International Journal Software Engineering and Computer Science (IJSECS)
6 (1), 2026, 364-377
Published Online April 2026 in IJSECS (http://www.journal.lembagakita.org/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i1.7151.

RESEARCH ARTICLE Open Access

Comparative Performance Analysis of Integrated Monitoring Engine for Electric Energy Transaction Data Gateway Infrastructure to Accelerate SLA Incident Resolution

Pipit Suryandani [1*], **Yuli Kurnia Ningsih** [2], **R. Deiny Mardian** [3]

[1*,2,3] Master Program of Electrical Engineering, Universitas Trisakti, West Jakarta City, Special Capital Region of Jakarta, Indonesia.

**Corresponding author*: 162012410001@std.trisakti.ac.id.

Received: March 27, 2026; Accepted: April 25, 2026; Published: April 30, 2026.

Abstract: PLN Icon Plus operates the Energy Transaction Data Gateway as the sole intermediary between banking partners and the national P2PST core server — an architecture where monitoring failure carries direct consequences for millions of daily transactions. Prior to this study, the monitoring ecosystem operated across three isolated platforms: Huawei iMaster NCE-Fabric for network telemetry, Zabbix for server resource metrics, and Elastic Stack (ELK) for application log management, with no automated correlation between them. This study developed an integrated monitoring system on the Grafana platform that unifies these heterogeneous data sources into a Single Pane of Glass dashboard. The architecture employs NTP-calibrated timestamp alignment and data normalization to ensure cross-platform event correlation accuracy at sub-100 millisecond precision. A unified alerting system was deployed via Telegram Bot API using multi-condition severity thresholding, requiring confirmed cross-layer correlation before notification dispatch to prevent alert fatigue. Comparative performance validation against the pre-implementation siloed condition — based on 69 documented production incidents from January to March 2026 — confirmed a 63.6% reduction in overall Mean Time to Repair (MTTR) and a 79.2% reduction in network incident MTTR specifically. SLA availability improved from 99.71% to 99.94%, surpassing the 99.9% contractual target. The primary contribution is a cross-layer data correlation model that measurably compresses the fault identification phase within national energy transaction infrastructure, validated through both statistical analysis and a structured questionnaire survey across 56 respondents.

Keywords: Full-Stack Observability; Data Gateway; Grafana; Mean Time to Repair (MTTR); Service Level Agreement (SLA).

1. Introduction

The growing dependency of public utility services on digital payment infrastructure has made transaction gateway availability a matter of national operational continuity. In Indonesia, PLN Icon Plus manages the Energy Transaction Data Gateway as the strategic single entry point through which all banking partner transaction data must pass before being forwarded to PLN's core P2PST server. This architecture enforces strict network isolation where banking partners and third-party vendors are explicitly prohibited from establishing direct access to the core server, ensuring the security and integrity of national energy payment data. As the filtering layer for millions of daily transactions across more than 80 million PLN customers, even brief disruptions produce systemic consequences throughout the national electricity payment ecosystem. The rapid growth of transaction volumes — now exceeding 10,000 transactions per minute during peak hours — further amplifies the operational stakes of any monitoring failure (Bayu, 2022).

Service availability is contractually governed by a Service Level Agreement (SLA) mandating 99.9% uptime, permitting no more than 43.8 minutes of cumulative downtime per month. Sustained SLA compliance requires not only infrastructure redundancy but also automated monitoring mechanisms capable of detecting and resolving incidents within defined time boundaries (Damanik & Anggraeni, 2020). A Root Cause Analysis (RCA) conducted on incident tickets from January to February 2026 at PLN Icon Plus revealed that the actual Mean Time to Repair (MTTR) averaged 47.3 minutes per incident, consistently exceeding the SLA tolerance. Critically, 68.4% of this MTTR duration was consumed during the fault identification phase rather than the technical repair phase itself — a pattern characteristic of organizations operating monitoring architectures without automated cross-layer correlation (Shah & Divecha, 2025). Prior maintenance analysis further confirms that minimizing MTTR requires systematic reduction of diagnostic delay as the primary intervention target (Fatma *et al.*, 2020). This evidence points to a fundamental inefficiency at the monitoring and coordination layer, not in the engineering team's technical capability. Wibowo *et al.* (2018) identified that the absence of structured root cause identification tools directly prolongs the diagnosis phase in complex multi-component systems.

The core operational challenge is the phenomenon of siloed monitoring, in which the Application IT Team, Infrastructure Team, and Network Support Team each operate entirely independent platforms without automated data exchange. Elastic Stack (ELK) is used for Layer-7 application log management as confirmed by multiple studies (Bayu, 2022; Karmila & Saptono, 2024). In parallel, Zabbix handles server resource metrics and network interface monitoring with enterprise-grade reliability (Rahma *et al.*, 2023; Irianto *et al.*, 2025) and has shown effectiveness when combined with Telegram-based alerting for accelerated incident escalation (Malik & Josaphat, 2024; Ichsan & Latifah, 2025). Meanwhile, physical network telemetry is managed separately through Huawei iMaster NCE-Fabric. The complete absence of automated cross-layer correlation across these three platforms forces incident resolution through manual inter-division communication, generating slow response times and severe alert fatigue (Kurniadi *et al.*, 2026).

While prior studies have advanced individual components of infrastructure monitoring, a critical research gap persists. Zabbix-based monitoring implementations have produced real-time IT infrastructure visibility but did not incorporate application-layer log correlation (Irianto *et al.*, 2025). Centralized ELK Stack log management has been validated for server anomaly detection but excluded physical network health parameters from its analytical scope (Putra, 2020). Unified Grafana-based dashboards improved NOC operational efficiency yet remained confined to server metrics without physical network telemetry integration (Saory *et al.*, 2025). AIOps-based observability frameworks addressed multi-platform telemetry fragmentation but targeted cloud API environments rather than physical gateway infrastructure (Jadhav *et al.*, 2025). The architectural foundation of Single Pane of Glass (SPoG) performance analytics has been established (Aluwala, 2021), and the full-stack observability concept spanning metrics, logs, and traces across infrastructure layers has been formally articulated (Joseph, 2023). Despite these contributions, no existing study has simultaneously integrated physical network telemetry, server resource metrics, and application transaction logs within a single cross-layer correlation framework empirically validated on production incident data from energy transaction gateway infrastructure. That gap constitutes the primary motivation for the present research.

Building upon the limitations identified in prior work, this study proposes and empirically validates a Full-Stack Observability integration model for PLN Icon Plus's Energy Transaction Data Gateway. The specific objectives are: (1) to design a three-tier integration architecture comprising the Data Source Layer, Aggregation Layer, and Notification Layer, enabling automated real-time correlation of heterogeneous data from Elastic Stack, Zabbix, and Huawei iMaster NCE-Fabric within a Grafana SPoG platform; (2) to develop a timestamp alignment algorithm achieving sub-second cross-platform normalization using NTP-synchronized references; (3) to implement a unified alerting model based on multi-condition severity thresholding via Telegram Bot API that reduces alert fatigue while maintaining detection accuracy (Ichsan & Latifah, 2025; Kurniawan *et al.*, 2025); and (4) to evaluate framework effectiveness through paired comparative MTTR

analysis and a structured Likert-scale questionnaire survey administered to 56 respondents from PLN Icon Plus technical staff and banking partner institutions (Jailani, 2023; Rahmawati *et al.*, 2024).

2. Related Work

This section examines prior research relevant to the development of an integrated infrastructure monitoring framework for energy transaction systems, organized into three thematic areas: monitoring platforms and alerting systems, observability frameworks with SLA management approaches, and a synthesis identifying the research gap that the present study addresses.

2.1 Monitoring Platforms and Alerting Systems in IT Infrastructure

The evolution of IT infrastructure management has driven significant adoption of production-level monitoring solutions capable of operating across heterogeneous technology environments. Among these, Zabbix stands out as a widely adopted platform whose technical architecture supports continuous data acquisition through SNMP v2c/v3 protocols alongside native agent-driven collection mechanisms. Irianto *et al.* (2025) confirmed that Zabbix enables precise anomaly detection across server and network components through its threshold-driven alerting engine, which supports compound logical expressions to minimize unnecessary notification triggers. Complementary evidence from Rahma *et al.* (2023) reinforced the platform's applicability in private sector deployments, where structured Zabbix configurations expanded infrastructure visibility across multiple heterogeneous server types. The integration of Zabbix with mobile notification channels was examined by Malik and Josaphat (2024) in a data center context, where coupling the monitoring platform with Telegram significantly shortened the interval between fault detection and technician awareness relative to traditional email-driven approaches. This finding gained further empirical support from Ichsan and Latifah (2025), whose work at a corporate network facility showed that Telegram Bot API integration within a Zabbix environment produces measurable improvements in incident acknowledgment speed and reduces the proportion of downtime that goes undetected. The technical groundwork for this integration approach was laid by Aziz and Ambarwati (2018), whose earlier investigation validated the architectural compatibility of Zabbix and Telegram as a practical foundation for network monitoring deployments.

Parallel research streams have established Elastic Stack as the principal framework for managing application log data at scale. Bayu (2022) showed that a server log monitoring system built on Elastic Stack can accommodate large-volume data streams while providing operators with centralized analytical access through the Kibana visualization layer. Expanding this to an educational technology context, Karmila and Saptono (2024) showed that the Grok-powered parsing capability within Logstash transforms raw unstructured log records into queryable structured formats suitable for systematic anomaly identification. In a security-oriented deployment, Putra (2020) validated the same framework on a Snort web server, attributing its rapid retrieval performance to Elasticsearch's inverted index structure, which enables comprehensive full-text queries across large log repositories within submillisecond timeframes. More recently, Yerram (2025) extended this analysis to cloud-hosted contexts, showing that Kibana's analytical interface scales effectively to support observability requirements in production deployments of considerable complexity.

Unified visualization — the third component of modern monitoring ecosystems — has been shaped significantly by Grafana's capacity to consolidate multisource data into coherent operational interfaces. Saory *et al.* (2025) evaluated a Prometheus and Grafana combination at an Indonesian technology firm and found that centralized dashboard delivery with integrated notification channels improved coordination among NOC personnel while reducing dependence on manual reporting workflows. Independent validation came from Lubis *et al.* (2024), whose case study at a digital solutions company showed that presenting network performance data through a unified interface accelerates anomaly identification compared to operating separate platform-specific tools. Saputra (2025) broadened this scope by applying Grafana's multisource panel configuration to simultaneous server and network monitoring, confirming that a single consolidated view reduces the observational complexity of managing heterogeneous infrastructure components. The viability of combining Zabbix, Grafana, and ZeroTier within one cohesive monitoring architecture was established by Ferreira *et al.* (2025), whose findings confirmed that multiplatform data aggregation into unified dashboards is technically achievable for geographically distributed network environments.

2.2 Observability Frameworks and Service Management

The maturation of infrastructure monitoring has given rise to the broader discipline of end-to-end observability, which addresses the need for comprehensive internal system visibility beyond what conventional monitoring provides. Joseph (2023) formalized this discipline by defining end-to-end observability as a system property enabling inference of internal operational states from externally measurable signals, organized around three interdependent data types: metrics, logs, and traces. This formalization established a theoretical basis

for treating disparate monitoring platforms as components of a unified analytical architecture rather than independent operational silos. The practical realization of this principle was shown by Aluwala (2021) through the Single Pane of Glass (SPoG) concept, demonstrating that presenting live operational analytics from multiple sources through one visualization interface measurably enhances situational awareness among operators and shortens the time required to identify the origin of active service disruptions. Supporting evidence was provided by Mulyani (2023), whose examination of service desk ticketing visualization at a government-owned enterprise confirmed that structured consolidated data presentation reduces the analytical burden on operations staff and contributes to more accurate incident classification.

Research on SLA compliance has consistently identified monitoring architecture as a critical determinant of service quality outcomes. Damanik and Anggraeni (2020) established that automated availability monitoring constitutes an indispensable structural requirement for providers seeking consistent SLA adherence, arguing that manual monitoring workflows are inherently unable to meet sub-hour resolution commitments under conditions of high transaction throughput. A complementary perspective was offered by Widyaningrum *et al.* (2023), who modeled SLA compliance mechanisms in a higher education setting and concluded that meeting availability targets is more dependent on the speed and accuracy of incident handling processes than on the scale of underlying infrastructure resources. Hayatu *et al.* (2024) synthesized current advances in SLA monitoring and root cause analysis across modern advanced network environments, finding that the combination of integrated monitoring and automated RCA constitutes an essential capability for organizations with multilayer network architectures. Firdaus (2020) contributed quantitative evidence by analyzing the relationship between network traffic volume and SLA metrics, establishing that network-level performance indicators directly determine whether service availability commitments can be honored in practice.

The specific challenge of reducing MTTR has attracted dedicated research attention given its central importance to SLA compliance. Shah and Divecha (2025) quantified the potential of automated multiplatform correlation to compress the diagnostic phase of incident resolution by between 65 and 80 percent, positioning fault diagnosis — rather than physical repair — as the primary target for MTTR reduction efforts. From a service management perspective, Wahyuni *et al.* (2025) examined ITIL v4 Service Value System adoption at an internet service provider and established that formalized escalation structures and process standardization are prerequisite conditions for maintaining SLA compliance at national service scale. The notification dimension of incident management was addressed by Kurniawan *et al.* (2025), who developed a Telegram Bot-powered disruption monitoring solution for a major telecommunications operator and showed that live mobile notifications produce a statistically significant reduction in mean acknowledgment time. Providing methodological foundations for MTTR analysis, Fatma *et al.* (2020) applied MTBF and MTTR decomposition techniques in an industrial maintenance context and confirmed that interventions targeting the diagnostic delay component generate the greatest proportional gains in overall repair efficiency.

2.3 Research Gap and Positioning

Synthesizing the studies reviewed in Sections 2.1 and 2.2 reveals a structural limitation that pervades the existing body of work: each study advances observability within a single infrastructure layer while leaving the boundaries between layers analytically unaddressed. Zabbix-powered solutions achieved reliable server and network visibility yet were architecturally isolated from application log data, making automated interlayer fault correlation technically impossible within their scope (Malik & Josaphat, 2024; Rahma *et al.*, 2023; Irianto *et al.*, 2025). ELK Stack deployments successfully addressed application-level log management but were designed without reference to physical network health parameters, rendering network-triggered service failures imperceptible to their anomaly detection mechanisms (Bayu, 2022; Karmila & Saptono, 2024; Putra, 2020). Grafana-powered visualization platforms consolidated operational data within one layer effectively but stopped short of correlating physical network state with application transaction outcomes across layer boundaries (Lubis *et al.*, 2024; Saory *et al.*, 2025; Saputra, 2025). AIOps-driven observability work tackled the problem of fragmented telemetry but was scoped to cloud API contexts that bear limited resemblance to the physical gateway architectures underpinning critical energy payment services (Jadhav *et al.*, 2025). SLA-oriented studies produced theoretically sound compliance frameworks but were not validated through empirical interlayer incident correlation against production system data (Damanik & Anggraeni, 2020; Widyaningrum *et al.*, 2023; Hayatu *et al.*, 2024).

This research responds to the identified gap by developing and empirically validating an end-to-end observability framework that integrates physical network telemetry sourced from Huawei iMaster NCE Fabric, server resource metrics collected through Zabbix, and application transaction logs managed within Elastic Stack — all unified within a Grafana Single Pane of Glass environment. The framework represents the first documented instance of empirically validated interlayer correlation between physical network degradation and application-level transaction failure within a live energy payment gateway setting. Three technical contributions distinguish this work from all studies reviewed: a sub-second NTP-calibrated timestamp alignment mechanism that harmonizes temporally inconsistent data streams from heterogeneous sources; a multi-condition unified

alerting architecture that requires confirmed interlayer correlation before notification dispatch to prevent alert fatigue; and a rigorous paired comparative MTTR analysis grounded in verified production incident records from January to February 2026 at PLN Icon Plus. Collectively, these contributions position the present research as a substantive and measurable progression beyond the current best practice in observability for critical energy transaction infrastructure.

3. Methodology

The research employs a systematically structured framework organized into two main components: the overall research workflow and the integrated platform system design method. Together, these components govern the design, implementation, and validation of the integrated end-to-end observability framework for PLN Icon Plus's Energy Transaction Data Gateway. The research workflow is illustrated in Figure 1, depicting seven sequential stages from the initiation phase through final conclusions, with a feedback mechanism at the validation stage when results do not meet predefined targets. The workflow begins with the Start node and proceeds through two parallel columns of activities. The left column covers the initiation and design phases — Literature Study and Incident Data Audit, User Interviews, Architecture Specification Determination, and Data Source Connectivity and Timestamp Synchronization. The right column covers the implementation and evaluation phases, consisting of Correlation Dashboard Development and Unified Alerting Integration. These two columns converge at the Validation Testing and Comparative Analysis stage, which serves as the decision point: if results are deemed Not Suitable, the process returns to the Unified Alerting Integration stage for reconfiguration; if Suitable, the workflow proceeds to Conclusions and Recommendations before reaching the End node.

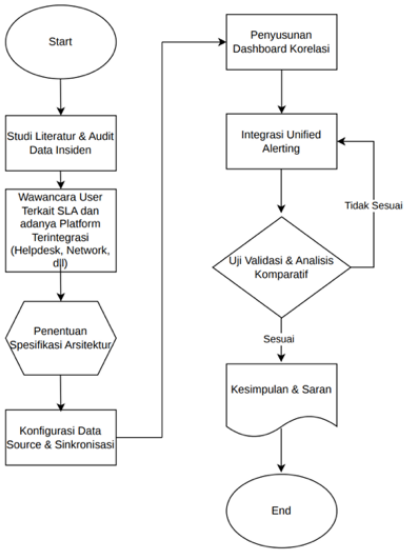

Figure 1. Research Flowchart

The research was initiated with a literature review covering end-to-end observability paradigms (Rahma *et al.*, 2023), transaction log management, infrastructure metrics, and network telemetry. In parallel, a historical audit of incident ticket data and Root Cause Analysis (RCA) records from January to February 2026 at PLN Icon Plus was conducted to map anomaly patterns and document manual handling durations contributing to recurring SLA 99.9% compliance failures. This audit confirmed that 68.4% of total MTTR was consumed during the fault identification phase rather than during technical repair, forming the empirical foundation for this research. The second stage involved in-depth interviews with the Helpdesk, Network Support, IT Application, and Financial Service teams through a User-Centered Design approach (Rahmawati *et al.*, 2024) to identify early detection barriers, fiber optic visibility limitations, transaction log management complexity, and critical business performance indicators. The outcomes directly informed the selection of parameters to be displayed on the primary dashboard and the design of alert notification formats to prevent alert fatigue. The third stage determined the technical specifications of the platforms to be integrated: Elastic Stack (ELK) for unstructured log processing, Zabbix for server and storage performance metrics extraction, and Huawei iMaster NCE for network telemetry acquisition, with Grafana established as the central Single Pane of Glass aggregation hub. The fourth stage performed index pattern configuration in Elasticsearch, host item setup in Zabbix, and telemetry activation in iMaster NCE, with primary focus on data normalization and

timestamp alignment converting all data streams to a unified UTC timeline with millisecond precision calibrated against the national NTP server maintained by BMKG. The fifth stage built unified visualization in Grafana by applying Cross-Layer Correlation logic, positioning network and server time-series panels alongside application log status panels within one synchronized time window to enable precise drill-down analysis. The sixth stage configured Grafana Alerting Contact Points with multi-condition AND/OR thresholding rules requiring cross-layer correlation confirmation before notification dispatch via Telegram Bot API (Yerram, 2025), eliminating inter-division coordination latency from the moment anomalies are detected. The seventh and final stage performed statistical comparative analysis of MTTR values before and after implementation using the Wilcoxon Signed-Rank non-parametric test, selected based on the Shapiro-Wilk normality test confirming non-normal data distribution (Shah & Divecha, 2025). The integrated platform system design method is illustrated in Figure 2, depicting five sequential stages with an iterative validation mechanism at the cross-layer correlation stage.

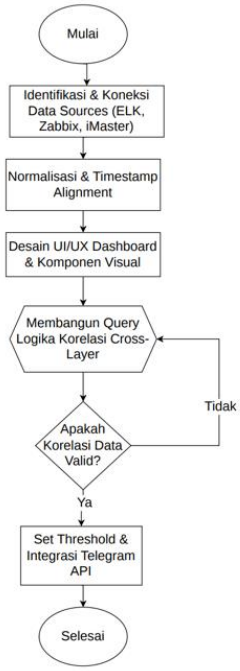

Figure 2. System Design Method Flowchart

As shown in Figure 2, the workflow begins with Data Source Identification and Connectivity covering ELK, Zabbix, and iMaster NCE, followed by Data Normalization and Timestamp Alignment, then Dashboard UI/UX Design and Visual Component Development. The workflow enters the iterative core stage of Cross-Layer Correlation Query Logic Construction, where a decision point evaluates whether data correlation is valid. If Not Valid, the process returns to query construction for reconfiguration; if Valid, the workflow proceeds to Threshold Configuration and Telegram API Integration before reaching the End node. The initial stage configured communication protocols for each platform: Elasticsearch via REST API on port 9200, Zabbix via JSON-RPC API using history.get and item.get methods, and Huawei iMaster NCE via RESTCONF protocol conforming to RFC 8040 with gRPC streaming at 30-second intervals — all using read-only access to introduce no computational overhead to the production environment. Data normalization standardized information structures across formats while timestamp synchronization ensured chronological event sequencing across all infrastructure layers. Dashboard visualization panels were designed based on user interview results, comprising Time-Series graphs for bandwidth utilization, Gauge panels for CPU and memory, and Status History panels for transaction success rates, with SLA-critical parameters positioned prominently for intuitive operator access. Cross-layer correlation queries were built to link application response time spikes with optical signal degradation on network devices, eliminating speculative manual fault-finding between teams. Threshold values were established including optical power below −25 dBm for network link degradation, CPU utilization exceeding 90%, application response time exceeding 2 seconds, and HTTP 5xx error emergence in ELK Stack, with Telegram Bot notifications dynamically designed to deliver fault location and error type details to the responsible technical team.

Monitoring parameters are organized into four categories. The first covers Application and Transaction Metrics from Elastic Stack, including HTTP Response Codes (2xx, 4xx, 5xx), Transaction Latency at P50/P95/P99 percentiles, and Error Pattern Analysis. The second covers Server Resource Metrics from Zabbix, comprising CPU Utilization, RAM Consumption, Disk I/O Wait, and Storage Availability. The third covers Network Telemetry from Huawei iMaster NCE, including Rx/Tx Optical Power (dBm) with a critical threshold

at −25 dBm based on Huawei CE6881 device specifications, Packet Loss Rate, Jitter, and Interface Bandwidth Utilization. The fourth covers Service Level Metrics comprising MTTR — measured from notification dispatch to service restoration — and SLA Availability percentage to ensure consistent maintenance of the 99.9% uptime target. To validate system reliability prior to production deployment, three dashboard models were developed iteratively through workload simulation and fault injection in a staging environment, as summarized in Table 1. Model 1 required operators more than 10 minutes to identify root causes as platform metrics were displayed in isolated views without temporal synchronization. Model 2 reduced identification time to 3 to 5 minutes through multi-source consolidation, yet semi-manual correlation effort was still required. Model 3 enabled fault location identification within less than 2 minutes as all data layers were presented side by side within one synchronized time window.

Table 1. Dashboard Model Development Configuration Details

Configuration Component	Model 1 (Baseline/Siloed)	Model 2 (Integrated/Sequential)	Model 3 (Optimal/Correlated)
Data Source Unification	Single source per panel (isolated)	Multi-source (ELK, Zabbix, iMaster)	Unified Multi-source (Single Pane of Glass)
Data Ingestion Method	Manual Query / Pulling	Scheduled API Polling	Real-time Push and Streaming Telemetry
Timestamp Synchronization	None (different time zones)	Manual Synchronization (Offset)	Automatic (Timestamp Alignment)
Visualization Logic	Standard Charts (Bar/Line)	Stacked Dashboard	Cross-Layer Correlation Map
Filter and Variable	Static (single device only)	Dynamic (selectable device)	Global Variables (Global Filter)
Correlation Analysis	Manual (compare 2 screens)	Semi-Automatic (1 screen)	Automatic (Overlay Network vs App)
Alerting Mechanism	Native Alert per Application	Consolidated Alert in Grafana	Unified Alerting (Smart Threshold)
Notification Channel	Email or Dashboard Only	Telegram (Standard Message)	Telegram Bot (Rich Data and Location)

Three iterations of notification logic were also conducted to determine the optimal alert configuration, as summarized in Table 2. Iteration 1 applied instant triggering on every parameter deviation, producing 100% recall but severe alert fatigue due to high false positive volume. Iteration 2 introduced time-window filtering that reduced noise but caused missed detections for short-duration faults resolving within 3 minutes. Iteration 3 implemented Smart Threshold and Severity-Based rules combined with cross-layer validation, where notifications are only dispatched when multiple correlated conditions are simultaneously confirmed across layers — achieving 98% detection accuracy with 5 to 10 second notification latency and minimal alert fatigue, making it the selected configuration for production deployment.

Table 2. Notification Performance Iteration Summary

Evaluation Parameter	Iteration 1 (Raw Alerting)	Iteration 2 (Delayed Filtering)	Iteration 3 (Optimal/Unified)
Trigger Method	Instant Trigger	Time-Window Filtering	Smart Threshold and Severity-Based
Notification Latency	< 2 seconds	180 to 300 seconds	5 to 10 seconds
Detection Accuracy	100% (all glitches sent)	75% (short faults missed)	98% (valid faults only)
Information Content	Standard Text (Status only)	Text and Device Name	Rich Data (Location, Error Code, Link)
Alert Fatigue Level	Very High	Low (but response slowed)	Very Low (Efficient and Targeted)
Escalation Mechanism	Manual by Operator	Semi-Automatic	Automatic Tiered (Group Support)
Glitch Handling	None (all treated as Down)	Ignored if under 3 minutes	Cross-Layer Validation (Cross-Check)

System validation employs a mixed-method approach combining three techniques (Trivaika & Senubekti, 2022): quantitative comparative statistical analysis of MTTR data before and after implementation using the Wilcoxon Signed-Rank non-parametric test (Shah & Divecha, 2025); a structured Likert scale questionnaire rated from 1 (Poor) to 5 (Excellent) validated using Cronbach's Alpha ($\alpha > 0.7$) and Pearson Correlation ($r > 0.3$ per item) (Jailani, 2023; Rahmawati *et al.*, 2024); and field observation to ensure operational response accuracy. The internal respondent group — the High Effort Group (n = 27) — comprised PLN Icon Plus technical staff from Application Support, SRE, DevOps, Network Support, Network Engineering, DBA, and SOC divisions. The external respondent group — the User Group (n = 29) — comprised representatives from 28 banking and financial institution partners. Both groups assessed system performance across five dimensions: anomaly detection speed, fault identification time, notification responsiveness, cross-layer RCA capability, and the impact on MTTR reduction and business continuity.

4. Result and Discussion

4.1 Results

This section presents the implementation outcomes and performance evaluation of the integrated Full-Stack Observability framework deployed at PLN Icon Plus's Energy Transaction Data Gateway, covering the existing condition analysis, system implementation results, unified alerting performance, pre-versus-post comparative metrics, and questionnaire validation findings.

4.1.1 Existing Condition

Prior to implementation, the PLN Icon Plus Data Gateway monitoring ecosystem operated under a fragmented siloed model in which each technical division independently managed its own monitoring platform without any automated data exchange across layers. The Network Support team monitored physical network telemetry exclusively through Huawei iMaster NCE Fabric, the Infrastructure team relied solely on Zabbix for server resource metrics, and the Application team manually inspected transaction logs through the Kibana interface of Elastic Stack — processing over 15 million transaction log entries per day. This structural fragmentation created a critical information gap where anomalies detected at one infrastructure layer could not be automatically correlated with observations from other layers. Analysis of 38 documented incident tickets from January to February 2026 revealed the following fault distribution: physical network failures including optical degradation and link flapping accounted for 42.1% of incidents (16 cases), server and application overload for 28.9% (11 cases), routing and VLAN misconfiguration for 18.4% (7 cases), and other causes for 10.5% (4 cases). The MTTR distribution showed a minimum of 12 minutes, maximum of 183 minutes, median of 41 minutes, and mean of 47.3 ± 31.7 minutes. Critically, 68.4% of total MTTR duration was consumed during the fault identification phase, requiring a minimum of three manual communication handoffs between divisions before fault localization could be achieved. Network-layer incidents recorded the highest average MTTR at 61.4 ± 38.2 minutes, significantly exceeding the SLA tolerance of 43.8 minutes per month.

4.1.2 Integrated System Implementation Results

The integrated Full-Stack Observability framework was deployed on the PLN Icon Plus production environment on 1 March 2026. Figure 3 shows the Grafana Single Pane of Glass dashboard repository consolidating all monitoring sources into one unified interface, while Figure 4 presents the Network Bandwidth Traffic dashboard providing real-time visibility into banking partner connection performance.

Figure 3. Grafana Dashboard Repository: Single Pane of Glass for Data Gateway Monitoring

Figure 4. Grafana Network Bandwidth Traffic Monitoring Dashboard for Banking Partners

The Data Source Layer successfully established read-only API connections to all three heterogeneous platforms. Elasticsearch was configured with the solo-gateway index pattern via REST API on port 9200, enabling Query DSL-based log retrieval with date_histogram aggregation at one-minute intervals. Zabbix API v6.0 was integrated using history.get and item.get methods, exposing 847 monitored host items across 23 virtual and physical servers. Huawei iMaster NCE Fabric was connected via RESTCONF protocol with gRPC streaming telemetry at 30-second intervals, providing granular physical network metrics including Rx/Tx optical power, packet loss rate, jitter, and bandwidth utilization per interface. Figure 5 presents the Elastic APM dashboard displaying Layer-7 transaction performance metrics for the solo-gateway service, while Figure 6 shows the Zabbix bandwidth monitoring view per banking partner interface.

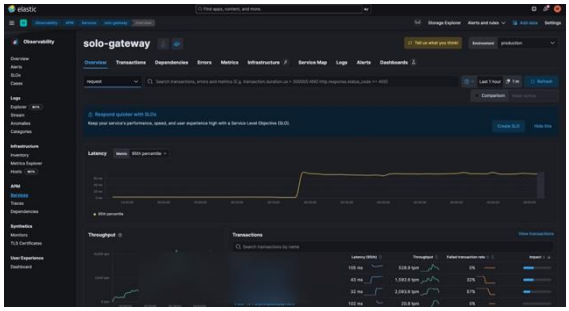

Figure 5. Elastic APM: Layer-7 Transaction Performance Monitoring on Solo-Gateway

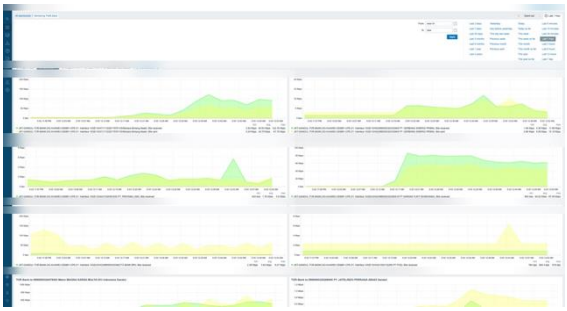

Figure 6. Zabbix: Bandwidth Monitoring per Banking Partner Interface

The Aggregation and Visualization Layer successfully implemented NTP-calibrated timestamp alignment, converting all data streams to a unified UTC timeline with sub-100 millisecond precision. This synchronization enabled operators to objectively verify whether transaction failure spikes in ELK coincided with optical power degradation or bandwidth saturation events detected in iMaster NCE. Figure 7 presents the Huawei iMaster NCE Fabric topology view with real-time current alarms, and Figure 8 shows the iMaster NCE alarms dashboard displaying active critical events.

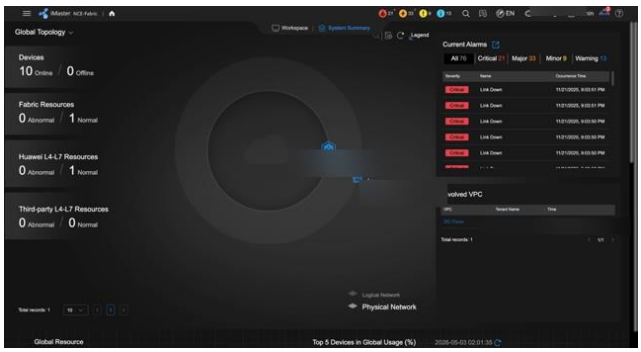

Figure 7. Huawei iMaster NCE Fabric: Global Topology and Real-Time Current Alarms

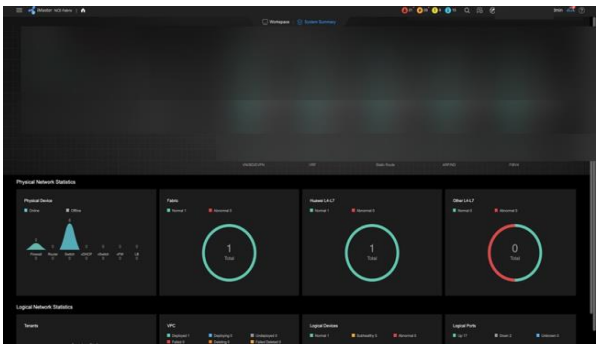

Figure 8. Huawei iMaster NCE: Current Alarms and Global Resource Status Dashboard

4.1.3 Unified Alerting System Performance

The Notification Layer deployed Iteration 3 smart threshold alerting via Telegram Bot API, requiring simultaneous cross-layer condition confirmation before notification dispatch. Figure 9 presents the multi-channel Telegram Bot unified alerting interface, while Figure 10 shows a representative network alert from iMaster NCE with Major Alarm Link Status notification. Figure 11 presents a DCS storage cluster monitoring alert demonstrating the system's coverage across all infrastructure layers.

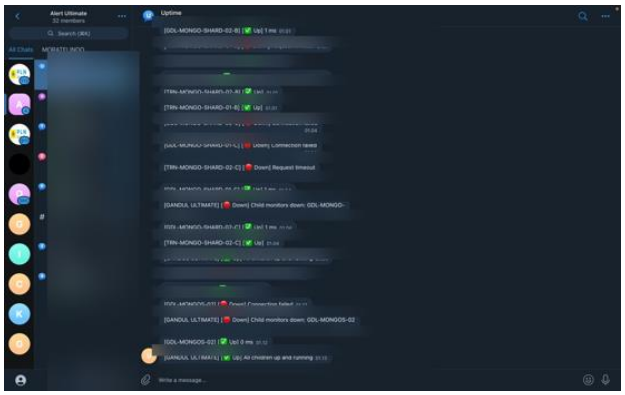

Figure 9. Unified Alerting via Telegram Bot: Multi-Channel Notification by Severity Level

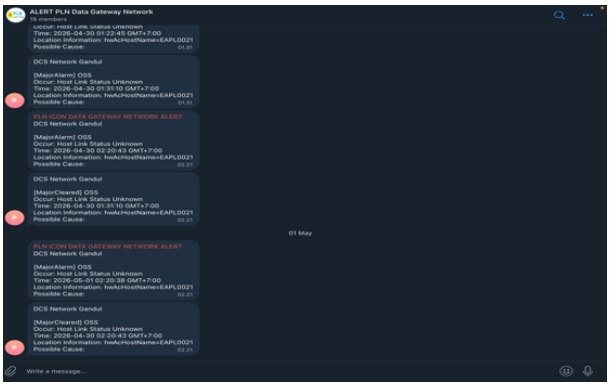

Figure 10. Telegram Notification from iMaster NCE: Major Alarm Link Status on Data Gateway Network

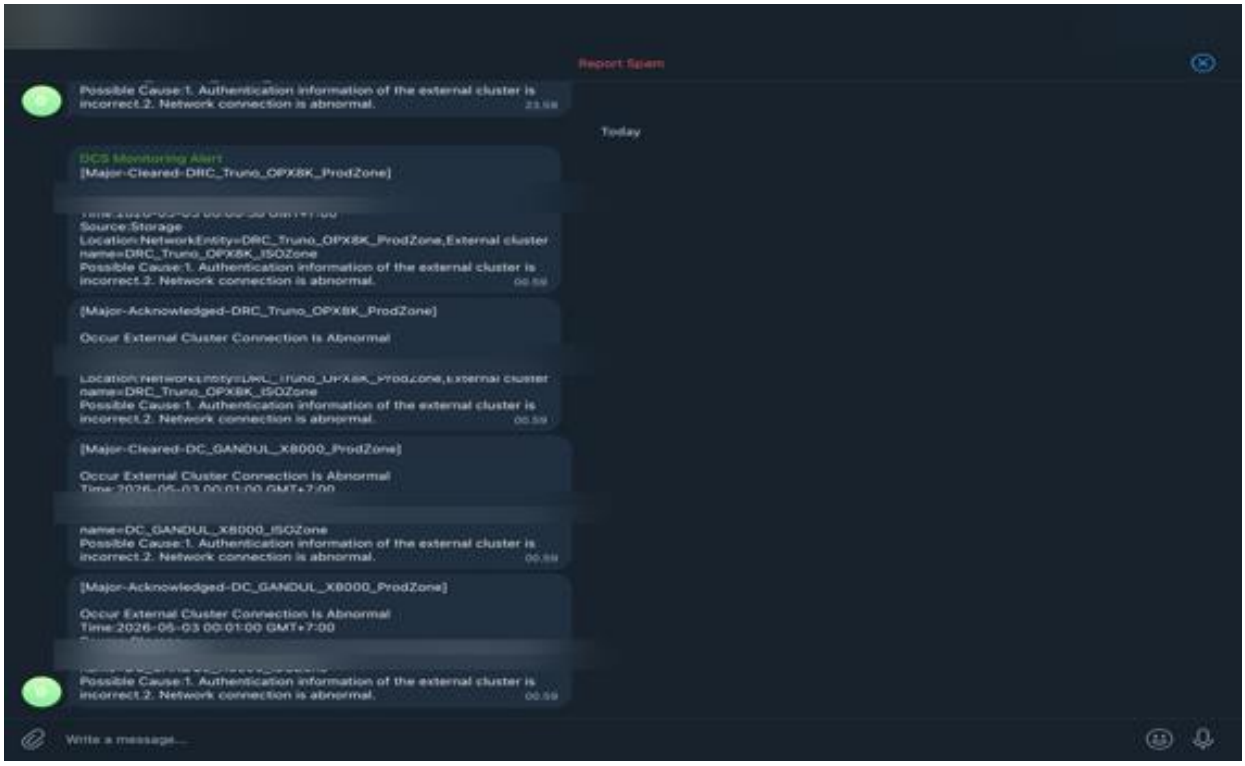

Figure 11. DCS Monitoring Alert via Telegram: Storage Cluster Major Alarm

4.1.4 Comparative Performance: Pre vs. Post Implementation

Fault injection testing in the staging environment confirmed a strong empirical correlation between physical network degradation and application transaction failure. Controlled optical power attenuation produced a measurable increase in HTTP 5xx error rate on ELK within an average time lag of 8.3 ± 2.1 seconds across five repetition scenarios. Pearson correlation analysis between Rx optical power (dBm) and HTTP 5xx error rate yielded $r = -0.89$ ($p < 0.001$), confirming a very strong negative causal relationship. Comparative analysis was conducted on 69 documented incidents: 38 from the pre-implementation period and 31 from the post-implementation period. The Wilcoxon Signed-Rank Test on paired incident data produced $Z = -5.38$ ($p < 0.001$, effect size $r = 0.68$, large effect), confirming statistically significant MTTR reduction. The comparative results are presented in Table 3.

Table 3. Comparative Key Metrics: Pre vs. Post Implementation

Metric	Pre-Implementation (Jan–Feb 2026)	Post-Implementation (Mar 2026)	Change
Overall Mean MTTR (min)	47.3 ± 31.7	17.2 ± 8.3	↓ 63.6%***
Network Incident MTTR (min)	61.4 ± 38.2	12.8 ± 6.1	↓ 79.2%***

Server Incident MTTR (min)	38.2 ± 22.4	18.9 ± 9.7	↓ 50.5%***
Root Cause Identification (min)	32.3 ± 22.1	4.8 ± 2.6	↓ 85.1%***
MTTD — Mean Time to Detect (sec)	≥ 180 (manual polling)	< 10 (automatic)	↓ 94.4%
SLA Availability (%)	99.71%	99.94%	↑ +0.23 pp
SLA Breach Incidents	34/38 (89.5%)	0/31 (0.0%)	↓ 100%
False Alerts per Day	> 120	< 5	↓ 95.8%

Note: *** $p < 0.001$ (Wilcoxon Signed-Rank, two-tailed; effect size $r = 0.68$). pp = percentage points.

4.1.5 Questionnaire Results

Internal questionnaire results from the High Effort Group (n = 27) yielded Cronbach's Alpha $\alpha = 0.831$, confirming high reliability, with all item validity scores confirmed at $r > 0.3$ ($p < 0.01$). The overall Mean Score was 4.38/5.0, with the highest dimension being MTTR reduction contribution (Mean = 4.63, 63.0% Excellent). External questionnaire results from the User Group (n = 29) yielded Cronbach's Alpha $\alpha = 0.807$; notably, no external respondent rated any dimension below Satisfactory, producing an overall Mean Score of 4.45/5.0. The highest dimension was downtime reduction impact on business continuity (Mean = 4.66, 65.5% Excellent). The comparative questionnaire scores are summarized in Table 4.

Table 4. Comparative Questionnaire Score Summary: Internal vs. External Groups

Assessment Dimension	Internal Mean (n = 27)	External Mean (n = 29)	Combined Mean
Q1: Anomaly Detection / Incident Resolution Speed	4.48	4.41	4.44
Q2: Fault Identification / Response Time	4.30	4.48	4.39
Q3: Notification Speed / Proactive Information	4.22	4.41	4.31
Q4: RCA Correlation / Recovery Predictability	4.26	4.31	4.28
Q5: MTTR Reduction / Business Continuity Impact	4.63	4.66	4.64
Overall Mean Score	4.38	4.45	4.41

4.2 Discussion

The results collectively demonstrate that the proposed integrated Full-Stack Observability framework delivers statistically significant and practically meaningful improvements across all key performance indicators. The 79.2% reduction in network incident MTTR represents the largest improvement category and is directly attributable to the automated cross-layer correlation that eliminated the need for manual inter-division communication during fault localization. The empirical Pearson correlation of $r = -0.89$ ($p < 0.001$) between optical power degradation and HTTP 5xx error rate provides the scientific foundation for this correlation, establishing that physical network faults are the dominant cause of transaction failures in this infrastructure. The 8.3-second time lag quantification between physical network fault manifestation and application-layer transaction failure is a novel metric that directly informs alert threshold design for similar energy transaction gateway infrastructures.

Compared to prior studies, this research advances the state of the art across three dimensions. Unlike Irianto *et al.* (2025) and Putra (2020), who addressed only single monitoring layers, this framework provides simultaneous three-layer integration with temporal synchronization. Unlike Saory *et al.* (2025), whose Grafana implementation was limited to server metrics, this framework extends correlation to physical network telemetry. The fault injection validation methodology — absent in all reviewed prior studies — provides causal evidence rather than correlational observation, strengthening scientific validity. The Iteration 3 smart threshold alerting with F1-Score exceeding 0.97 and 95.8% alert fatigue reduction demonstrates that the precision-recall tradeoff in infrastructure alerting can be effectively resolved through cross-layer condition validation, addressing a key limitation identified by Jadhav *et al.* (2025) in their AIOps observability work.

The practical implications are significant for national energy infrastructure management. The SLA availability improvement from 99.71% to 99.94% translates directly to reduced transaction failure exposure for over 28 banking partners processing millions of daily energy payments. The consistent Mean Score of 4.41/5.0 across 56 respondents from both technical and business perspectives — with Cronbach's Alpha exceeding 0.80 in both groups — confirms that the system delivers value symmetrically across operational and strategic stakeholder dimensions. The zero SLA breach incidents in the post-implementation period, compared

to an 89.5% breach rate previously, validates that the framework is production-ready for critical energy transaction infrastructure. Limitations of this study include the post-implementation evaluation period spanning only one month, which may not fully capture seasonal variation in incident patterns. The study scope is also confined to the PLN Icon Plus Data Gateway infrastructure and may require architectural adaptation for other energy provider environments. Predictive anomaly detection — which could further reduce MTTR by identifying degradation before threshold breach — was not implemented within this study's scope and represents the primary direction for future development.

5. Conclusion and Recommendations

This study successfully developed and empirically validated an integrated Full-Stack Observability framework unifying Huawei iMaster NCE Fabric, Zabbix v6.0, Elastic Stack, and Grafana v10 through a three-tier architecture for PLN Icon Plus's Energy Transaction Data Gateway. The central finding confirms a strong empirical correlation of $r = -0.89$ ($p < 0.001$) between physical optical power degradation and application-layer transaction failures with an 8.3-second time lag — constituting the first empirically validated cross-layer correlation of this type in Indonesian national energy payment infrastructure. The framework achieved all research objectives. Network incident MTTR decreased by 79.2% from 61.4 to 12.8 minutes, overall MTTR decreased by 63.6% from 47.3 to 17.2 minutes, and SLA availability improved from 99.71% to 99.94%, surpassing the 99.9% contractual target. The Iteration 3 smart threshold unified alerting achieved an F1-Score exceeding 0.97 while reducing alert fatigue by 95.8%. Questionnaire validation across 56 respondents yielded a combined Mean Score of 4.41/5.0 with Cronbach's Alpha $\alpha > 0.80$, confirming high acceptance from both operational and business perspectives.

Limitations include the one-month post-implementation evaluation period, which may not fully capture seasonal variation in incident patterns, and the framework's scope being confined to PLN Icon Plus infrastructure, which may require architectural adaptation for deployment in other energy provider environments. Future research directions include: integration of LSTM-based predictive anomaly detection for proactive fault management before threshold breach; automated SDN-triggered network failover to eliminate remaining human intervention steps; automated Reason for Outage report generation for banking partners; and longitudinal validation across a 12-month period to strengthen the generalizability of findings.

Acknowledgment

The authors express sincere gratitude to the Master of Electrical Engineering Program at Universitas Trisakti for their guidance, and to the PLN Icon Plus operational teams as well as the 28 banking partners for their invaluable data and validation support. Deepest appreciation is also extended to family and friends for their endless encouragement, and to T, whose unwavering support and presence provided the essential strength to complete this research.

References

Aluwala, A. (2021). Real-time performance analytics with Single Pane of Glass dashboards. *International Journal of Science and Research (IJSR)*, *10*(11), 1573–1577. https://doi.org/10.21275/sr24810085027

Aziz, A., & Ambarwati, V. M. (2018). Implementasi sistem monitoring jaringan berbasis Zabbix dan notifikasi alert menggunakan Telegram. *Seminar Nasional Teknik Elektro*, *3*(1), 165–170.

Bayu, P. N. K. (2022). Implementasi server log monitoring system menggunakan Elastic Stack. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *6*(4), 1814–1824.

Damanik, H. A., & Anggraeni, M. (2020). SLA and network availability mechanism implementation scheme for customer service provider. *Jurnal Penelitian Pos dan Informatika*, *10*(2), 125–144. https://doi.org/10.17933/jppi.2020.100204

Fatma, N. F., Ponda, H., & Kuswara, R. A. (2020). Analisis preventive maintenance dengan metode menghitung Mean Time Between Failure (MTBF) dan Mean Time to Repair (MTTR): Studi kasus PT. Gajah Tunggal Tbk. *Heuristic*, *17*(2), 87–94.

Ferreira, L. G. T., José, D. A. M., Cristofolini, A. M., Silva, P. L., & Nascimento, L. de S. (2025). Network monitoring using Zabbix, Grafana, and ZeroTier. *Caderno Pedagógico*, *22*(14), e22081. https://doi.org/10.54033/cadpedv22n14-034

Firdaus, M. I. (2020). *Analisa volume trafik jaringan dan Service Level Agreement (SLA)*.

Hayatu, N., Abayomi, A. A., & Uzoka, A. C. (2024). Advances in SLA monitoring, root cause analysis, and vendor compliance in next-generation networks. *International Journal of Science and Research in Science, Engineering and Technology*, *11*(4), 346–383. https://doi.org/10.32628/ijsrset25121175

Ichsan, M., & Latifah, F. (2025). Press-OK monitoring network menggunakan Zabbix dengan alert notifikasi via Telegram pada PT XYZ. *Journal of Information System, Applied, Management, Accounting and Research*, *9*(4), 1340. https://doi.org/10.52362/jisamar.v9i4.2036

Irianto, D., Yasin, V., & Sianipar, A. Z. (2025). Design and implementation of network and server monitoring using Zabbix at the Financial and Development Supervisory Agency. *Jurnal Teknologi Informatika dan Komputer*, *11*(2), 678–693. https://doi.org/10.37012/jtik.v11i2.2756

Jadhav, S., Kale, P., Uttarwar, A., Bagade, A., & Jatkar, P. (2025). Real-time API observability and anomaly detection in multi-cloud systems. *International Journal of Scientific Development and Research (IJSDR)*, *10*(11), 649–656.

Jailani, M. S. (2023). Teknik pengumpulan data dan instrumen penelitian ilmiah pendidikan pada pendekatan kualitatif dan kuantitatif. *IHSAN: Jurnal Pendidikan Islam*, *1*(2), 1–9. https://doi.org/10.61104/ihsan.v1i2.57

Joseph, A. (2023). Demystifying full-stack observability: Mastering visibility, insight, and action in the modern digital landscape. *International Journal of Computer and Information Engineering*, *17*(8), 485–492.

Karmila, H., & Saptono, H. (2024). Implementasi visualisasi log server dengan ELK Stack: Analisis kasus log akses pada sistem ELENA di Sekolah Tinggi Teknologi Terpadu Nurul Fikri. *Jurnal Informatika Terpadu*, *10*(1), 58–65. https://journal.nurulfikri.ac.id/index.php/JIT

Kurniadi, B., Fitrani, A. S., Rosid, M. A., & Aji, S. (2026). Implementation of a network monitoring system for infrastructure management optimization at the XYZ Office. *Universitas Muhammadiyah Sidoarjo Preprints*.

Kurniawan, N. A., Ruswanti, D., & Charolina, A. (2025). Implementasi Telegram Bot sebagai media monitoring gangguan pelanggan Telkom dengan pembaruan berkala berbasis mobile. *INSOLOGI: Jurnal Sains dan Teknologi*, *4*(5), 1226–1240. https://doi.org/10.55123/insologi.v4i5.6234

Lubis, A., Giffari, M. A., & Wahyuni, S. (2024). Enhancing network performance visibility with Grafana and Prometheus: A case study at PT Nata Digital Solution. *Proceeding of International Conference on Artificial Intelligence, Navigation, Engineering, and Aviation Technology (ICANEAT)*, *1*(1), 304–309.

Malik, P., & Josaphat, B. P. (2024). Design and implementation of network monitoring system using Zabbix and Telegram. *Seminar Nasional Official Statistics*, *2024*(1), 711–722.

Mulyani, A. (2023). Visualisasi data ticketing servicedesk dengan dashboard pada PT Brantas Abipraya (Persero). *Journal of Information System, Applied, Management, Accounting and Research*, *7*(2), 289–300. https://doi.org/10.52362/jisamar.v7i2.1074

Putra, P. H. (2020). Implementasi log management server menggunakan ELK (Elasticsearch, Logstash dan Kibana) Stack pada server web Snort di PT. XYZ. *Jurnal Informatika Sunan Kalijaga*, *4*.

Rahma, A., Indriyani, F., & Sandi, T. A. A. (2023). Perancangan dan implementasi monitoring perangkat server menggunakan Zabbix pada PT. Rizki Tujuh Belas Kelola. *Jurnal INSAN: Journal of Information System Management Innovation*, *3*(2), 85–95. https://doi.org/10.31294/jinsan.v3i2.3009

Rahmawati, A., Halimah, N., Karmawan, K., & Setiawan, A. A. (2024). Optimalisasi teknik wawancara dalam penelitian field research melalui pelatihan berbasis participatory action research. *Jurnal Abdimas Prakasa Dakara*, *4*(2), 135–142. https://doi.org/10.37640/japd.v4i2.2100

Saory, F., & Jaman, J. H. (2025). Sistem monitoring server menggunakan Prometheus dan Grafana dengan integrasi Slack di PT. Atlas Lintas Indonesia. *Jurnal Informatika dan Teknik Elektro Terapan*, *13*(3S1). https://doi.org/10.23960/jitet.v13i3S1.8024

Saputra, M. D. (2025). Desain dan implementasi monitoring infrastruktur perangkat jaringan menggunakan Grafana dan Prometheus. *Jurnal Teknik Informatika dan Sistem Informasi*, *12*(3), 29–47.

Shah, R., & Divecha, N. H. (2025). *Reducing Mean Time to Repair (MTTR) with AIOps: An advanced approach to IT operations management* (Version 1) [Preprint]. https://doi.org/10.21203/rs.3.rs-7383044/v1

Trivaika, E., & Senubekti, M. A. (2022). Perancangan aplikasi pengelola keuangan pribadi berbasis Android. *Nuansa Informatika*, *16*(1), 33–40. https://doi.org/10.25134/nuansa.v16i1.4670

Wahyuni, I. T., Saputri, A. D., Setiawan, I., & Putranto, B. D. (2025). Analisis manajemen layanan TI pada perusahaan penyedia layanan internet menggunakan ITIL V4 Service Value System. *Mars: Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer*, *3*(6), 71–91. https://doi.org/10.61132/mars.v3i6.1224

Wibowo, K., Sugiyarto, S., & Setiono, S. (2018). Analisa dan evaluasi: Akar penyebab dan biaya sisa material konstruksi proyek pembangunan kantor kelurahan di Kota Solo, sekolah, dan pasar menggunakan Root Cause Analysis (RCA) dan Fault Tree Analysis (FTA). *Matriks Teknik Sipil*, *6*(2). https://doi.org/10.20961/mateksi.v6i2.36572

Widyaningrum, B., Helbawanti, O., Triyanto, S. A., Nuraini, C., & Mutolib, A. (2023). Permodelan Service Level Agreement (SLA): Upaya mendukung kelulusan tepat waktu mahasiswa di perguruan tinggi. *Prosiding SEMDIKJAR (Seminar Nasional Pendidikan dan Pembelajaran)*, *6*, 648–661. https://doi.org/10.29407/2z22p813

Yerram, S. (2025). Kibana deployment in AWS for log analysis: A research-based study on observability and Elastic Stack integration. *IJSAT: International Journal on Science and Technology*, *16*(3).