Original Article

e-ISSN: 2775-5037
p-ISSN: 2614-3917

Original Article

Log Analyzer for Operational Data Monitoring and Visualization

Apipah Yulia Marsis[1]✉, Widi Hastomo[2]
[1,2]Department of Information Technology, Ahmad Dahlan Institute of Technology and Business, Indonesia
Korespondensi Email: apipahyuma@gmail.com✉

Abstract:

Monitoring and visualization of operational data play a crucial role in supporting data-driven system management. A common challenge encountered is that log data from various systems are not integrated into a single central repository, which then impedes monitoring efforts. This study outlined the use of a log analyzer tool to facilitate the monitoring and visualization of operational data through an integrated dashboard. The applied methodology includes log data acquisition, data transformation into a structured database schema, and information representation in a visual format. The developed system provided the capability to observe real-time operational conditions and system performance metrics. Research findings indicate that the implementation of a log analyzer contributed to improved data understanding and increased monitoring process effectiveness. Overall, the results of this study conclude that log analyzer can be a supporting tool in the process of data-driven decision-making in the operational environment of an institution.

Keywords: Log Analyzer, System Monitoring, Operational Data, Data Visualization, Integrated Dashboard

Submited	: 10 January 2026
Revised	: 31 January 2026
Acceptance	: 8 February 2026
Publish Online	: 9 February 2026

Introduction

Operational system monitoring is a crucial aspect of managing information technology infrastructure because systems generate large volumes of operational data that needs to be analyzed to support data-driven decision-making [1][2]. System logs are the primary source of operational data as they record system activities and internal conditions, but their potential for monitoring is often not optimal [3][4]. Problems arise when log data is scattered across various system components and not integrated, making it difficult to monitor the overall system conditions in real-time [5][6]. Various studies show that log analysis can improve system observability and assist in detecting operational anomalies [7][8]. To support understanding, log data is generally presented through visual monitoring dashboards, although the analysis and visualization processes are still often separated [9-12] . Therefore, integrating log analyzers and monitoring dashboards is necessary to improve monitoring efficiency and data-driven decision-making [13-15].

Modern operational system monitoring largely relies on log and event data continuously generated by information systems [16][17]. Log data is used to represent system activities and system states through pattern extraction and modeling processes [18][19]. Process mining approaches are utilized to abstract event logs, enabling systematic understanding of workflows and system behavior [20][21]. In large-scale systems, monitoring is generally implemented in a distributed manner to enhance the scalability and reliability of log data processing [22]. Literature indicates that operational data in modern systems is characterized by high volume and diversity, thereby requiring efficient and structured processing mechanisms [23-25]. The results of log analysis are then presented through unified dashboards and visualization platforms to support continuous operational monitoring [26-28]. However, the increasing complexity of systems and log data poses interpretation challenges, necessitating adaptive and transparent analysis approaches [29][30].

The development of monitoring systems drives the adaption of an integrated approach that combines system logs, event records, and various data sources to support real-time system condition monitoring [31-33]. The quality and reliability of log recording play a crucial role in system observability and the effectiveness of operational analysis and decision-making [34-37]. In addition to the analytical aspects, the literature also emphasizes the importance of security mechanisms in log management and analysis to maintain data integrity and the reliability of system monitoring processes [38]. To manage data complexity, data-driven methods supported by artificial intelligence are widely applied for pattern identification and system deviation detection [39][40]. The literature also indicates that modern monitoring systems can involve various supporting technologies to enhance system responsiveness and decision-making efficiency [41-45]. Furthermore, data visualization and dashboards play a crucial role in translating technical data into information that is easily understood by users [46][47]. In general, integrated data-based monitoring systems contribute to improved operational effectiveness, asset management, and risk mitigation through continuous analysis and early warning mechanisms [48-55].

Research Method

This section outlined the research approach applied, covering the study design, system framework, information collection procedures, data processing methods, visual dashboard development, algorithm design, and system validation and performance assessment stages. This approach is designed to facilitate the development and validation of a data monitoring system that integrates system indicators with operational information.

Research Design

This research was conducted using an implementation case study research methodology, with an emphasis on developing a data monitoring platform as a practical tool to facilitate supervision of the system's operational status [56]. This monitoring platform is built based on the evaluation of log data and parameters; operational data was collected, processed, and archived in a structured repository to support continuous observation and evaluation of the system's [57].

The output from the data processing is then displayed through a visual dashboard to convey operational information concisely and optimize data-driven decision-making [58]. The research design establishes a comprehensive workflow, starting from data acquisition, archiving in a structured database, to information representation through an instrument panel as an integrated monitoring system unit. The research process includes several stages:

- Identification of monitoring needs.
- Construction of system architecture and data models.
- Development of data acquisition and retention modules.
- System alignment and control panel arrangement.
- System validation to verify operationality and data consistency.

System Architecture and Environment

The system architecture developed in this research is structured as a modular monitoring pipelines. This integrates data collection, data archiving, and visual data presentation stages into an organized workflow. The objective is to facilitate the simultaneous monitoring of system conditions. This method allows each part of the system to operate independently while also integrating as a cohesive monitoring architecture unit, in line with the principles of monitoring system architecture that relies on data visualization [59].

This system is implemented using several components. PostgreSQL functions as a relational database that manages structured operational data storage. Prometheus and Node Exporter are chosen to collect various system performance metrics. Meanwhile, Grafana is utilized as a means of data visualization through the presentation of monitoring dashboards. architecture The clear division of functions between these components is designed to ensure orderly data management and simplify system monitoring and analysis activities. Prometheus is configured to periodically collect system metrics provided by Node Exporter, while PostgreSQL is utilized to store operational data generated during monitoring activities. Information from both these sources is subsequently accessed by Grafana and displayed in a unified monitoring dashboard. This allows for a concise and easily understandable presentation of system status, in line with the function of an interactive dashboard as a real-time system condition monitoring interface [60].

All parts of this system are operated within a virtualization environment. The primary goals of this step are to isolate the system, maintain configuration uniformity among different components, and simplify the deployment, administration and evaluation processes of the monitoring system being developed.

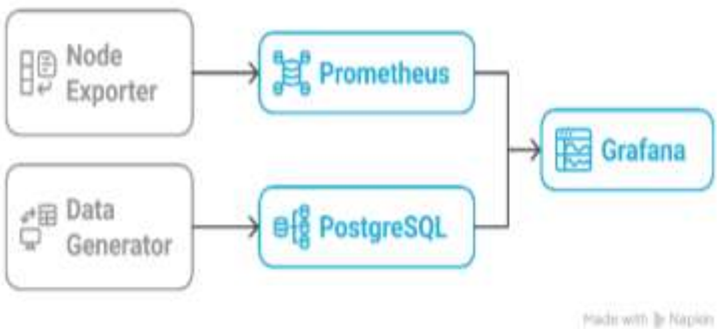

Figure 1. System architecture of the data monitoring platform

Data Acquisition

Data collection in this study originates from two primary sources: system indicators and operational information. The goal is to obtain a holistic perception of the system's state, encompassing both infrastructure performance aspects and the data dynamics occurring within it. This strategy supports a more integrated system observation by simultaneously integrating technical and operational dimensions.

System indicator information is gathered through Node Exporter, which provides data on server performance, such as CPU utilization, memory, disk, and network. This data is then collected periodically by Prometheus, a monitoring platform focused on time-series data. Additionally, the study also integrates operational information generated by the data generator system. This information reflects operational activities and is stored in a database as complementary monitoring elements. Information from both sources is then applied to the data processing and visualization phase to support the continuous system condition monitoring mechanism.

Data Processing and Extract, Transform, Load (ETL)

The data processing in this research adopts the Extract, Transform, Load (ETL) methodology to prepare acquired data for use in the monitoring and visualization system. The ETL process is applied to two main data categories: system metric data and operational data, with the aim of ensuring data structure uniformity and facilitating interoperability between system components.

In the extract phase, system metric data is obtained from Prometheus, which handles time-series data from the Node Exporter collection. This data reflects the server's performance status over specific time spans. Additionally, operational data is extracted through a data generator mechanism that produces structured information to describe the system's operational activities.

The transformation phase is carried out to adjust the data format and structure to align with the storage schema and visualization requirements. Operational data will be structured into a relational format with consistently defined attributes, while metric data is prepared to support time-based aggregation and monitoring processes, thus enabling integrated utilization of data from both sources.

In the load phase, the data that has undergone the transformation process is then stored into the relevant storage system. A relational database is used to store operational data, while a time-series storage system is utilized for metric data. The loaded data can then be accessed by the visualization layer to support the integrated and continuous presentation of monitoring information. To provide a clearer overview of the data processing flow applied, a summary of the stages in the ETL process is presented in Table

Table 1. Summary of Data ETL Process Stages

Stage	Process	Description
Extract	Data Retrieval	System metric data is extracted from Prometheus and operational data is obtained from the data generator.
Transform	Data	Data is structured to conform to the

	Structuring	storage schema and validated to ensure consistency across system components.
Load	Data Storage	Processed data is stored in the appropriate database system.
Output	Data Ready for Visualization	Data is available for use in the monitoring dashboard.

Dashboard Procedure

This phase outlines procedures for presenting monitoring results in a visual dashboard format. This dashboard serves as the primary means for concisely and clearly communicating system status, encompassing both system metrics and operational information. Information is presented in an integrated manner to facilitate continuous system status oversight. In this regard, the dashboard acts as an interactive visual interface that supports system status observation and data-driven decision-making [61].

Grafana was chosen as the visualization software due to its capabilities in processing and displaying data from multiple sources. In the initial phase, data source configuration was performed by integrating Grafana with Prometheus for system metric access, and with PostgreSQL for retrieving structured operational data. Thus, two different types of data can be consolidated in one visualization environment. The dashboard is composed of several fundamental sections. The section measuring system metrics contains server performance indicators, including CPU, memory, disk, and network utilization. Meanwhile, the operational data section presents an overview of transactions, data distribution, and system activity patterns. Data is updated periodically based on the retrieval frequency from each source, so the main screen automatically displays the current system status. Visualization through monitoring dashboards facilitates a deeper observation of the system status and tracked operational data. To differentiate the types of data displayed, the visualization is categorized into two main views: a dashboard that presents system performance indicators, and an operational data dashboard that presents several main views including Executive, Detail, Regional, and Sales. All were developed using the Grafana platform to display real-time sales analysis.

Algorithm and Pseudocode

The algorithm is designed to manage system processes in handling data, covering the stages of acquisition, processing, storage, and presentation of information on a monitoring dashboard. Broadly, the monitoring system involves two fundamental algorithmic paths: algorithms for system metric processing and algorithms for operational data processing. These two paths are formulated to operate independently, distinguishing the characteristics of their respective data, but are connected at the visualization phase to facilitate comprehensive monitoring of system conditions.

To concretely elaborate the logical sequence of the monitoring system, this research presents pseudocode that serves as a structured overview of the crucial stages in the process. The pseudocode is presented for two main paths, namely system metric monitoring and operational data monitoring. Each pseudocode specifically outlines the sequence in terms of data acquisition, data transformation, archiving, and pre-visualization stages.

Table 2. Pseudocode system metric monitoring

Input	Process	Output
• Server/VM metrics endpoint from Node Exporter (:9100) • Prometheus scrape configuration (prometheus.yml) • Grafana datasource configuration (Prometheus)	1. Start Node Exporter on the monitored server. 2. Node Exporter exposes system metrics, including CPU, memory, disk, and network usage. 3. Prometheus periodically scrapes metrics from the Node Exporter endpoint. 4. Collected metrics are stored as time-series data in Prometheus. 5. Grafana connects to Prometheus as a data source. 6. Grafana queries system metrics through dashboard panels. 7. Dashboard panels are refreshed based on the configured refresh interval.	Real-time system metrics visualization on Grafana dashboard.

Table 3. Pseudocode operational data monitoring

Input	Process	Output
• Operational data generation mechanism • PostgreSQL database configuration • Grafana data source configuration (PostgreSQL)	1. Execute the data generator to produce operational data records. 2. Validate the generated data to ensure structural consistency. 3. Establish a connection to the PostgreSQL database. 4. Insert operational data into the corresponding database tables. 5. Grafana connects to PostgreSQL as a data source. 6. Grafana queries operational data using predefined queries. 7. Dashboard panels are updated periodically based on the refresh interval.	Visualization of operational data on the monitoring dashboard.

Testing and Evaluation

The testing and assessment phase is carried out to ensure that the monitoring system that has been developed can operate according to the research objectives. Testing focuses on system functionality, data consistency, and the stability of the overall monitoring process. This assessment aims to measure the system's readiness in supporting continuous monitoring of system conditions. Functional testing is carried out by verifying the data flow from each crucial element of the system. Node Exporter is tested to verify that system metrics are successfully exposed and collected by Prometheus, while operational data generated by the data generator module is tested to ensure it can be accurately stored in the PostgreSQL database. Furthermore, Grafana is tested to confirm its ability to access both data sources and present monitoring information according to the designed dashboard.

In addition, the data verification process is carried out by comparing the information collected in the database and monitoring system with the data shown on the dashboard. This activity aims to ensure there are no discrepancies in values or data loss in the information acquisition and presentation cycle. An observation of basic system performance is also conducted by focusing on the regular data update periods on the dashboard, as well as the system's operational stability during the monitoring phase. Findings from the testing indicate that the system is capable of functioning stably and facilitates monitoring capabilities in accordance with the specified requirements.

Results

This section describes the result of the monitoring system implementation and data visualization as as described in the methodology. These results are obtained through the integration of system metric pipelines and operational information into a Grafana dashboard. This allows for an integrated view of performance and operational activities. In addition to presenting the implementation results and dashboard, this section also includes an assessment of the system's functional performance and an analysis of crucial findings to evaluate the effectiveness of the developed data-driven monitoring method.

Results of System Implementation (Monitoring and ETL)

In accordance with the implementation architecture diagram in Figure 1, system monitoring development involves two main flows: the system metrics flow and the operational data flow. These two flows are then integrated at the visualization level using the Grafana platform. The system metrics flow utilizes Node Exporter to expose server performance parameters such as CPU, memory, disk, and network. Subsequently, Prometheus periodically collects (scrapes) this data and stores it in a time-series database for visualization on the performance dashboard.

Meanwhile, the operational data flow is implemented through a data generator module that produces transaction data in a structured format. This data is then stored in a PostgreSQL database and serves as the primary source for visualizing operational indicators in Grafana. Thus, the monitoring covers not only infrastructure status but also operational activities comprehensively within a unified interface. A summary of the implementation results of each element along with its success metrics is detailed in Table 4. This table indicates that the essential components have functioned as intended and have been effectively integrated in the visualization phase.

Table 4. Summary of monitoring system implementation results

Component	Function	Implementation Status
Node Exporter	Collects system metrics	Successfully running and connected to Prometheus
Prometheus	Stores metric data	Time-series data has been successfully stored and visualized
Data Generator	Produces transaction data	Successfully generated data according to the defined schema
PostgreSQL	Stores operational data	Data were successfully stored and are accessible by Grafana
Grafana	Displays an integrated dashboard	Dashboard successfully display real-time metric and operational data

Through the achievement of integration between these two flows, the subsequent discussion will focus on the visual representation of the dashboard and the interpretation of the metrics presented in each dashboard view in the following section.

Results Dashboard

The monitoring system was implemented through the Grafana interface, which is divided into two data classifications: system metrics and operational data. This integrated design aims to simplify interpretation, allowing users to evaluate infrastructure conditions and operational activities within a single review context. By utilizing this interface, supervision is carried out in a consolidated manner, presenting more intuitive visualizations compared to the original data sources. The created graphical interface includes the Metrics Dashboard (Local) along with a series of operational dashboards, consisting of four visualization sections: Executive, Detail, Regional, and Sales.

a. System Metrics Dashboard

The system metrics dashboard is utilized to directly monitor server performance, focusing on CPU, memory, disk, and network parameters. The primary objective is to identify deviations or unusual load increases more promptly. In the context of the internship report, the implementation of this dashboard presented through the Local Metrics Dashboard. Figure 2 presenting a Local System Metrics Dashboard implemented using Grafana. This dashboard provides real-time visualization of key performance indicators to monitor system behavior.

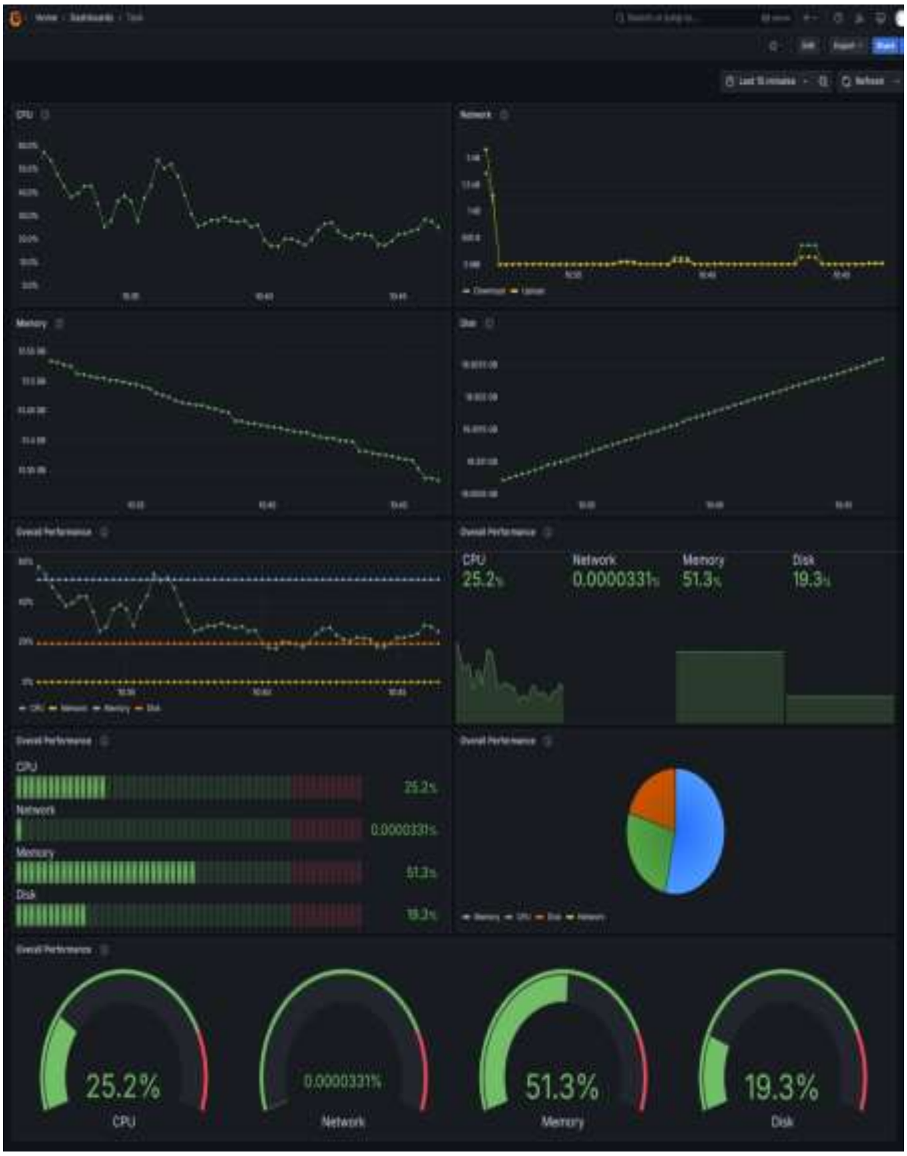

Figure 2. Local System Metrics Dashboard implemented using Grafana

b. Operational Data Dashboard

The operational data display presents a visual representation of transaction information stored in the database, which is then processed through specific requests to be reflected in Grafana. The design of this visualization aims to facilitate users in monitoring operational activities from a single integrated location, covering performance summaries to detailed breakdowns categorized by geographical area and time range. Broadly, the operational data display is divided into four main sections: Executive Dashboard, Detail Dashboard, Regional Dashboard, and Sales Dashboard.

Executive Dashboard displays a summary of key metrics (e.g., total number of transactions/sales and their trends) to provide a brief overview of the operational status. Figure 3 presenting an Executive Operational Data Dashboard implemented using Grafana. This dashboard provides a comprehensive performance summary visualization to facilitate users in real-time monitoring of overall operational status.

Figure 3. Executive Operational Data Dashboard implemented using Grafana

Detail Dashboard provides an in-depth view for detailed analysis, such as comparing performance between units and presenting a ranked list (best/worst) as needed for analysis. Figure 4 presenting a Detail Operational Data Dashboard implemented using Grafana. This dashboard enables in-depth monitoring of operational data to help users analyze data performance more effectively.

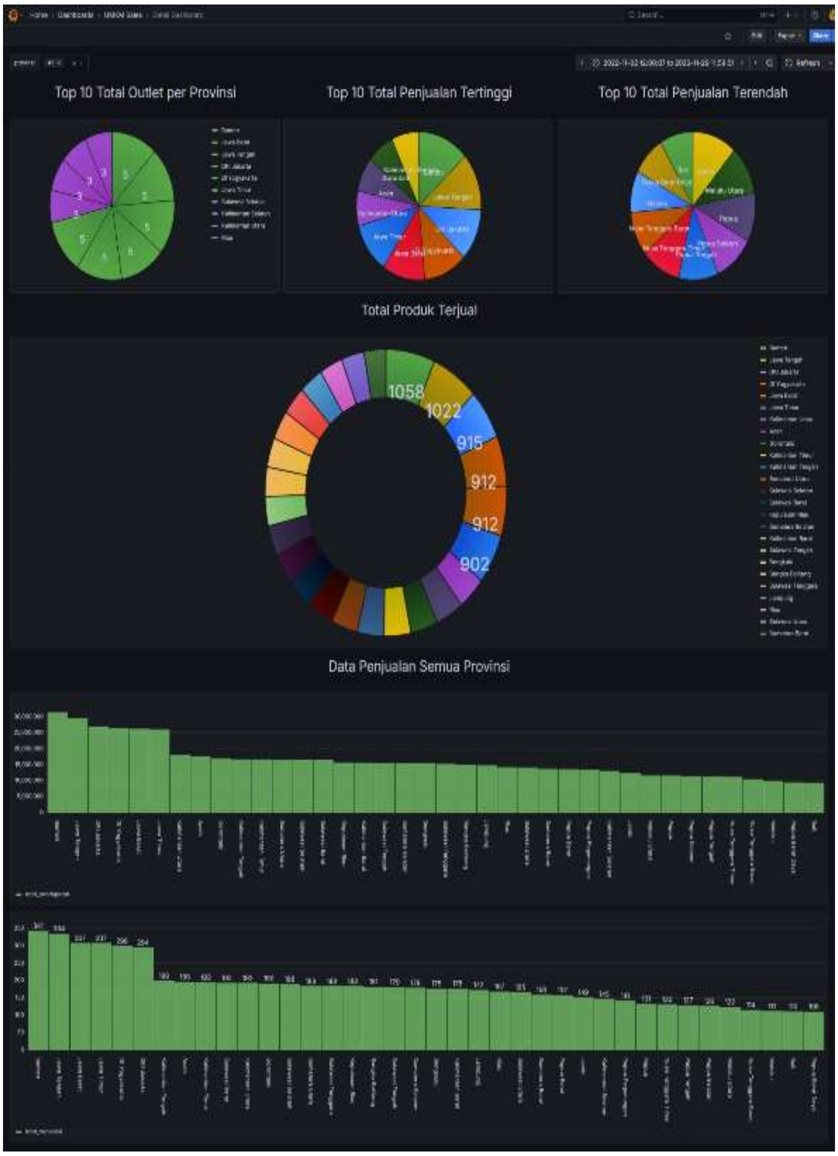

Figure 4. Detail Operational Data Dashboard implemented using Grafana

Regional Dashboard shows the distribution and performance comparison based on geographical location to identify areas with the highest and lowest activity levels. Figure 5 presenting a Regional Operational Data Dashboard implemented using Grafana. This dashboard visualizes geographic distribution and regional performance comparisons to help users understand performance differences between regions.

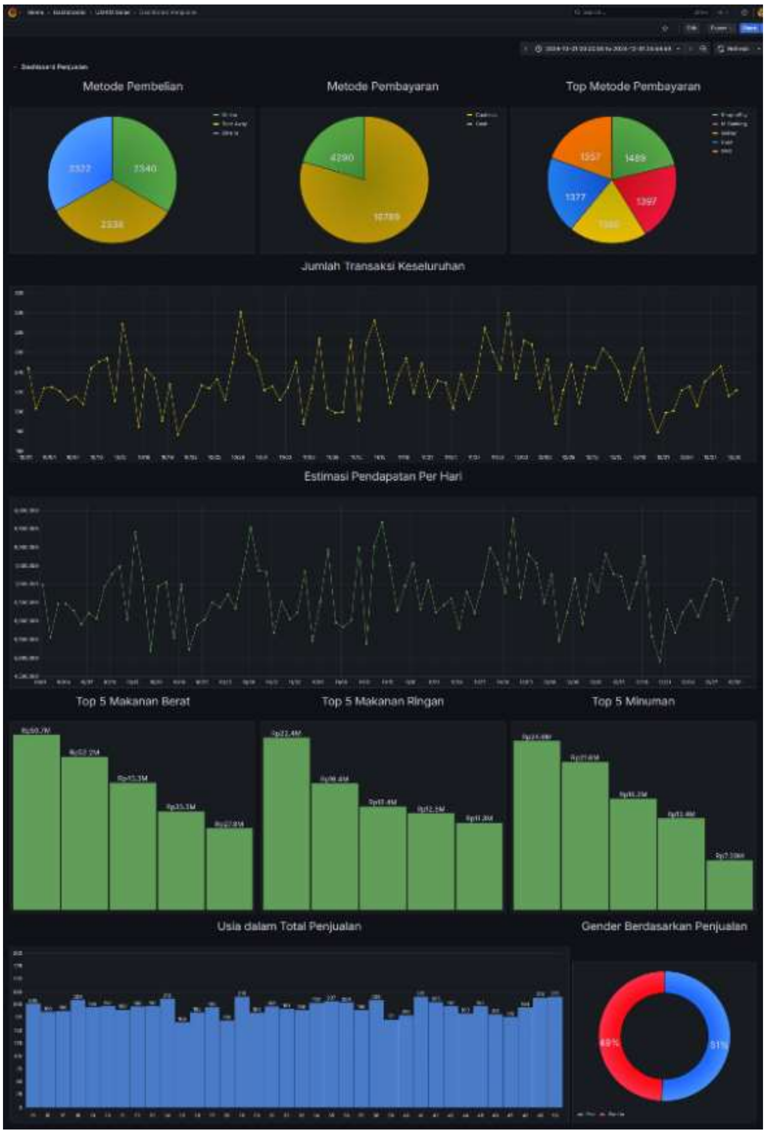

Figure 5. Regional Operational Data Dashboard implemented using Grafana

Sales Dashboard focuses on monitoring sales trends based on time ranges that allow for clearer observation of changes in sales patterns. Figure 6 presenting a Sales Operational Data Dashboard implemented using Grafana. This dashboard displays sales data visualizations based on various key aspects related to consumer behavior and product performance.

Figure 6. Sales Operational Data Dashboard implemented using Grafana

Through the integrated dashboard, users can efficiently monitor and identify shifts in system performance patterns or operational activities based on the presented trend charts.

3.3 Results of System Testing and Evaluation

Testing was conducted to verify that the developed monitoring system functioned as designed, encompassing both system metric pipeline and operational data pipeline, and to confirm consistent integration within Grafana. The testing in this section concentrated on functional aspects, specifically the interconnections between components, successful data acquisition and storage, and data readability on the dashboard. Test Case (TC) codes, such as TC-01, were used to each test case. A summary of the functional testing results for each component is presented in Table 5, while a summary of the implementation evaluation, based on the system's primary aspects, is shown in Table 6.

Table 5. Functional Testing Results of the Monitoring System

Test ID (TC)	Scenario/ Component	Test Steps (Brief)	Expected Result	Actual Result (Brief)	Status
TC-01	Node Exporter active	Access metrics endpoint on host	Endpoint responds and metrics are available	Endpoint OK, metrics appear	Pass
TC-02	Prometheus scraping	Check targets page	Target detected and status is UP	Target UP	Pass

	target				
TC-03	Metric data stored	Run a simple metric query	Data appears according to time range	Time-series data displayed	Pass
TC-04	Generator produces data	Run generator module	Transaction data is formed	Data formed according to schema	Pass
TC-05	Data stored in database	Check table (sample query/row)	Data stored without error	Data is queryable	Pass
TC-06	Prometheus Datasource in Grafana	Test datasource connection	Connection successful	Connection OK	Pass
TC-07	Grafana operational DB Datasource	Test datasource connection	Connection successful	Connection OK	Pass
TC-08	Executive Dashboard	Open dashboard and check main panel	Panel rendered and populated	Panel displayed normally	Pass
TC-09	Detail Dashboard	Open dashboard and check tables/rangkings	Detail Panel displayed and readable	Panel displayed normally	Pass
TC-10	Regional Dashboard	Open dashboard and check region panel	Region panel displayed according to query	Panel displayed normally	Pass
TC-11	Sales Dashboard	Open dashboard and check sales trends	Trends displayed according to period	Trends displayed normally	Pass
TC-12	Dashboard data update	Observe panel refresh	Values change following update	Refresh executed	Pass

Table 6. Summary of Implementation Evaluation Results

Evaluation Aspects	Indicators	Success Criteria	Results
Component Connectivity	Prometheus Target Status	Target Detected and UP	Fulfilled
Data Source Integration	Grafana Datasource Connection	Prometheus & DB Connected	Fulfilled

Dashboard Readability	Panel Rendered	Panel Displays Without Error	Fulfilled
Data Updates	Refresh/Update Interval	Consistent Updates	Fulfilled
Operational Data Consistency	Panel Value vs. Query Result	Consistent Values	Fulfilled
Display Stability	Panel Errors/Timeouts	No Errors During Testing	Fulfilled

Based on the findings presented in Table 5, it can be concluded that all crucial elements in the system metrics pipeline and operational data pipeline function as intended and are well-integrated, achieving the visualization stage on the Grafana platform. Further evaluation outlined in Table 6 indicates that the data source integration process, dashboard display clarity, and data update frequency exhibit consistent performance. This enables the dashboard to serve as a central interface for comprehensively monitoring system status and operational activities. Therefore, the developed monitoring system has successfully met the basic functional requirements for oversight and the presentation of information relevant to the implementation objectives.

Discussion

Based on the dashboard visualization output and functional verification, the monitoring system that has been built indicates that the integration of system metric data with operational data can be used for integrated supervision through Grafana. This integration process provides convenience for users because the observation of infrastructure status and operational activities can be carried out within a single analysis framework, leading to a more organized interpretation of system conditions.

In terms of system metrics, the Metrics Dashboard (Local) facilitates the observation of CPU, memory, disk, and network with trend representations so that performance fluctuations can be identified more efficiently. For operational data, the Executive Dashboard presents an overview of conditions, the Detail Dashboard supports in-depth analysis through performance comparison between units, the Regional Dashboard aids in understanding disparities based on geography, and the Sales Dashboard simplifies the monitoring of sales patterns according to time ranges. Overall, the test findings indicate that the core system components function as intended up to the visualization phase; however, further evaluation development is still possible through the establishment of quantitative measures (such as data update latency and load testing) and the integration of an early warning system (alerting).

Conclusion

Based on the problem statement established, this research successfully achieved its objective of integrating operational data monitoring and visualization. This was accomplished through the utilization of tools such as log analyzers and Grafana dashboards. The system was built by designing two main process pipelines: first, system metric monitoring using Node Exporter and Prometheus; second, operational data monitoring with a data generator and PostgreSQL. Both pipelines were then integrated in the visualization stage using Grafana. As a result, information about the infrastructure's condition, including CPU, memory, disk, and network, along with operational activities, can be observed on a single integrated platform. This display presents a more concise and informative analysis, ultimately supporting data-driven decision-making in the operational environment.

Functional testing results confirmed that all essential components operated according to specifications up to the visualization phase. This included interconnections between components, data source integration, clarity of panel displays, and uniformity of data updates on the dashboard. In the next development phase, the system can be enhanced with the implementation of an early warning system (alerting) and quantitative analysis, such as measuring data update latency and load testing. This step will strengthen the measurable proof of monitoring effectiveness and prepare the system for deployment in more demanding operational scenarios.

References

[1] S. Kauffman, "Science of Computer Programming Log analysis and system monitoring with nfer," *Sci. Comput. Program.*, vol. 225, p. 102909, 2023, doi: 10.1016/j.scico.2022.102909.

[2] R. Nai *et al.*, "Computer Law & Security Review : The International Journal of Technology Law and Practice Leveraging process mining and event log enrichment in European public procurement analysis : a case study," *Comput. Law Secur. Rev. Int. J. Technol. Law Pract.*, vol. 57, no. May, p. 106144, 2025, doi: 10.1016/j.clsr.2025.106144.

[3] M. De, C. Cabello, T. Prince, and M. R. Machado, "Intelligent Systems with Applications AIOps for log anomaly detection in the era of LLMs : A systematic literature review," *Intell. Syst. with Appl.*, vol. 28, no. July, p. 200608, 2025, doi: 10.1016/j.iswa.2025.200608.

[4] M. Wurzenberger, G. Höld, M. Landauer, and F. Skopik, "Computers & Security Analysis of statistical properties of variables in log data for advanced anomaly detection in cyber security," *Comput. Secur.*, vol. 137, no. November 2023, p. 103631, 2024, doi: 10.1016/j.cose.2023.103631.

[5] K. Aida, "The Log Analysis Environment to Supoort Classroom Using CoursewareHub," *Procedia Comput. Sci.*, vol. 270, pp. 3431–3439, 2025, doi: 10.1016/j.procs.2025.09.468.

[6] L. Gurina and L. Branch, "ScienceDirect Cyber resilience models of systems for monitoring and operational dispatch control of electric power systems," *IFAC Pap.*, vol. 55, no. 9, pp. 485–490, 2022, doi: 10.1016/j.ifacol.2022.07.084.

[7] J. Kim, H. Lee, S. Jeong, and S. Ahn, "Sound-based remote real-time multi-device operational monitoring system using a Convolutional Neural Network (CNN)," *J. Manuf. Syst.*, vol. 58, no. PA, pp. 431–441, 2021, doi: 10.1016/j.jmsy.2020.12.020.

[8] M. Kumar, A. Kumar, K. Joshi, and A. Sumanth, "Revolutionizing hybrid additive manufacturing : the impact of digital shadow-driven smart dashboard and augmented reality on operational efficiency," *Manuf. Lett.*, vol. 44, pp. 1405–1414, 2025, doi: 10.1016/j.mfglet.2025.06.160.

[9] M. Karami, N. Hafizi, and A. Nickfarjam, "Heliyon Development of minimum data set and dashboard for monitoring adverse events in radiology departments," *Heliyon*, vol. 10, no. 9, p. e30054, 2024, doi: 10.1016/j.heliyon.2024.e30054.

[10] D. Lyon *et al.*, "Dashboard proposition for health monitoring of production system in the automotive industry," *IFAC Pap.*, vol. 54, no. 1, pp. 780–786, 2021, doi: 10.1016/j.ifacol.2021.08.091.

[11] A. Sorour and A. S. Atkins, "Journal of Electronic Science and Technology Big data challenge for monitoring quality in higher education institutions using business intelligence dashboards," *J. Electron. Sci. Technol.*, vol. 22, no. 1, p. 100233, 2024, doi: 10.1016/j.jnlest.2024.100233.

[12] H. Saputra, R. Alayham, A. Helmi, M. D. Ghazali, and W. O. Sumartini, “Jo ur na l P re r f,” *Nat. Hazards Res.*, 2025, doi: 10.1016/j.nhres.2025.09.005.
[13] A. Ur, I. G. Muhammad, H. M. Khalid, Z. Said, A. Iqbal, and S. M. Muyeen, “Techno economics and energy dynamics of a solar powered smart charging infrastructure for electric vehicles with advanced IoT based monitoring and RFID based security,” *Sustain. Futur.*, vol. 8, no. November, p. 100376, 2024, doi: 10.1016/j.sftr.2024.100376.
[14] A. Olkhovikov *et al.*, “Decentralized Oil Extraction Infrastructure via Blockchain-Enabled IoT Networks,” *Blockchain Res. Appl.*, p. 100425, 2025, doi: 10.1016/j.bcra.2025.100425.
[15] J. Ribeiro, P. Lima, and F. Nunes, “SoftwareX Trial Monitor : Scaffolding personalised Web dashboards for Human – Computer Interaction field trials,” *SoftwareX*, vol. 16, p. 100883, 2021, doi: 10.1016/j.softx.2021.100883.
[16] G. Park and W. M. P. Van Der Aalst, “Computers in Industry Operational process monitoring: Anobject-centric approach,” *Comput. Ind.*, vol. 164, no. August 2024, p. 104170, 2025, doi: 10.1016/j.compind.2024.104170.
[17] M. Mayr, S. Luftensteiner, and G. C. Chasparis, “ScienceDirect Abstracting Process Mining Event Logs From Process-State Data To Monitor Control-Flow Of Industrial Manufacturing Processes,” *Procedia Comput. Sci.*, vol. 200, pp. 1442–1450, 2022, doi: 10.1016/j.procs.2022.01.345.
[18] Z. Xu, W. Wang, C. Liu, and X. Hu, “LogERT : Stable log template mining method based on evolving re-search trees,” *Array*, vol. 28, no. October, p. 100543, 2025, doi: 10.1016/j.array.2025.100543.
[19] P. L. Foalem, F. Khomh, and H. Li, “Studying logging practice in machine learning-based applications,” *Inf. Softw. Technol.*, vol. 170, no. July 2023, p. 107450, 2024, doi: 10.1016/j.infsof.2024.107450.
[20] R. Falach *et al.*, “SleepEEGpy : a Python-based software integration package to organize preprocessing , analysis , and visualization of sleep EEG data,” *Comput. Biol. Med.*, vol. 192, no. PA, p. 110232, 2025, doi: 10.1016/j.compbiomed.2025.110232.
[21] S. He and Y. Cui, “Computers & Education A systematic review of the use of log-based process data in computer-based assessments,” *Comput. Educ.*, vol. 228, no. June 2023, p. 105245, 2025, doi: 10.1016/j.compedu.2025.105245.
[22] S. Huang, Y. Li, and J. Wu, “Distributed state estimation for linear time- invariant dynamical systems : A review of theories and algorithms,” *Chinese J. Aeronaut.*, vol. 35, no. 6, pp. 1–17, 2022, doi: 10.1016/j.cja.2021.06.010.
[23] E. Lodhi, X. Liu, G. Xiong, M. A. Khan, and Z. Lodhi, “Energy Conversion and Management : X SmartPV-AIoT : an AIoT-integrated framework for fault diagnosis and remote monitoring in photovoltaic systems,” *Energy Convers. Manag. X*, vol. 27, no. March, p. 101117, 2025, doi: 10.1016/j.ecmx.2025.101117.
[24] A. H. Awad *et al.*, “Heliyon Low-cost IoT-Based sensors dashboard for monitoring the state of health of mobile harbor cranes : Hardware and software description,” *Heliyon*, vol. 10, no. 22, p. e40239, 2024, doi: 10.1016/j.heliyon.2024.e40239.
[25] A. Alshuhail, A. Alshahrani, H. Mahgoub, and M. Ghaleb, “Machine edge-aware IoT framework for real-time health monitoring : Sensor fusion and AI-driven emergency response in decentralized networks,” *Alexandria Eng. J.*, vol. 129, no. May, pp. 1349–1361, 2025, doi: 10.1016/j.aej.2025.08.030.
[26] C. Kelly, J. Fawkes, R. Habermehl, D. De Ferreyro, and N. Zimmerman, “PLUME Dashboard : A free and open-source mobile air quality monitoring dashboard,” *Environ. Model. Softw.*, vol. 160, no. November 2022, p. 105600, 2023, doi: 10.1016/j.envsoft.2022.105600.
[27] P. Wijayakusuma, G. Persada, N. Hakim, and B. Li, “HardwareX TerraGrow : Integrated platform for real time plant monitoring and automated watering system with IoT and fuzzy Sugeno Algorithm,” *HardwareX*, vol. 24, no.

November, p. e00724, 2025, doi: 10.1016/j.ohx.2025.e00724.
[28] D. Cota, "Journal of Open Innovation : Technology , Market , and Complexity BHiveSense : An integrated information system architecture for sustainable remote monitoring and management of apiaries based on IoT and microservices," vol. 9, no. July, 2023, doi: 10.1016/j.joitmc.2023.100110.
[29] V. Zieglmeier, "The Inverse Transparency Toolchain : A Fully Integrated and Quickly Deployable Data Usage Logging Infrastructure," *Softw. Impacts*, vol. 17, no. July, p. 100554, 2023, doi: 10.1016/j.simpa.2023.100554.
[30] L. Das, B. Gjorgiev, and G. Sansavini, "Uncertainty-aware deep learning for monitoring and fault diagnosis from synthetic data," *Reliab. Eng. Syst. Saf.*, vol. 251, no. July, p. 110386, 2024, doi: 10.1016/j.ress.2024.110386.
[31] H. Miller, J. Shrestha, O. Lefebvre, and N. Maccarty, "Use of an integrated suite of sensors to simultaneously monitor fuel consumption , air quality , and adoption provides important insights and validates impact metrics for household stoves," *Dev. Eng.*, vol. 7, no. September, p. 100099, 2022, doi: 10.1016/j.deveng.2022.100099.
[32] J. Wang, K. Mizuno, S. Tabeta, T. Matsuoka, and T. Odake, "Multi-dataset-integrated Coral-Lab segmentation with enhanced towed camera array for rapid large-scale coral reef monitoring and mapping," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 143, no. June, p. 104819, 2025, doi: 10.1016/j.jag.2025.104819.
[33] A. A. Sukmandhani and M. Zarlis, "Monitoring Applications for Vehicle based on Internet of Things Monitoring Applications for Vehicle based on Internet of Things (IoT) using the MQTT Protocol (IoT) using the MQTT Protocol," *Procedia Comput. Sci.*, vol. 227, pp. 73–82, 2023, doi: 10.1016/j.procs.2023.10.504.
[34] M. Kern, M. Landauer, F. Skopik, and E. Weippl, "Computers & Security A logging maturity and decision model for the selection of intrusion detection cyber security solutions," *Comput. Secur.*, vol. 141, no. December 2023, p. 103844, 2024, doi: 10.1016/j.cose.2024.103844.
[35] S. Kroeger, M. Wegmann, P. Ehmke, and M. F. Zaeh, "Extracting Event Logs from Value Stream Simulation in Production Networks for Data Farming Based Strategic Network Design," *Procedia CIRP*, vol. 130, pp. 1282–1289, 2024, doi: 10.1016/j.procir.2024.10.240.
[36] A. Mastropaolo, V. Ferrari, L. Pascarella, and G. Bavota, "Log statements generation via deep learning : Widening the support provided to developers," *J. Syst. Softw.*, vol. 210, no. November 2023, p. 111947, 2024, doi: 10.1016/j.jss.2023.111947.
[37] P. Jin, N. Kim, and D. Jeong, "Forensic Science International : Digital Investigation Enhancing DFIR in orchestration Environments : Real-time forensic framework with eBPF for windows," *Forensic Sci. Int. Digit. Investig.*, vol. 53, no. S, p. 301923, 2025, doi: 10.1016/j.fsidi.2025.301923.
[38] N. Hamamoto, S. Yokoyama, A. Takefusa, and K. Aida, "The Log Analysis Environment to Supoort Classroom Using CoursewareHub," *Procedia Comput. Sci.*, vol. 192, pp. 3154–3164, 2021, doi: 10.1016/j.procs.2021.09.088.
[39] Y. Sun, J. Wai, and Z. Yang, "SemiSMAC : A semi-supervised framework for log anomaly detection with automated hyperparameter tuning," *Inf. Softw. Technol.*, vol. 187, no. March, p. 107869, 2025, doi: 10.1016/j.infsof.2025.107869.
[40] D. Shamsollahi, O. Moselhi, and K. Khorasani, "Automation in Construction Data integration using deep learning and real-time locating system (RTLS) for automated construction progress monitoring and reporting," *Autom. Constr.*, vol. 168, no. PA, p. 105778, 2024, doi: 10.1016/j.autcon.2024.105778.
[41] M. Ma, C. Xu, and J. Han, "Application of an intelligent electrical fire monitoring system based on the EC-IOT framework in high-rise residential buildings," *Syst. Soft Comput.*, vol. 7, no. April, p. 200257, 2025, doi: 10.1016/j.sasc.2025.200257.
[42] A. Sultan, L. S. Saoud, M. Elmezain, M. Heshmat, L. Seneviratne, and I. Hussain,

"Ecological Informatics Autonomous robotic systems for coral reef monitoring : Review and open research issues," *Ecol. Inform.*, vol. 92, no. November, p. 103511, 2025, doi: 10.1016/j.ecoinf.2025.103511.
[43] L. Naizabayeva, D. Zaitov, and N. Seilova, "Integrating Smart Traffic Systems with Real-Time Air Quality Monitoring to Minimize Emissions and Improve Urban Health," *Procedia Comput. Sci.*, vol. 251, pp. 603–608, 2024, doi: 10.1016/j.procs.2024.11.156.
[44] J. Plavšić and I. Mišković, "VR - based digital twin for remote monitoring of mining equipment : Architecture and a case study," *Virtual Real. Intell. Hardw.*, vol. 6, no. 2, pp. 100–112, 2024, doi: 10.1016/j.vrih.2023.12.002.
[45] T. Bányai, S. Nazir, and P. Veres, "Real-Time Operational Decision Making in Municipal Waste Collection Systems Using Internet of Things Technologies," *IFAC Pap.*, vol. 59, no. 10, pp. 148–153, 2025, doi: 10.1016/j.ifacol.2025.09.027.
[46] O. S. Yousif and R. Zakaria, "Web-Based Dashboard of Data Integration for Green Highway Performance Management," *J. Eng. Res.*, vol. 11, no. 2, pp. 60–70, 2022, doi: 10.36909/jer.15477.
[47] M. Just and P. Schubert, "A Dashboard for the Visualisation of Areas of Collaboration Analytics," *Procedia Comput. Sci.*, vol. 256, pp. 360–368, 2025, doi: 10.1016/j.procs.2025.02.131.
[48] B. Srinivasan, W. Li, C. J. Ruth, T. J. Herrman, D. Erickson, and S. Mehta, "Current Research in Biotechnology Rapid quantification of aflatoxin in food at the point of need : A monitoring tool for food systems dashboards," *Curr. Res. Biotechnol.*, vol. 6, no. October, p. 100153, 2023, doi: 10.1016/j.crbiot.2023.100153.
[49] R. Schmocker, S. Sampaio, and K. Cormican, "Unlocking potential : Critical success factors for AI integration in remote patient monitoring systems for post-cardiac surgery care," vol. 256, pp. 861–869, 2025, doi: 10.1016/j.procs.2025.02.188.
[50] M. Hasan *et al.*, "Review article A critical review on control mechanisms , supporting measures , and monitoring systems of microgrids considering large scale integration of renewable energy sources," *Energy Reports*, vol. 10, no. November, pp. 4582–4603, 2023, doi: 10.1016/j.egyr.2023.11.025.
[51] A. Iyda *et al.*, "Internet of Things A conceptual IoT-based early-warning architecture for remote monitoring of COVID-19 patients in wards and at home," *Internet of Things*, vol. 18, p. 100399, 2022, doi: 10.1016/j.iot.2021.100399.
[52] S. Desjardins and D. Lau, "Enhanced operational modal analysis and change point detection for vibration-based structural health monitoring of bridges," *J. Infrastruct. Intell. Resil.*, vol. 3, no. 4, p. 100121, 2024, doi: 10.1016/j.iintel.2024.100121.
[53] C. Zurrón, R. Urraca, and A. Sanz-garcia, "A data-driven methodology for monitoring Total Quality Management (TQM) systems in the Industry 4 . 0 era," *Comput. Ind. Eng.*, vol. 210, no. February, p. 111549, 2025, doi: 10.1016/j.cie.2025.111549.
[54] S. Kumar, S. Chakraborty, B. Verbrugge, and S. Helsen, "Internet of Things Intelligent data-driven condition monitoring of power electronics systems using smart edge – cloud framework," *Internet of Things*, vol. 26, no. February, p. 101158, 2024, doi: 10.1016/j.iot.2024.101158.
[55] A. Febrianto, M. Suef, M. Saiful, and K. Dede, "Results in Engineering Operational efficiency and sustainable asset management of heavy equipment in industry : a data-driven framework," vol. 27, no. July, 2025.
[56] P. F. Karjou, S. K. Saryazdi, P. Stoffel, and D. Müller, "Energy & Buildings Practical design and implementation of IoT-based occupancy monitoring systems for office buildings : A case study," *Energy Build.*, vol. 323, no. July, p. 114852, 2024, doi: 10.1016/j.enbuild.2024.114852.
[57] T. Willoughby, P. Johnson, J. Alvarez, L. O. Silva, and J. Smeets,

"CONSUMPTION REQUIRED TO OPERATE THE," *Int. J. Part. Ther.*, vol. 17, p. 100963, 2025, doi: 10.1016/j.ijpt.2025.100963.

[58] S. Hjelle, P. Mikalef, N. Altwaijry, and V. Parida, "Information & Management Organizational decision making and analytics : An experimental study on dashboard visualizations," *Inf. Manag.*, vol. 61, no. 6, p. 104011, 2024, doi: 10.1016/j.im.2024.104011.

[59] H. Chen and H. Wang, "Design of monitoring data visualization system of water resources based on J2EE architecture," *Desalin. Water Treat.*, vol. 269, pp. 303–312, 2022, doi: 10.5004/dwt.2022.28734.

[60] C. Rondinoni, T. Marchetti, L. A. Betioli, V. H. Malheiro, R. H. Matsuda, and O. B. Filho, "Dashboard for Real-time monitoring of robotized transcranial magnetic stimulation platform," *Brain Stimul.*, vol. 18, no. 4, p. 1339, 2025, doi: 10.1016/j.brs.2025.05.032.

[61] Z. Azadmanjir, M. Sadeghi-naini, M. Dashtkoohi, and M. Moradi-, "Informatics in Medicine Unlocked The design of a quality improvement dashboard for monitoring spinal cord and column injuries," *Informatics Med. Unlocked*, vol. 47, no. August 2023, p. 101489, 2024, doi: 10.1016/j.imu.2024.101489.