communicating network incidents to technicians through a commonly used communication channel. The system is also expected to support faster identification of network problems and improve the efficiency of operational monitoring. From an academic perspective, this research contributes an implementation and evaluation model for integrating PRTG-based network monitoring with automated whatsapp alerting in an ISP environment. The results may serve as a reference for further research concerning real-time network monitoring, automated incident notification, mikrotik infrastructure management, and the integration of network monitoring platforms with messaging-based alert systems.

Method

This research employed a research and development (R&D) approach because the study was intended not only to identify and describe existing network monitoring problems but also to design, implement, and evaluate a monitoring system that addresses the operational requirements of the research object. The R&D approach was considered appropriate because the output of the research is a functional system that integrates network monitoring and automated notification mechanisms. The development process involved requirement analysis, system architecture design, implementation, integration, and functional evaluation. The research was conducted in the testing network environment of PT Nextmedia Teknologi Indonesia, Sukomoro Branch. The testing environment was intentionally separated from the primary production network to minimize the risk of disrupting operational services and customer connectivity. This separation also provided a controlled environment for conducting network fault simulations and evaluating the response of the proposed monitoring and notification system. Consequently, network disturbances could be generated deliberately without directly affecting the service level agreement (SLA) and services delivered to customers.

Data were collected using observation, interviews, and documentation. Observation was conducted to identify the existing network topology, types of network devices, monitoring procedures, communication paths, and common network disturbances encountered during operational activities. The observation process also examined how administrators and the Network Operation Center (NOC) currently identify and respond to network problems. Interviews were conducted with NOC personnel and technical staff who are directly involved in network monitoring and troubleshooting. The interviews focused on identifying limitations in the existing manual monitoring process, determining the network parameters that need to be monitored, identifying the types of incidents that require immediate notification, and determining the information that should be included in automated alerts. The results of the interviews were subsequently used as requirements for designing the proposed monitoring and notification system.

Documentation was used to collect supporting evidence throughout the implementation and testing processes. The documentation included network topology information, device configurations, PRTG dashboard displays, sensor configurations, testing results, system response measurements, and examples of whatsapp notifications generated by the system. These three data collection techniques were used complementarily to ensure that the developed system was based on actual operational requirements and that the implementation results could be properly documented. The research procedure consisted of several sequential stages: (1) requirements analysis, (2) system architecture design, (3) PRTG server preparation and implementation, (4) mikrotik device integration, (5) sensor configuration, (6) whatsapp gateway API integration, and (7) system testing and evaluation. The requirements analysis stage identified the monitoring needs and notification requirements based on observations and interviews. The architecture design stage then defined the relationship between the mikrotik devices, PRTG monitoring server, network infrastructure, and whatsapp gateway API.

The implementation stage began with the preparation of the monitoring server and installation of PRTG network monitor. The mikrotik devices were subsequently registered as monitoring objects and configured with the required monitoring mechanisms. Sensors were configured to observe important network conditions, including host status, ping, packet loss, latency, interface status, and uptime. These parameters were selected because they provide information about device availability, connectivity quality, interface conditions, and service continuity. After the monitoring mechanism had been established, PRTG was integrated with the whatsapp gateway API. The integration was designed so that a relevant monitoring event generated by PRTG could trigger an automated notification containing essential incident information, such as the affected device, detected condition, status, and event time. This mechanism was intended to reduce dependence on continuous dashboard observation and enable NOC personnel or administrators to receive incident information more promptly.

The overall architecture of the proposed monitoring system is presented in Figure 1. The architecture consists of the mikrotik RouterOS device as the primary network object being monitored, the PRTG network monitor server as the centralized monitoring platform, and the whatsapp gateway API as the automated notification channel. The supporting network infrastructure consists of a switch and client computers used for connectivity and testing purposes.

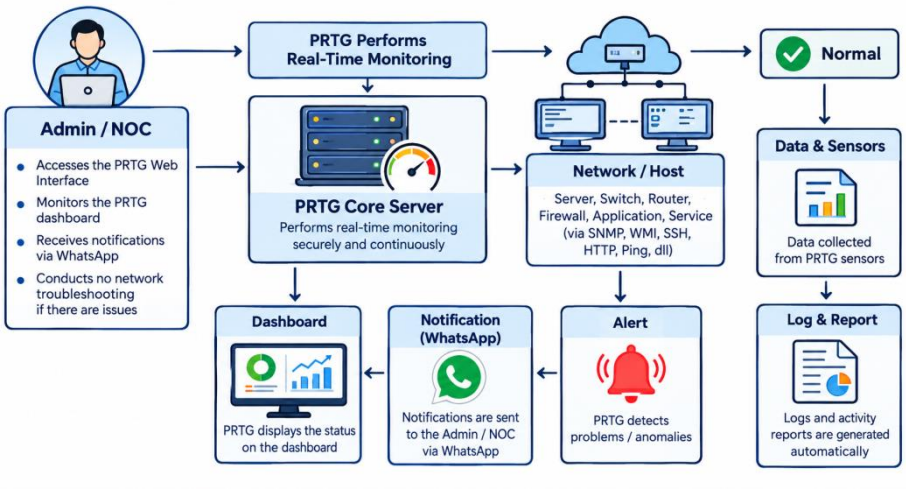

Figure 1 PRTG-based network monitoring system architecture

As illustrated in Figure 1, the mikrotik device provides network and device status information to the PRTG monitoring server through the configured monitoring mechanism. PRTG processes the collected information through its sensors and presents the results through a centralized dashboard. When a monitored parameter reaches a predefined abnormal condition, PRTG generates an alert event. The alert is then forwarded to the whatsapp gateway API, which processes the request and delivers an automated notification to the designated administrator or NOC personnel. This architecture establishes an integrated flow from network condition detection to automated incident notification. The operational sequence of the monitoring and notification process is presented in Figure 2. The workflow begins when the mikrotik device operates under normal network conditions. PRTG continuously monitors the configured sensors and evaluates the collected values against the defined monitoring conditions. When no abnormal condition is detected, the monitoring process continues normally. When a fault or abnormal condition is detected, PRTG generates an alert and initiates the notification process through the configured integration mechanism.

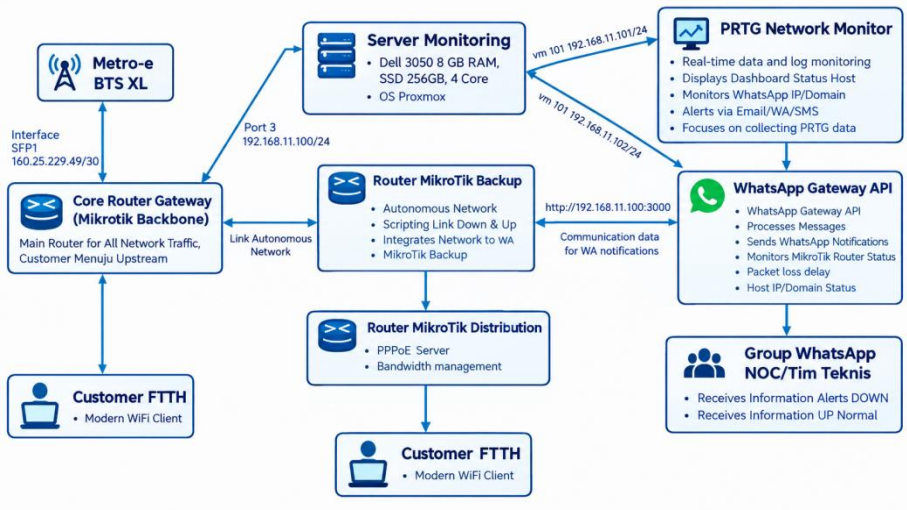

Figure 2 flowchart of the PRTG monitoring and whatsapp notification process

As shown in Figure 2, the notification workflow connects the three principal components of the system: Mikrotik, PRTG, and whatsapp gateway API. The process is

designed as an automated, event-driven mechanism. A network disturbance first occurs on the monitored infrastructure, followed by detection by the corresponding PRTG sensor. The resulting alert is then processed by the notification integration and transmitted through the whatsapp gateway API. Finally, the administrator or NOC personnel receives the notification and can initiate the appropriate troubleshooting process. This workflow allows network incidents to be communicated without requiring continuous manual inspection of the PRTG dashboard. The hardware and software components used in the implementation are summarized in Table 1.

Table 1 hardware and software components used in the system

Nu.	Component	Function
1	Mikrotik routerOS	Acts as the network gateway and primary monitoring object
2	Ubuntu linux server	Provides the server environment for the monitoring system
3	PRTG network monitor	Performs centralized monitoring of hosts, network parameters, and services
4	Whatsapp gateway API	Delivers automated network incident notifications through whatsapp
5	Network switch	Provides network connectivity and distributes traffic between devices
6	Client PC	Serves as a network endpoint for connectivity and fault-simulation testing

As presented in Table 1, the system requires a combination of network hardware and software components that operate as an integrated monitoring environment. mikrotik routerOS functions as the primary network infrastructure being monitored, while the Ubuntu Linux server provides the server environment for the monitoring implementation. PRTG functions as the central monitoring and alert-generation platform, whereas the whatsapp gateway API serves as the communication channel for automated notifications. The switch and client PC support network connectivity and controlled testing activities. The monitoring configuration focused on several parameters representing network availability and performance. Host status was used to determine whether a monitored device was reachable or unavailable. Ping and latency were used to evaluate connectivity and response time between the monitoring server and the monitored device. Packet loss was used to identify degradation in communication quality, while interface status was used to determine whether a monitored network interface was operational. Uptime was used to observe service continuity and device availability over time. System evaluation consisted of three complementary testing approaches: Black box testing, fault simulation, and response-time measurement. Black box testing was used to verify whether each functional component produced the expected output based on a given input without examining its internal implementation. The functional tests included sensor operation, alert generation, integration with the whatsapp gateway API, and delivery of notification messages.

Fault simulation was performed to evaluate the ability of the monitoring system to detect predefined network disturbances. Several scenarios were introduced in the controlled testing environment, including loss of connectivity, interface deactivation, increased latency, and other selected abnormal network conditions. The resulting PRTG status and notification behavior were then observed and recorded. The purpose of these simulations was to determine whether the system could correctly identify the simulated conditions and trigger the appropriate notification. Response-time measurement was conducted to evaluate the timeliness of the alerting mechanism. The measurement

considered the interval between the occurrence or detection of a simulated network disturbance and the receipt of the corresponding whatsapp notification. The results were recorded for each testing scenario and used to assess the responsiveness of the integrated monitoring and notification mechanism.

All fault simulations were conducted within the designated testing network rather than the production network. This approach was necessary because some testing scenarios required deliberate disruption of network connectivity or changes to device and interface conditions. Conducting these experiments in an isolated environment reduced the possibility of affecting customer services and operational SLA commitments. The controlled testing environment also enabled the same fault scenarios to be reproduced under predefined conditions. Each scenario was observed from the occurrence of the network disturbance, detection by the relevant PRTG sensor, generation of the alert, transmission through the whatsapp gateway API, and final receipt of the notification by the designated recipient. The complete sequence was documented to provide evidence for evaluating both the functional performance and responsiveness of the proposed system. Through this methodology, the research evaluated not only whether PRTG could monitor mikrotik infrastructure, but also whether the integration with the whatsapp gateway API could transform detected network events into timely and actionable notifications. The combination of functional testing, controlled fault simulation, and response-time measurement therefore provides a basis for assessing the effectiveness of the proposed real-time monitoring and automated alerting system.

Results and discussion

Existing monitoring condition and system implementation

Before the implementation of the proposed system, network monitoring at PT Nextmedia Teknologi Indonesia, Sukomoro Branch, was primarily conducted through manual inspection of mikrotik routers or through information received from customers when service disruptions occurred. This condition resulted in a reactive monitoring process because network administrators could not continuously observe the condition of all network devices. Consequently, a network problem could remain undetected until the device was manually checked or a customer reported a service interruption. The initial observation therefore confirmed the need for a centralized monitoring mechanism capable of continuously providing information about network conditions.

Following the implementation, PRTG network monitor was established as the centralized monitoring platform. The implementation began with the installation and configuration of PRTG, followed by the registration of the mikrotik devices that would be monitored. The devices were organized into appropriate device groups to simplify monitoring and management. After a device was successfully registered, relevant sensors were configured according to the monitoring requirements. PRTG subsequently collected network information periodically and presented the results through a centralized dashboard. The monitoring parameters selected for the mikrotik infrastructure are presented in Table 2.

Table 2 monitoring parameters for mikrotik infrastructure

Nu.	Parameter	Description
1	Host status	Indicates whether the monitored device is reachable or unreachable
2	Ping	Indicates basic network connectivity

Nu.	Parameter	Description
3	Packet loss	Measures the percentage of packets lost during communication
4	Latency	Measures the delay in network communication
5	Interface status	Indicates the operational condition of a network interface
6	Uptime	Indicates the duration for which a device has remained operational

As shown in Table 2, the selected parameters represent both network availability and connectivity quality. Host status and ping provide basic information about device reachability, while packet loss and latency provide additional information regarding communication quality. Interface status provides information about the operational condition of individual interfaces, whereas uptime represents the continuity of device operation. The combination of these parameters enables administrators to obtain a broader view of network conditions rather than relying solely on device availability. The resulting monitoring dashboard is presented in Figure 3.

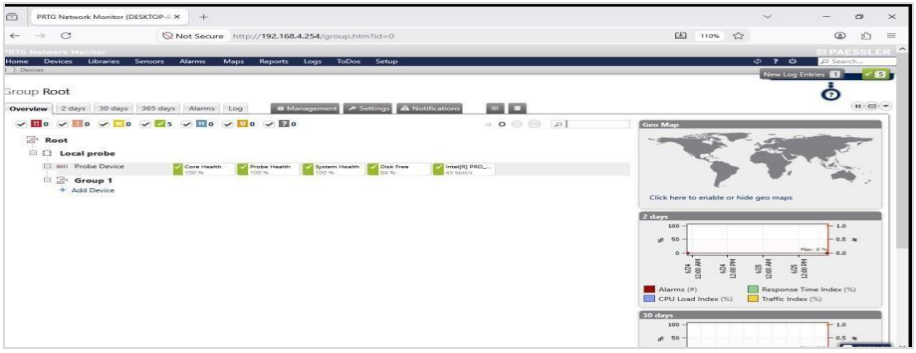

Figure 3 PRTG dashboard after network monitoring implementation

Figure 3 shows the PRTG dashboard containing device groups, monitored hosts, and sensor information. The centralized dashboard represents a substantial change from the previous manual monitoring mechanism because network information can be accessed from a single interface. Administrators can therefore identify changes in device status and sensor conditions without logging into each mikrotik device individually. This configuration also provides a more structured basis for subsequent fault detection and notification.

Functional testing results

Functional testing was conducted to determine whether the main components of the proposed system operated according to the defined requirements. Five test scenarios were used: disconnecting a mikrotik device, restoring the connection, sending an automated whatsapp notification, monitoring latency, and monitoring packet loss. The results of the functional tests are presented in Table 3.

Table 3 functional testing results

Nu.	Test scenario	Expected result	Result
1	Disconnect the mikrotik device	Host status changes to DOWN	Successful
2	Restore the mikrotik connection	Host status changes to UP	Successful
3	Send whatsapp notification	Notification is delivered automatically	Successful
4	Monitor latency	Latency data is displayed	Successful
5	Monitor packet loss	Packet-loss data is displayed	Successful

As presented in Table 3, all functional test scenarios produced the expected results. When the mikrotik connection was intentionally interrupted, PRTG successfully detected the change and displayed the host as down. When the connection was restored, the host status returned to up. The system also successfully displayed latency and packet-loss information and transmitted automated whatsapp notifications. These results indicate that the fundamental monitoring, alerting, and notification functions were successfully integrated. The process of registering a monitored device is illustrated in Figure 4.

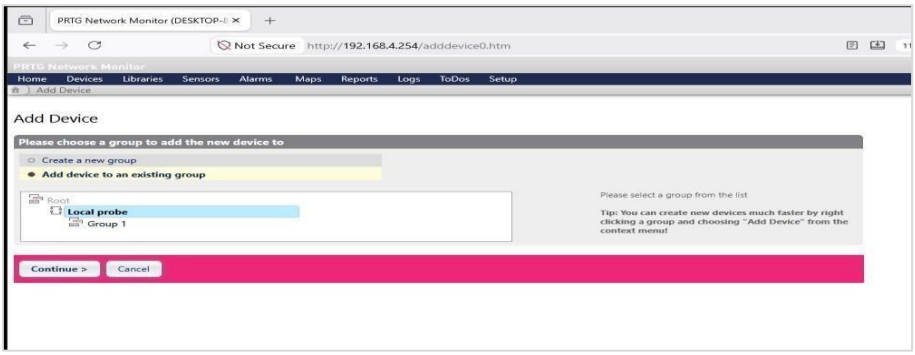

Figure 4 add device process in PRTG

Figure 4 illustrates the process of adding a mikrotik device to PRTG. This stage is essential because the monitoring process cannot begin until the device has been successfully registered and recognized by the monitoring platform. After the device was successfully added, the administrator could proceed with the configuration of sensors corresponding to the required monitoring parameters. The sensor configuration stage is shown in Figure 5.

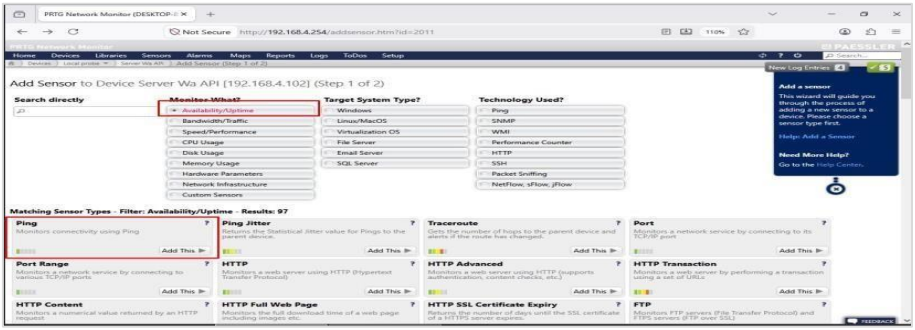

Figure 5 sensor selection and configuration in PRTG

Figure 5 demonstrates the selection and addition of sensors used to obtain information from the monitored host. The administrator can select sensors according to the required monitoring objectives, including sensors related to connectivity, packet loss, latency, interface conditions, and other relevant parameters. Appropriate sensor selection is important because the quality and relevance of the information displayed by PRTG depend on the monitoring functions configured for each device.

Network fault detection results

After the functional tests were completed, fault-detection tests were performed to evaluate the system under simulated abnormal network conditions. Four scenarios were selected based on conditions that may occur in network operations: interface

down, link disconnection, high packet loss, and high latency. The purpose of this stage was to determine whether the monitoring system could identify changes from normal network conditions and subsequently initiate the notification process. The results of the fault-detection tests are summarized in Table 4.

Table 4 network fault detection results

Nu.	Fault scenario	PRTG detection	Whatsapp notification
1	Interface down	Detected	Delivered
2	Link disconnection	Detected	Delivered
3	High packet loss	Detected	Delivered
4	High latency	Detected	Delivered

Table 4 demonstrates that all four simulated fault conditions were successfully detected by PRTG and followed by whatsapp notification delivery. Interface-down and link-disconnection scenarios provided information regarding device or connection availability, while high packet loss and high latency provided indications of degraded network quality. The results demonstrate that the proposed system was capable of monitoring both binary availability conditions and performance-related network conditions. The notification output generated by the integrated system is shown in Figure 6.

Figure 6 up/down notification delivered through whatsapp gateway API

As illustrated in Figure 6, the notification contains essential incident information, including the device name, IP address, up/down status, and event time. The concise notification format enables technicians to identify the affected device and its condition without continuously accessing the PRTG dashboard. This mechanism is particularly useful when administrators or NOC personnel are not actively viewing the monitoring interface.

System response time and stability

Response-time testing was performed to evaluate the timeliness of the integrated monitoring and notification mechanism. The measurement focused on three stages: the time required for PRTG to detect a network disturbance, the time required to transmit the notification through the whatsapp gateway API, and the resulting total response time. The test results are presented in Table 5.

Table 5 system response-time testing results

Nu.	Measurement component	Average time
1	PRTG network fault detection	5–10 seconds
2	Whatsapp notification delivery	3–5 seconds
3	Total system response	8–15 seconds

The results in Table 5 show that PRTG detected the simulated network disturbances within approximately 5–10 seconds, while whatsapp notification delivery

required approximately 3–5 seconds. The resulting total response time was approximately 8–15 seconds. Within the controlled testing environment, this indicates that the integrated system was capable of converting a network event into an operational notification within a relatively short period. It is important to interpret these values within the context of the testing environment. The measured response time represents the combined behavior of network detection, PRTG processing, integration processing, and whatsapp gateway delivery. Therefore, the result should not be interpreted as a universal response time for all deployment environments. Variations in network conditions, PRTG configuration, server performance, gateway processing, and Internet connectivity may produce different results in production environments. In addition to response time, system stability was observed while monitoring several devices simultaneously. During the testing period, the PRTG dashboard continued to display device and sensor statuses, monitoring processes remained operational, and whatsapp notifications were delivered normally. No significant delays or failures were observed within the scope of the conducted tests. The whatsapp gateway interface used in the integration is presented in Figure 7.

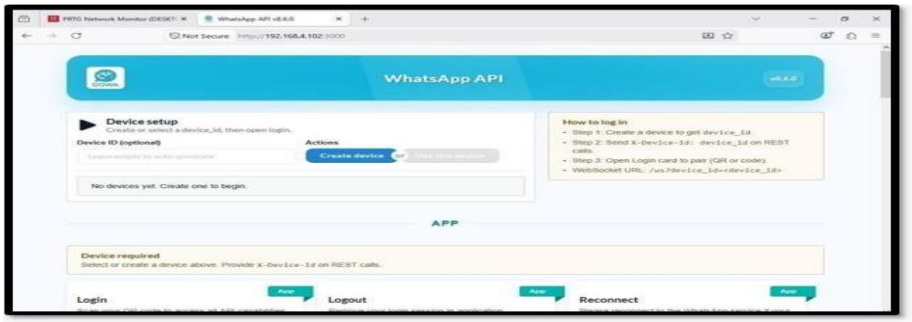

Figure 7 whatsapp gateway API interface after integration

Figure 7 shows the whatsapp gateway API interface used as the communication layer between the monitoring system and the whatsapp recipient. The successful delivery of notifications confirms that the integration mechanism was operational. The gateway therefore served as an important component in extending PRTG's monitoring capability from a dashboard-based system into an automated communication mechanism.

Sensor and monitoring verification

Further verification was performed to ensure that the configured sensors were actively collecting information from the monitored hosts. Figure 8 shows an example of a sensor that was successfully added and had started monitoring a host.

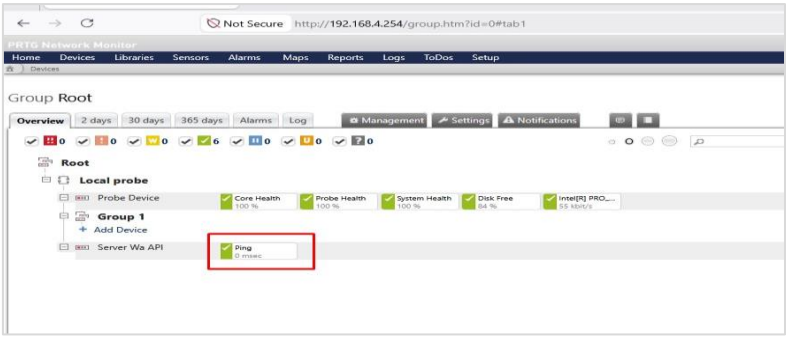

Figure 8 successfully configured sensor monitoring a host

As shown in Figure 8, the sensor was successfully registered and began collecting information from the monitored host. This result confirms the operational relationship between device registration, sensor configuration, data collection, dashboard visualization, and alert generation. The sensor therefore represents the fundamental measurement layer of the proposed monitoring architecture.

Comparison before and after implementation

To evaluate the practical impact of the implementation, the monitoring conditions before and after system deployment were compared. Before implementation, network information was obtained primarily through manual device inspection or customer reports. After implementation, network conditions could be monitored centrally through PRTG, while selected incidents could be communicated automatically through whatsapp. The comparison is presented in Table 6.

Table 6 comparison of monitoring conditions before and after implementation

Aspect	Before implementation	After implementation
Monitoring mechanism	Manual device inspection and customer reports	Centralized monitoring through PRTG dashboard
Fault detection	Relatively delayed and reactive	Faster through sensors and automated alerts
Information delivery	Dependent on manual communication	Automated through whatsapp gateway API
Technician response	Dependent on when the incident became known	Supported by faster incident notification
Status documentation	Limited and not centralized	Available through dashboard and monitoring records

Table 6 indicates that the principal improvement was not simply the replacement of manual monitoring with a software platform, but the transformation of the overall information flow. Previously, the detection process depended heavily on periodic inspection and external reports. Following implementation, the monitoring process became centralized, sensor-based, and supported by automated communication. This change reduces the dependency on continuous manual observation and provides technicians with earlier information about network abnormalities.

The findings demonstrate that the integration of PRTG network monitor with a whatsapp gateway API can improve the operational mechanism for monitoring mikrotik infrastructure within the tested ISP environment. The functional tests showed that the system successfully monitored network parameters, detected simulated disturbances, and delivered automated notifications. More importantly, the response-time test demonstrated that the complete monitoring-to-notification process could operate within approximately 8–15 seconds under the conditions of the testing environment. This finding indicates that the proposed architecture can transform network monitoring from a predominantly reactive process into a more proactive incident-awareness mechanism.

The findings are consistent with previous research concerning centralized network monitoring. Fuzi et al. (2023) demonstrated the application of PRTG network monitor for monitoring networks and servers, supporting the use of centralized monitoring to improve infrastructure visibility. Similarly, Daud et al. (2024) reported the use of PRTG for enterprise infrastructure monitoring. The present research extends these findings by incorporating automated whatsapp-based incident communication

into the monitoring workflow rather than limiting the monitoring function to dashboard visualization.

The findings are also related to previous studies on mikrotik-based monitoring. Ardiansyah et al. (2026) discussed the utilization of mikrotik RouterOS for network monitoring and an API-based monitoring approach. Their findings support the feasibility of using mikrotik capabilities as part of an automated monitoring environment. In the present study, however, mikrotik infrastructure is positioned as the monitored network object within a centralized PRTG architecture. At the same time, the notification mechanism is implemented through an external whatsapp gateway API. This creates an end-to-end workflow that connects infrastructure monitoring with operational communication.

The successful notification results are consistent with research on automated network alerts. Safinah (2026) investigated automated notification mechanisms for computer network monitoring, while Febriani et al. (2025) examined the implementation of a whatsapp gateway API for network monitoring. Kavitha et al. (2024) also highlighted the potential of whatsapp as a real-time notification medium. The present findings strengthen these observations by demonstrating their application in an integrated PRTG–mikrotik environment and by measuring the resulting notification response time. Therefore, the contribution of this research is not merely the use of whatsapp as a notification medium, but the integration of automated alert generation, message transmission, and network-event detection into a single operational workflow.

The detection of interface failure, link disconnection, high packet loss, and high latency also supports the broader argument that real-time monitoring can improve awareness of abnormal network conditions. Kebande et al. (2021) emphasized the relevance of real-time monitoring in modern network environments, while Zhang et al. (2025) discussed real-time monitoring combined with alert mechanisms. In addition, Hassan & Mansour (2024) highlighted the relationship between delayed fault detection and network downtime in ISP environments. The present research provides an implementation-level response to this issue by establishing an automated mechanism that detects selected network disturbances and immediately communicates them to operational personnel.

Nevertheless, the results should be interpreted within the limitations of the research. The implementation was conducted in a testing environment rather than the production network, meaning that the measured response times and stability results may differ under actual production traffic and larger-scale deployment. The notification mechanism also depends on the availability of the Internet connection and whatsapp gateway API service. Furthermore, the monitoring parameters used in this study represent selected aspects of network availability and performance and do not cover every possible network-security or infrastructure-performance indicator.

Overall, the findings indicate that the proposed integration provides a practical improvement over the previous manual monitoring mechanism. PRTG provides centralized visibility and automated event detection, while the whatsapp gateway API extends this capability by delivering incident information directly to technicians. The combination reduces dependence on manual dashboard observation, improves the timeliness of fault information, and provides a more structured basis for network incident response. In this sense, the main value of the implementation lies in the integration of monitoring, detection, notification, and operational awareness into a continuous workflow for mikrotik infrastructure.

Conclusions

This research successfully implemented a real-time network monitoring and automated whatsapp alerting system for mikrotik infrastructure using PRTG network monitor at PT Nextmedia Teknologi Indonesia, Sukomoro Branch. The results demonstrate that PRTG was able to centrally monitor key network parameters, including host status, ping, packet loss, latency, interface status, and uptime. Functional and fault-detection testing showed that the system successfully identified simulated conditions such as interface failure, link disconnection, high packet loss, and high latency, and subsequently delivered automated notifications through the whatsapp gateway API. The response-time evaluation showed a PRTG detection time of approximately 5–10 seconds, a whatsapp notification delivery time of 3–5 seconds, and a total system response time of approximately 8–15 seconds under the testing conditions. These findings indicate that the proposed system can provide faster network incident awareness compared with the previous manual and reactive monitoring process.

From a practical perspective, the integration of PRTG and whatsapp gateway API contributes to a more centralized, automated, and proactive monitoring mechanism for mikrotik infrastructure. The system reduces dependence on continuous manual dashboard observation and enables NOC personnel or administrators to receive network incident information directly through whatsapp, thereby supporting faster identification and initial response to network problems. However, the findings should be interpreted within the scope of the study. The implementation and testing were conducted on the internal and testing networks of the Sukomoro Branch and therefore do not fully represent the complexity of the production network. The system also depends on Internet connectivity and the availability of the whatsapp gateway API. In addition, the monitoring configuration focused primarily on basic network availability and performance parameters and did not comprehensively address quality of service, application performance, advanced security mechanisms, notification-data encryption, API authentication hardening, or protection against cyberattacks. The study also did not compare PRTG performance with other network monitoring platforms.

Future research should therefore evaluate the proposed system in a larger-scale production or multi-site ISP environment to determine its scalability, reliability, and performance under actual operational traffic. Further studies could expand the monitoring parameters to include comprehensive Quality of Service indicators, application-level performance, traffic analysis, and security-related metrics. The whatsapp integration could also be strengthened through encrypted notification data, more robust API authentication, access control, and security mechanisms against unauthorized access or cyberattacks. In addition, comparative experiments involving PRTG and alternative network monitoring platforms could be conducted using standardized indicators such as detection accuracy, response time, resource consumption, scalability, reliability, and notification performance. Such improvements would provide a broader empirical basis for assessing the effectiveness and applicability of integrated real-time network monitoring and automated alerting systems in ISP environments.

References

Aara, F. J., Ninganagouda, N., & Suresh, S. (2019). Computer Network Services And Applications. *Journal of Scientific Research and Technology*, *3*(1), 1–9. https://jsrtjournal.com/index.php/JSRT/article/view/192

Aldea, C. L., Bocu, R., & Solca, R. N. (2023). Real-Time Monitoring and Management of Hardware and Software Resources in Heterogeneous Computer Networks through an Integrated System Architecture. *Symmetry*, *15*(6), 1134. https://doi.org/10.3390/sym15061134

Ardiansyah, A., Hartinah, H., & Saputra, W. (2026). Middleware-Assisted Dynamic IP Blacklisting for MikroTik RouterOS Using Flask and AbuseIPDB: A Single-Site Operational Case Study. *Journal of Electrical Engineering and Informatics*, *4*(1), 14–23. https://doi.org/10.59562/jeeni.v4i1.14043

Arvianto, D., Kamisutara, M., & Kristiana, W. A. (2024). Development Of Paessler Router Traffic Grapher Network Monitoring Application. *IJEEIT: International Journal of Electrical Engineering and Information Technology*, *7*(2), 80–87. https://jurnal.narotama.ac.id/index.php/ijeeit/article/view/2912

Averineni, A. (2025). Leveraging Machine Learning for Anomaly Detection in Telecom Network Management. *Journal of Computer Science and Technology Studies*, *7*(4), 08–20. https://doi.org/10.32996/jcsts.2025.7.4.2

Bojović, Ž. C., & Bojović, P. D. (2025). Software-Defined Networking: Architecture, Protocols, and Applications. *Foundations and Trends® in Networking*, *15*(1–3), 1–317. https://doi.org/10.1561/1300000077

Christanto, F. W., Rivaldo, S. V., Laia, F., & Firdaus, E. A. (2026). Debouncing-Based Automated Network Failover System with Multi-Channel Real-Time Notifications. *International Journal on Perceptive and Cognitive Computing*, *12*(2), 40–50. https://doi.org/10.31436/ijpcc.v12i2.705

Daud, A. Y., Tan, J. X., & Ooi, W. J. (2024). Paessler Router Traffic Grapher (PRTG) Network Monitoring: An Implementation Process in Vitrox. *Journal of Digital System Development*, *2*(2), 1–11. https://doi.org/10.32890/jdsd2024.2.2.1

Fathima, A., & Devi, G. S. (2024). Enhancing university network management and security: a real-time monitoring, visualization & cyber attack detection approach using Paessler PRTG and Sophos Firewall. *International Journal of System Assurance Engineering and Management*. https://doi.org/10.1007/s13198-024-02448-y

Febriani, S., Arrozaq, M. A., Nugraha, R., & Lestari, A. C. (2025). Intelligent Flood Monitoring System Using IoT and Real-Time Notifications. *Jurnal Informasi, Sains Dan Teknologi*, *8*(2), 570–582. https://doi.org/10.55606/isaintek.v8i2.384

Fuzi, M. F. M., Mahdzir, M. F. M., Halim, I. H. A., & Ruslan, R. (2023). Performance Analysis of Open-Source Network Monitoring Software in Wireless Network. *Journal of Computing Research and Innovation*, *8*(2), 31–44. https://jcrinn.com/index.php/jcrinn/article/view/375

Hassan, A. B., & Mansour, M. (2024). Performance Evaluation of Bidirectional Forwarding Detection (BFD) over the Virtual Router Redundancy Protocol (VRRP). *Procedia Computer Science*, *251*, 256–264. https://doi.org/10.1016/j.procs.2024.11.108

Hegde, P., & Varughese, R. J. (2022). Predictive Maintenance in Telecom: Artificial Intelligence for predicting and preventing network failures, reducing downtime and maintenance costs, and maximizing efficiency. *Journal of Mechanical, Civil and Industrial Engineering*, *3*(3), 102–118. https://doi.org/10.32996/jmcie.2022.3.3.13

Kampourakis, K. E., Gkioulos, V., Kavallieratos, G., & Lin, J.-C. (2025). Digital Twin-Enabled Incident Detection and Response: A Systematic Review of Critical Infrastructures Applications. *International Journal of Information Security*, *24*(5),

194. https://doi.org/10.1007/s10207-025-01113-0

Kavitha, P., Yatheesh.K.C., Anand, A., Sreenivasan, S., Mohammed S, H., Borah, N., & Saikia, D. (2024). The Development of Early Flood Monitoring and a WhatsApp-Based Alert System for Timely Disaster Preparedness and Response in Vulnerable Communities. *CC 2023*, 18. https://doi.org/10.3390/engproc2024062018

Kebande, V. R., Karie, N. M., & Ikuesan, R. A. (2021). Real-time monitoring as a supplementary security component of vigilantism in modern network environments. *International Journal of Information Technology*, *13*(1), 5–17. https://doi.org/10.1007/s41870-020-00585-8

Khawar, M. W., Salman, W., Shaheen, S., Shakil, A., Iftikhar, F., & Faisal, K. M. I. (2024). Investigating the Most Effective AI/ML-Based Strategies for Predictive Network Maintenance to Minimize Downtime and Enhance Service Reliability. *Spectrum of Engineering Sciences*, *2*(4), 115–132. https://thesesjournal.com.medicalsciencereview.com/index.php/1/article/view/66

Lee, J. A., Wachira, B. W., Kennedy, J., Asselin, N., & Mould-Millman, N.-K. (2024). Utilisation of WhatsApp for Emergency Medical Services in Garissa, Kenya. *African Journal of Emergency Medicine*, *14*(1), 38–44. https://doi.org/10.1016/j.afjem.2024.01.002

Mahmood, A., & Yuksel, M. (2025). Resource Sharing on the Internet: A Comprehensive Survey on ISP Peering. *ACM Computing Surveys*, *57*(7), 1–37. https://doi.org/10.1145/3715906

Murphy, K., Lavignotte, A., & Lepers, C. (2024). Fault Prediction for Heterogeneous Telecommunication Networks Using Machine Learning: A Survey. *IEEE Transactions on Network and Service Management*, *21*(2), 2515–2538. https://doi.org/10.1109/TNSM.2023.3340351

Nurcahyanto, H., Irawati, I. D., Hantoro, G. D., Purwanto, Y., & Awaluddin, M. (2025). A Systematic Literature Review on Machine Learning for Network Reliability in Telecommunications Industry. *IEEE Access*, *13*, 201098–201124. https://doi.org/10.1109/ACCESS.2025.3634554

Safinah, S. (2026). Automated Attendance Notifications and Parental Engagement: Evaluating a WhatsApp Gateway Information System in Islamic Boarding Schools. *International Journal of Educational Qualitative Quantitative Research*, *5*(1), 208–223. https://journal.qqrcenter.com/index.php/ijeqqr/article/view/229

Shahid, K., Ahmad, S. N., & Rizvi, S. T. H. (2024). Optimizing Network Performance: A Comparative Analysis of EIGRP, OSPF, and BGP in IPv6-Based Load-Sharing and Link-Failover Systems. *Future Internet*, *16*(9), 339. https://doi.org/10.3390/fi16090339

Tanimu, J. A., Bendiab, G., Kanta, A., & Shiaeles, S. (2026). AI-Driven Threat Detection and Automated Incident Response for Enhancing Network Security. *Network*, *6*(2), 32. https://doi.org/10.3390/network6020032

Troia, S., Mazzara, M., Savi, M., Zorello, L. M. M., & Maier, G. (2022). Resilience of Delay-Sensitive Services With Transport-Layer Monitoring in SD-WAN. *IEEE Transactions on Network and Service Management*, *19*(3), 2652–2663. https://doi.org/10.1109/TNSM.2022.3191943

Zhang, H., Zhang, R., & Sun, J. (2025). Developing real-time IoT-based public safety alert and emergency response systems. *Scientific Reports*, *15*(1), 29056. https://doi.org/10.1038/s41598-025-13465-7