International Journal of Social Science (IJSS)
Vol.6 Issue.2 August 2026, pp: 637-644
ISSN: 2798-3463 (Printed) | 2798-4079 (Online)
DOI: https://doi.org/10.53625/ijss.v6i2.13121

Development Of A Web-Based Atm Log Validator Application For Failed Cash Withdrawal Transaction Validation And Filtering

By
Berthine Saul Yunior Wabia[1], Eko Sediyono[2]
[1,2]Department of Informatics Engineering, Faculty of Information Technology, Satya Wacana Christian University, Indonesia
Email: [1]wabiaberthine.bw@gmail.com, [2]eko@uksw.edu

Article Info

Article history:
Received May 27, 2026
Revised Jun 27, 2026
Accepted Jul 30, 2026

Keywords:
ATM log validator,
client-side processing,
failed cash withdrawal transactions,
Prototype method,
web application

ABSTRACT

Automated Teller Machines (ATMs) generate large volumes of transaction log data, making manual identification of failed cash withdrawal transactions inefficient and error-prone. This study developed ATM Log Validator, a web-based application that automatically validates, filters, and displays failed cash withdrawal transactions from JSON log files. The application was built using HTML, CSS, and JavaScript with a client-side processing approach, ensuring that all data remain within the user's browser to enhance data privacy. The Prototype development method was employed, and system functionality was evaluated using Black Box Testing. Performance was assessed using a simulated dataset of 2,000 transaction logs generated from 50 ATM units across nine locations representing the operational environment of Bank Papua Manokwari Branch during 2021–2024. All functional test scenarios achieved a 100% success rate. The analysis identified 700 cash withdrawal transactions, including 85 failed transactions (12.14%). Failure rates declined from 13.92% in 2021 to 8.38% in 2024, with the highest rates recorded at Hotel Fajaroon and SPBU Jalan Baru. These results demonstrate that the application effectively supports ATM transaction auditing and reconciliation without requiring additional server infrastructure.

Corresponding Author:
Berthine Saul Yunior Wabia
Department of Informatics Engineering, Faculty of Information Technology, Satya Wacana Christian University, Indonesia
Email: wabiaberthine.bw@gmail.com

1. INTRODUCTION

Digital transformation in the banking sector has accelerated the adoption of electronic banking services to improve operational efficiency and service quality. Among these services, Automated Teller Machines (ATMs) remain one of the primary transaction channels, enabling customers to perform self-service transactions such as cash withdrawals, fund transfers, bill payments, and balance inquiries. Every ATM transaction is automatically recorded as an electronic transaction log (event log), containing essential information such as machine identification, transaction timestamp, transaction type, transaction amount, and transaction status. These transaction logs serve a strategic role by supporting system monitoring, auditing, transaction reconciliation, incident investigation, and service quality evaluation. Furthermore, effective transaction log management enhances the traceability of system activities, thereby improving operational accountability and facilitating data-driven decision-making [1].

Among the various ATM services, cash withdrawal transactions present one of the highest operational risks because they involve the direct disbursement of funds to customers. Failed cash withdrawal transactions may result from communication network disruptions, hardware malfunctions, system synchronization failures, or cash dispensing errors. Such failures can create inconsistencies among customer account balances, system records, and the cash dispensed by the ATM, thereby requiring timely and accurate auditing and reconciliation processes. In banking operations, transaction logs constitute the primary source of evidence for investigating disputed transactions, resolving customer complaints, and mitigating operational risks. Consequently, comprehensive, secure, and easily analyzable transaction logging has become a critical component of modern ATM security governance [2], [3].

Although transaction logs provide comprehensive records of ATM activities, identifying failed transactions is still frequently performed manually by inspecting JSON- or text-based log files using conventional text editors. This approach becomes increasingly inefficient when processing datasets containing thousands or even tens of thousands of log entries, as it requires substantial processing time, increases the likelihood of identification errors, and

International Journal of Social Science (IJSS)
Vol.6 Issue.2 August 2026, pp: 637-644
ISSN: 2798-3463 (Printed) | 2798-4079 (Online)
DOI: https://doi.org/10.53625/ijss.v6i2.13121

complicates repetitive auditing tasks. Previous studies have demonstrated that event logs constitute valuable sources of information for business process analysis, anomaly detection, and operational compliance assessment when supported by appropriate validation and filtering mechanisms. Moreover, advances in log mining and log analytics have shown that the quality of data validation during the initial processing stage substantially influences the accuracy of subsequent analyses. Therefore, there is a need for an application capable of automatically validating log file structures, filtering failed transactions, and presenting analytical results efficiently without increasing infrastructure complexity [4], [5].

To address these challenges, this study develops ATM Log Validator, a web-based application designed to validate the structure of ATM transaction logs in JSON format, automatically filter failed cash withdrawal transactions, and present the analysis results through an intuitive user interface. Unlike conventional auditing software, which typically requires dedicated installation and server-based infrastructure, the proposed application adopts a client-side processing architecture in which data reading, validation, filtering, searching, and sorting are performed entirely within the user's web browser using JavaScript. This approach provides several advantages, including faster response times, reduced dependence on network connectivity, and enhanced data confidentiality because log files are processed locally without being transmitted to external servers. In the context of sensitive financial information, client-side processing has increasingly been recognized as an effective strategy for minimizing data leakage risks while strengthening user privacy protection [6], [7].

Although research on event log analysis, transaction auditing, and anomaly detection has advanced considerably, most existing studies primarily emphasize process mining, machine learning, and fraud detection techniques that require substantial computational resources, database systems, or server-based infrastructures with relatively high implementation complexity. In contrast, research focusing on lightweight applications capable of automatically validating and filtering ATM transaction logs through client-side processing remains limited. Moreover, previous studies have largely concentrated on developing analytical algorithms rather than providing practical software tools that can be directly utilized by operational personnel and internal auditors to accelerate the identification of problematic transactions. This indicates a research gap in developing lightweight, user-friendly audit log applications that are independent of server infrastructure while still effectively supporting transaction validation and analysis [8], [9].

To address this gap, this study offers several contributions. First, the proposed application employs a client-side processing architecture, enabling all data processing to be performed locally on the user's device without transmitting sensitive information to external servers. Second, the application integrates JSON format validation, failed cash withdrawal transaction filtering, statistical visualization, search, and sorting functionalities within a lightweight and user-friendly web interface. Third, the proposed system is evaluated using a simulated dataset representing ATM operational conditions at Bank Papua Manokwari Branch from 2021 to 2024, providing a realistic implementation scenario without involving actual customer data. Accordingly, this study aims to design and develop the ATM Log Validator application, implement an automated validation and filtering mechanism for failed cash withdrawal transactions, evaluate its functionality through Black Box Testing, and analyze failed transaction patterns within the simulated dataset to assess the effectiveness of the proposed system [6], [8].

2. RESEARCH METHOD

Research Approach

This study adopted a Research and Development (R&D) approach in the field of software engineering to design, develop, and evaluate a web-based application, ATM Log Validator, for validating and filtering failed cash withdrawal transactions from ATM transaction logs. The software was developed using the Prototype development model, which facilitates iterative interaction between developers and users throughout the development process, allowing system requirements to be progressively refined. The Prototype model consists of four main stages: requirement gathering, prototype design, prototype evaluation and refinement, and system implementation. This development approach has been widely applied in web application development because it accelerates the development process while producing systems that better satisfy user requirements [10].

System Requirements Analysis

The initial stage involved identifying the functional requirements of the application based on the characteristics of ATM transaction logs commonly used in banking systems. The analyzed transaction logs contained key attributes, including transaction timestamp, ATM ID, ATM location, customer name, card number, transaction type, transaction amount, transaction status, and ending balance. Based on the requirements analysis, the application was designed to provide the following core functionalities: uploading JSON log files, validating data structure and format, generating transaction statistics, filtering failed cash withdrawal transactions, searching records, sorting

transaction data, and resetting the analysis results. These requirements served as the foundation for designing the user interface and implementing the application.

System Design and Implementation

The ATM Log Validator application was developed as a Single Page Application (SPA) using HTML5, CSS3, and vanilla JavaScript without relying on additional frameworks. All data processing operations adopted a client-side processing architecture, enabling file reading, JSON parsing, validation, filtering, searching, sorting, and statistical calculations to be executed entirely within the user's web browser. File reading was implemented using the FileReader API, whereas JSON data were interpreted using the JSON.parse() function. This architecture was selected to preserve the confidentiality of ATM transaction logs because all processing is performed locally on the user's device without transmitting data to external servers. In addition to enhancing data security, this approach reduces dependence on Internet connectivity during the analysis process.

Research Dataset

The study employed a synthetic dataset designed to resemble ATM operational conditions at Bank Papua Manokwari Branch during the 2021–2024 period. The dataset comprised 2,000 transaction records generated from 50 ATM units distributed across nine locations in Manokwari Regency. Each transaction record contained the transaction timestamp, ATM identifier, ATM location, customer name, card number, transaction type, transaction amount, transaction status, and ending balance. The dataset included three transaction types balance inquiry, fund transfer, and cash withdrawal with both SUCCESS and FAILED transaction statuses. All records used in this study were synthetically generated and did not contain actual customer transaction data or operational records from Bank Papua. The use of a simulated dataset ensured data confidentiality while enabling the application to be evaluated under conditions that closely resemble real-world operational workloads.

Data Analysis

Data analysis was conducted using a quantitative descriptive approach based on the transaction logs processed by the ATM Log Validator application. The analysis began with validating the uploaded JSON file structure to ensure compliance with the predefined format. The application then parsed the data and filtered transactions by transaction type so that only cash withdrawal transactions were processed further. Subsequently, transactions were classified according to their status (SUCCESS or FAILED) to determine the number of successful and failed transactions. Based on the filtered data, the application automatically generated descriptive statistics, including the total number of records, the number of cash withdrawal transactions, the number of successful transactions, and the number of failed transactions. Additional analyses examined the distribution of failed transactions by year, ATM location, and transaction amount to identify failure patterns within the simulated dataset. The analytical results were presented using tables and descriptive statistics to evaluate the effectiveness of the proposed application in identifying failed cash withdrawal transactions.

System Testing

System functionality was evaluated using Black Box Testing, a software testing method that assesses system functionality by comparing expected outputs with actual outputs without considering the internal program structure or source code. This method was selected because it evaluates the application from the user's perspective and determines whether each feature satisfies the predefined functional requirements. Eleven test scenarios were conducted, covering all major application functions, including JSON file upload, file format validation, processing of JSON objects and arrays, drag-and-drop functionality, filtering failed cash withdrawal transactions, data searching, table sorting, statistical calculations, reset functionality, dark mode, and processing of the simulated dataset containing 2,000 transaction records. The observed outputs from each scenario were compared with the expected outcomes to determine whether the application operated as specified.

$$Success\ Rate\ (\%) = \frac{N_s}{N_t} \times 100\%$$

N_s = number of successful test scenarios
N_t = total number of test scenatios

If all test scenarios produced the expected outputs, the application was considered to satisfy the predefined functional requirements. The testing results were subsequently analyzed descriptively to evaluate the effectiveness of the ATM Log Validator application in automatically validating and filtering failed cash withdrawal transactions.

3. RESULTS AND ANALYSIS

3.1. User Interface Implementation

International Journal of Social Science (IJSS)
Vol.6 Issue.2 August 2026, pp: 637-644
ISSN: 2798-3463 (Printed) | 2798-4079 (Online)
Crossref DOI: https://doi.org/10.53625/ijss.v6i2.13121

The ATM Log Validator application was successfully developed as a Single Page Application (SPA) that can be executed directly within a web browser without requiring software installation or server configuration. The main interface features a file upload area implemented as a drag-and-drop drop zone, allowing users to upload JSON log files either by dragging and dropping the file or by selecting it manually through the operating system's file selection dialog. The interface also includes four summary statistic cards that are automatically updated after the transaction log file has been uploaded and processed. Figure 1 illustrates the application interface after successfully uploading and processing the simulated dataset containing 2,000 ATM transaction log records representing the operational scenario of Bank Papua Manokwari Branch.

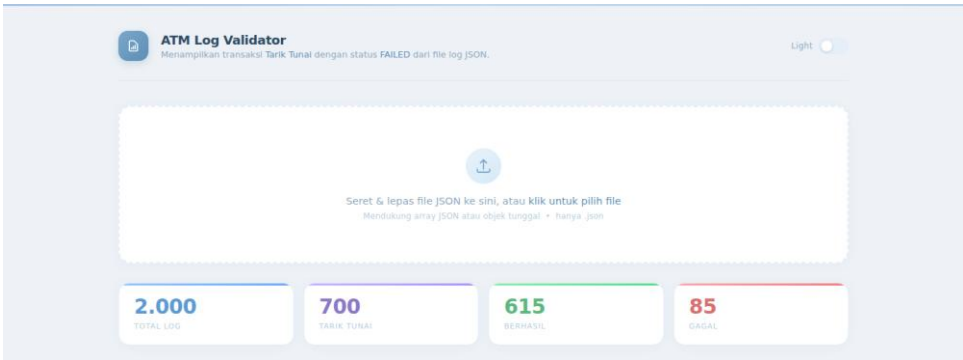

Figure 1. User Interface and Summary Statistics After Dataset Upload

3.2. Data Validation and Filtering Process

After the JSON file was uploaded, the application sequentially performed data structure validation, transaction type filtering, and transaction status classification, as described in the Methods section. Evaluation using the case study dataset containing 2,000 ATM transaction log records demonstrated that the application successfully identified 642 balance inquiry transactions (32.10%), 658 fund transfer transactions (32.90%), and 700 cash withdrawal transactions (35.00%), as summarized in Table 1. Among the 700 cash withdrawal transactions, the application classified 615 transactions as SUCCESS (87.86%) and 85 transactions as FAILED (12.14%). All failed transactions were automatically displayed in the results table, as illustrated in Figure 2.

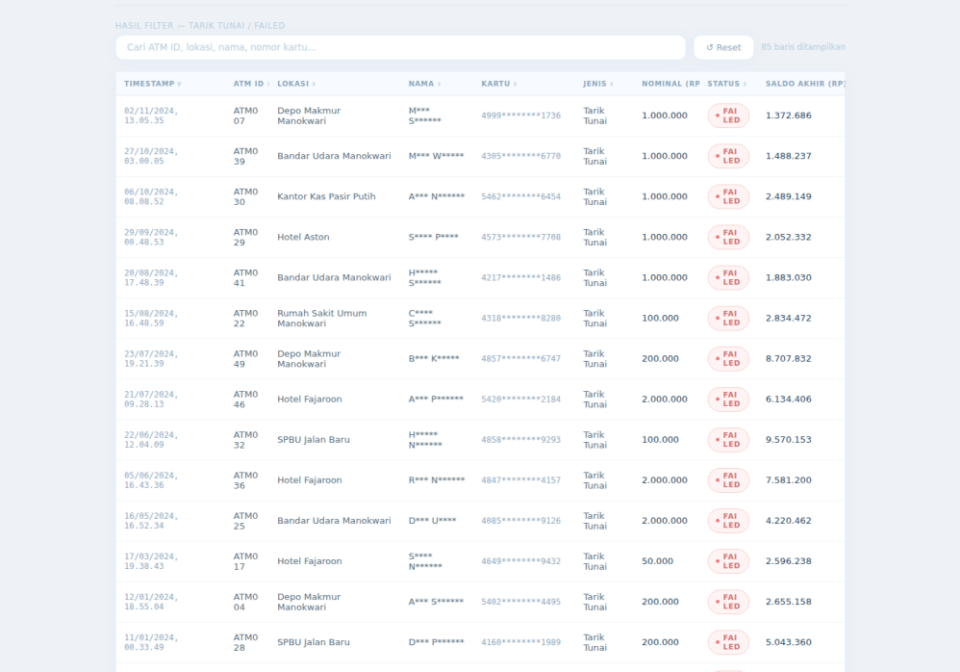

Figure 2. Results Table of Filtered Failed Cash Withdrawal Transactions

Table 1. Summary Statistics of the Case Study Dataset Processing Results (2,000 Records)

Category	Count	Percentage

International Journal of Social Science (IJSS)
Vol.6 Issue.2 August 2026, pp: 637-644
ISSN: 2798-3463 (Printed) | 2798-4079 (Online)
DOI: https://doi.org/10.53625/ijss.v6i2.13121

Total log records	2,000	100.00%
Balance Inquiry	642	32.10%
Fund Transfer	658	32.90%
Cash Withdrawal	700	35.00%
Cash Withdrawal — SUCCESS	615	87.86% of Cash Withdrawals
Cash Withdrawal — FAILED	85	12.14% of Cash Withdrawals

The results were consistent between the application's calculations and independent calculations performed using a separate analysis script. This consistency indicates that the application's filtering logic, based on transaction type followed by transaction status, accurately processed a relatively large transaction log dataset.

a. Black Box Testing

Functional testing was conducted using eleven test scenarios covering file validation, data filtering, and user interactions with the results table. The evaluation also included a dedicated scenario involving the upload of a simulated dataset containing 2,000 transaction records to verify that the application remained responsive when processing a substantially larger dataset than the small sample data used during initial testing. The complete testing results are summarized in Table 2.

Table 2. Black Box Testing Results of the ATM Log Validator Application

No.	Test Scenario	Expected Result	Test Result
1	Uploading a valid .json file	The system reads the file and displays the statistics panel and results table.	As expected
2	Uploading a file with an extension other than .json (e.g., .txt)	The system rejects the file and displays the message: "File must be in .json format."	As expected
3	Uploading a .json file containing invalid (corrupted) JSON	The system displays the message: "Invalid JSON: failed to parse file."	As expected
4	Uploading a JSON file containing a single object (instead of an array)	The system wraps the object into an array and continues processing.	As expected
5	Dragging and dropping a .json file into the upload area	The system processes the dropped file in the same manner as a manually uploaded file.	As expected
6	Dataset contains no FAILED cash withdrawal transactions	The system displays an empty-state view accompanied by an informative message.	As expected
7	Uploading the simulated dataset containing 2,000 transaction records (Manokwari case study)	The statistics display 2,000 total log records, 700 cash withdrawal transactions, 615 successful transactions, and 85 failed transactions.	As expected
8	Entering a keyword in the search field (e.g., "SPBU")	The table immediately displays matching rows based on at least one column.	As expected
9	Clicking a table column header to sort data	The data are sorted in ascending or descending order according to the selected column without page refresh.	As expected
10	Clicking the Reset button	All data, statistics, and the results table are cleared and restored to the initial state.	As expected
11	Enabling dark mode	The interface switches to the dark theme, and the preference is retained in the browser.	As expected

Based on Table 2, all eleven test scenarios (100%) produced results consistent with the expected outcomes, indicating that the application satisfied all predefined functional requirements. In particular, Test Scenario 7 demonstrated that the application successfully processed the operational-scale dataset containing 2,000 transaction records while accurately generating transaction statistics and maintaining responsive performance, confirming its suitability for practical transaction log analysis beyond small-scale sample datasets.

b. Analysis of Failed Cah Withdrawl Transactions in the Simulated ATM Dataset of Bank Ppua Manokwari Branch

To demonstrate the practical application of the proposed system in a realistic operational context, the 85 failed cash withdrawal transactions identified from the case study dataset were further analyzed according to three dimensions: year, ATM location, and transaction amount. As described in the Methods section, the dataset is entirely

International Journal of Social Science (IJSS)
Vol.6 Issue.2 August 2026, pp: 637-644
ISSN: 2798-3463 (Printed) | 2798-4079 (Online)
Crossref DOI: https://doi.org/10.53625/ijss.v6i2.13121

synthetic and was designed to emulate the operational environment of Bank Papua Manokwari Branch without representing actual customer transaction data.

Table 3 summarizes the annual failure rates of cash withdrawal transactions during the 2021–2024 period. The results indicate a gradual decline in the failure rate over time, decreasing from 13.92% in 2021 to 8.38% in 2024, despite the relatively stable number of cash withdrawal transactions each year (166–194 transactions). This trend suggests an overall improvement in transaction reliability within the simulated operational environment.

Table 3. Annual Failure Rate of Cash Withdrawal Transactions (2021–2024)

Year	Total Cash Withdrawals	Failed Transactions	Failure Rate
2021	194	27	13.92%
2022	173	25	14.45%
2023	166	19	11.45%
2024	167	14	8.38%

Table 4 presents the distribution of failure rates across the nine simulated ATM locations. Considerable variation was observed among locations. Hotel Fajaroon exhibited the highest failure rate (20.63%), followed by SPBU Jalan Baru (17.50%), whereas Manokwari City Mall recorded the lowest failure rate (5.80%). These findings illustrate how location-based analysis can assist banking institutions in identifying ATM sites that may require further operational evaluation or maintenance.

Table 4. Failure Rate of Cash Withdrawal Transactions by ATM Location

ATM Location	Total Cash Withdrawals	Failed Transactions	Failure Rate
Hotel Fajaroon	63	13	20.63%
SPBU Jalan Baru	80	14	17.50%
Kantor Kas Pasir Putih	87	13	14.94%
Depo Makmur Manokwari	76	11	14.47%
Rumah Sakit Provinsi Manokwari	75	8	10.67%
Hotel Aston	78	7	8.97%
Bandar Udara Manokwari	91	8	8.79%
Rumah Sakit Umum Manokwari	81	7	8.64%
Manokwari City Mall	69	4	5.80%

Table 5 presents the distribution of transaction amounts for the 85 failed cash withdrawal transactions. The simulated withdrawal amount of Rp2,000,000 accounted for the highest proportion of failed transactions (23.53%), followed by Rp1,000,000 (17.65%). The cumulative requested withdrawal amount associated with all failed transactions was Rp65,850,000. It should be noted that these amounts represent withdrawal requests recorded in failed transactions and do not necessarily indicate funds that were successfully debited from customers' accounts. Nevertheless, such information provides a useful reference for reconciliation and operational auditing processes.

Table 5. Distribution of Transaction Amounts for Failed Cash Withdrawal Transactions

Transaction Amount	Number of Failed Transactions	Percentage
Rp50,000	11	12.94%
Rp100,000	14	16.47%
Rp200,000	12	14.12%
Rp500,000	13	15.29%
Rp1,000,000	15	17.65%
Rp2,000,000	20	23.53%

c. Discussion

The testing results demonstrate that the client-side processing approach implemented in the ATM Log Validator application successfully performs data validation, filtering, searching, sorting, and statistical analysis without requiring communication with a backend server. The application effectively processed the simulated dataset containing 2,000 transaction log records, indicating that this architecture is suitable for small- to medium-scale operational data processing. These findings are consistent with the study by Abb and Rehse (2024), which emphasizes that event logs constitute a critical source of information for business process analysis and that efficient log management and data processing are essential for generating reliable analytical results. In addition to improving processing efficiency, local data processing enhances data confidentiality because sensitive information, such as customer identities and card numbers, remains on the user's device and is not transmitted to external servers. This approach is particularly relevant

in banking environments, where operational efficiency must be balanced with stringent customer data protection requirements [4], [11].

The analysis of the simulated dataset representing the operational environment of Bank Papua Manokwari Branch revealed that the failure rate of cash withdrawal transactions decreased from 13.92% in 2021 to 8.38% in 2024. Although the dataset is synthetic, this trend illustrates how improvements in infrastructure quality, ATM maintenance, and routine operational monitoring may contribute to reducing transaction failures over time. Furthermore, the observed variation in failure rates across ATM locations, ranging from 5.80% to 20.63%, suggests that operational evaluations should be conducted at the individual ATM level rather than relying solely on branch-level summaries. Event log analysis enables organizations to identify transaction patterns requiring further investigation, thereby supporting more efficient auditing than conventional manual inspection. This observation is consistent with the findings of Jans et al. (2014), who demonstrated that event log analysis can identify auditor-relevant transactions and complement traditional auditing procedures [4], [12].

The analysis of transaction amounts further showed that withdrawal requests of Rp1,000,000 and Rp2,000,000 accounted for the largest proportion of failed transactions, representing 41.18% of all failures. From an operational perspective, this pattern may provide an early indication for ATM operators to evaluate cash availability, cash dispenser performance, and transaction limit configurations. Although this study did not investigate the technical causes of transaction failures, these findings may assist banks in prioritizing maintenance activities and conducting further investigations of ATM units exhibiting relatively high failure rates. Consequently, the ATM Log Validator application functions not only as a data filtering tool but also as a decision support system that facilitates data-driven decision-making in ATM service management [3], [12]

From an implementation perspective, the proposed application offers several advantages over conventional manual identification procedures. It integrates JSON format validation, failed cash withdrawal transaction filtering, data searching, table sorting, and statistical visualization within a lightweight web-based interface. These characteristics enable operational staff and internal auditors to use the application without requiring additional software installation or dedicated server infrastructure. Moreover, the client-side processing architecture reduces dependence on network connectivity while preserving transaction data privacy, making the application particularly suitable for environments handling sensitive financial information. This implementation strategy aligns with contemporary log management practices that minimize data transmission as a means of reducing information leakage risks [3], [11].

Despite these contributions, this study has several limitations. First, the validation mechanism is limited to verifying the syntax and basic structure of JSON files and does not include schema validation or consistency checks across mandatory attributes. Second, the evaluation was conducted using a synthetic dataset; therefore, the observed transaction patterns and failure rates should not be interpreted as representing the actual operational conditions of Bank Papua Manokwari Branch. The use of synthetic data was intended to preserve customer confidentiality; however, future validation using anonymized operational datasets would provide a more realistic assessment of the application's performance. Third, the current implementation does not support exporting analysis results to CSV or PDF formats, requiring audit documentation to be performed manually. These limitations indicate opportunities for further enhancement to better support comprehensive operational auditing requirements [13], [14].

Future research may focus on implementing more comprehensive schema validation, integrating anonymized operational transaction data, adopting virtual scrolling techniques to improve performance when processing larger datasets, and incorporating report export functionality in CSV and PDF formats. In addition, integrating process mining techniques and artificial intelligence-based anomaly detection could further enhance the application's capability to automatically identify abnormal transaction patterns. These improvements are expected to broaden the applicability of the ATM Log Validator as an adaptive operational auditing tool that supports the ongoing digital transformation of the banking sector [14], [15].

4. CONCLUSION

This study successfully developed a web-based application, ATM Log Validator, using the Prototype development method to validate and filter failed cash withdrawal transactions from JSON-formatted ATM transaction logs. Functional evaluation using Black Box Testing demonstrated that all application features operated according to the predefined specifications, achieving a 100% success rate across all test scenarios. Evaluation using a simulated dataset containing 2,000 transaction records further confirmed that the application can efficiently and accurately identify failed cash withdrawal transactions while preserving data confidentiality through a client-side processing architecture. These findings indicate that the proposed application provides a practical and lightweight solution for supporting transaction auditing and reconciliation without requiring additional server infrastructure.

International Journal of Social Science (IJSS)
Vol.6 Issue.2 August 2026, pp: 637-644
ISSN: 2798-3463 (Printed) | 2798-4079 (Online)
DOI: https://doi.org/10.53625/ijss.v6i2.13121

Future studies are recommended to evaluate the proposed application using anonymized ATM transaction data to assess its performance under real operational conditions. In addition, the application could be enhanced by incorporating comprehensive schema validation, report export functionality in CSV and PDF formats, and more advanced log analysis techniques, such as process mining and artificial intelligence-based anomaly detection, to further improve auditing efficiency and support data-driven decision-making in banking operations.

REFERENCES

[M. A. Ashfaq, N. Haq, U. Arshad, M. Farooq, and S. U. R. Rehman, "Towards a Trustworthy and Efficient ETL Pipeline for ATM Transaction Data," Sukkur IBA J. Comput. Math. Sci., vol. 7, no. 2, pp. 9–24, Jan. 2024, doi: 10.30537/sjcms.v7i2.1338.

R. Guo, Y. Jia, and L. Shentu, "The effect of audit digital transformation on audit quality: evidence from digital bank confirmations," China J. Account. Stud., vol. 13, no. 3, pp. 247–281, Jul. 2025, doi: 10.1080/21697213.2024.2442769.

A. Panov, "Key Considerations to Effectively Plan and Determine the Scope of an ATM Security Audit Based on PCI DSS," ISACA J., vol. 2, 2024.

L. Abb and J.-R. Rehse, "Process-related user interaction logs: State of the art, reference model, and object-centric implementation," Inf. Syst., vol. 124, p. 102386, Sep. 2024, doi: 10.1016/j.is.2024.102386.

S. Gholamian and P. A. S. Ward, "A Comprehensive Survey of Logging in Software: From Logging Statements Automation to Log Mining and Analysis," 2021, arXiv. doi: 10.48550/ARXIV.2110.12489.

I. Esiyok, P. Berrang, K. Cohn-Gordon, and R. Künnemann, "Accountable Javascript Code Delivery," in Proceedings 2023 Network and Distributed System Security Symposium, San Diego, CA, USA: Internet Society, 2023. doi: 10.14722/ndss.2023.24096.

K. Sim, H. Heo, and H. Cho, "Combating Web Tracking: Analyzing Web Tracking Technologies for User Privacy," Future Internet, vol. 16, no. 10, p. 363, Oct. 2024, doi: 10.3390/fi16100363.

T. D. Krafft, M. P. Hauer, and K. Zweig, "Black-Box Testing and Auditing of Bias in ADM Systems," Minds Mach., vol. 34, no. 2, p. 15, May 2024, doi: 10.1007/s11023-024-09666-0.

D. Sigalov and D. Gamayunov, "Dead or alive: Discovering server HTTP endpoints in both reachable and dead client-side code," J. Inf. Secur. Appl., vol. 82, p. 103746, May 2024, doi: 10.1016/j.jisa.2024.103746.

Priyono, N. Alam, A. Arfian, J. Bastari, and B. O. Lubis, "Penerapan Metode Prototype pada Pengembangan Sistem Informasi Penjualan Jasa Instalasi Internet," Bul. Sist. Inf. Dan Teknol. Islam, vol. 6, no. 4, 2025, doi: https://dx.doi.org/10.33096/busiti.v6i4.3172.

O. Kamal, S. Sohail, and F. Bukhsh, "Optimizing Privacy-Utility Trade-Off in Healthcare Processes: Simulation, Anonymization, and Evaluation (Using Process Mining) of Event Logs:," in Proceedings of the 14th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Dijon, France: SCITEPRESS - Science and Technology Publications, 2024, pp. 289–296. doi: 10.5220/0012766800003758.

M. Jans, M. G. Alles, and M. A. Vasarhelyi, "A Field Study on the Use of Process Mining of Event Logs as an Analytical Procedure in Auditing," Account. Rev., vol. 89, no. 5, pp. 1751–1773, Sep. 2014, doi: 10.2308/accr-50807.

G. Elkoumy, A. Pankova, and M. Dumas, "Differentially private release of event logs for process mining," Inf. Syst., vol. 115, p. 102161, May 2023, doi: 10.1016/j.is.2022.102161.

S. A. Fahrenkrog-Petersen, H. Van Der Aa, and M. Weidlich, "Optimal event log sanitization for privacy-preserving process mining," Data Knowl. Eng., vol. 145, p. 102175, May 2023, doi: 10.1016/j.datak.2023.102175.

P. Ceravolo, M. Comuzzi, J. De Weerdt, C. Di Francescomarino, and F. M. Maggi, "Predictive process monitoring: concepts, challenges, and future research directions," Process Sci., vol. 1, no. 1, p. 2, Sep. 2024, doi: 10.1007/s44311-024-00002-4.