Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 8, No.3, July 2026, pp. 02603038-01 ~ 02603038-013
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v8i3.4219

Design and Development of SIPADMA: A Software Engineering Approach for Water Consumption Anomaly Detection in Public Water Utilities

Laras Wulansari*, Bambang Agus Herlambang

Faculty of Engineering and Informatics, Universitas PGRI Semarang, Jl. Sidodadi Timur No. 24 Semarang, Central Java 50232, Indonesia

*laraswulan0987@gmail.com

Abstract. Water utilities continuously face challenges in identifying abnormal customer water consumption caused by pipe leakage, illegal connections, meter reading errors, and unusual consumption behavior. Conventional manual inspection requires considerable time and often delays corrective actions. This study presents the design and development of SIPADMA (Sistem Deteksi Anomali Pemakaian Air), a web-based information system developed using Software Engineering principles to automate anomaly detection and support operational decision making. The system was developed following the Waterfall Software Development Life Cycle (SDLC), beginning with requirement analysis, UML-based system modeling, implementation using Flask, and functional testing. SIPADMA integrates statistical anomaly detection using Z-Score combined with rule-based scoring and provides interactive dashboards, anomaly filtering, customer history visualization, and automatic report generation. Evaluation using 3,996 billing records representing 666 customers demonstrated that the system successfully identified 163 anomalous accounts requiring further inspection. The developed system improves operational efficiency by automating data analysis and prioritizing field verification, demonstrating the effectiveness of software engineering practices in developing intelligent decision-support systems for public utilities.

Keywords: software engineering, SDLC, waterfall, information system, flask, anomaly detection

(Received 2026-04-30, Revised 2026-05-25, Accepted 2026-06-12, Available Online by 2026-07-17)

1. Introduction

Digital transformation has driven various organizations to utilize information technology to improve operational efficiency and service quality[1], [2]. One of the sectors that has developed significantly is the drinking-water service provider, which is required to deliver services that are fast, accurate, and sustainable. In managing water distribution, companies must monitor the millions of cubic meters of water distributed to customers so that water losses (Non-Revenue Water/NRW) can be minimized. Therefore, information systems that are able to process operational data automatically are becoming an increasingly important need in supporting data-driven decision making.

PDAM Tirta Moedal Kota Semarang, particularly the East Semarang Branch, still faces various obstacles in the process of identifying irregular customer water consumption. So far, the inspection process has been carried out manually by analyzing customer billing reports, which takes a considerable amount of time and depends heavily on the officer's experience. This condition means that indications of pipe leakage, meter recording errors, and unusual spikes in water consumption are often detected late, potentially increasing water losses and reducing the company's operational efficiency[3], [4], [5],[6].

The development of machine learning has opened new opportunities to help automate the analysis of operational data[7]. One approach that is widely used is *anomaly detection*, namely a method that aims to identify data that deviate from normal patterns[8],[9], [10]. In the utility sector, this method has been used to detect various abnormal conditions such as leakage in the distribution network, water theft, and errors in reading customer meters[11], [12], [13]. Nevertheless, most previous studies have focused more on improving the performance of detection algorithms than on developing software systems capable of integrating the data-analysis process with the operational needs of the organization. Recent systematic reviews further confirm the rapid adoption of machine learning for anomaly detection in smart water metering networks, including ensemble-based approaches that improve detection accuracy on imbalanced water-consumption data [24], [25].

From a software engineering perspective, the implementation of the *anomaly detection* method requires not only accurate algorithms but also a structured software development process so that the system is easy to use, maintain, and develop in the future[14], [15], [22]. Few previous studies have discussed how user needs are translated into system design, how the software architecture is built, and how the user interface is designed to support the monitoring and decision-making process effectively[16]. As a result, the resulting solution is often merely an analytical model without producing an application ready for use in an operational environment.

Based on these problems, this study develops SIPADMA (Sistem Deteksi Anomali Pemakaian Air), a web-based application designed using the Software Development Life Cycle (SDLC) Waterfall approach [23]. The system was developed from requirement analysis, modeling using the Unified Modeling Language (UML), and Flask-based application implementation, through to functional testing. In addition to integrating the Z-Score method and *rule-based scoring* to detect anomalies, the system also provides an interactive dashboard, customer data visualization, a search feature, data filtering, and automatic report generation so that it can support the PDAM operational monitoring process more effectively.

The main contribution of this study lies in applying software engineering principles to build an anomaly detection system that can be implemented directly in the operational environment of a drinking-water company. Unlike studies that focus only on algorithm development, this study produces an information system that integrates data processing, anomaly detection, information visualization, and reporting in a single web-based platform. Thus, the developed system is expected to improve the efficiency of the anomaly identification process while supporting data-driven decision making at clean-water service providers.

2. Methods

2.1. Research Design

This study uses the Research and Development (R&D) method with a Software Engineering approach to develop the Water Consumption Anomaly Detection System (SIPADMA). The software development process was carried out using the Software Development Life Cycle (SDLC) Waterfall model. The Waterfall model was chosen because the system requirements had been clearly identified based on field observations, so that each development stage could be carried out sequentially, from requirement

analysis to system implementation. In addition, this model produces complete documentation at every stage, making it suitable for developing information systems in an organizational environment with relatively stable operational needs.
The research stages consist of requirement analysis, system design, software implementation, system testing, and evaluation of the implementation results. All stages were carried out systematically so that the resulting system meets user needs while maintaining good software quality.

2.2. Research Stages

The research was carried out through five main stages, as shown in Figure 1.

Figure 2. Research Stages

2.2.1. Problem Identification

The initial stage of the research was carried out through direct observation at Perumdam Tirta Moedal, East Semarang Branch. The observation aimed to understand the running business process, particularly the process of monitoring customer water consumption and identifying anomalies. In addition to observation, interviews were also conducted with employees responsible for managing customer data to obtain information about the operational obstacles faced. The identification results show that the anomaly detection process is still carried out manually by examining customer billing data every month. This process takes a considerable amount of time, is difficult to perform on large volumes of data, and depends heavily on the officer's experience, so it may cause delays in handling leaks and irregular water consumption.

2.2.2. Requirement Analysis

The requirement analysis stage was carried out to obtain the system's functional and non-functional requirements. The requirement analysis was conducted through several activities, namely observation of the business process, interviews with users, study of customer billing documents, and analysis of the running system. Based on the analysis, two main user groups were obtained, namely the Employee and the Division Head.
The system's functional requirements were then determined, including:

Table 1. Functional & Non-Functional Requirement Analysis

No	Functional Requirements	Non-Functional Requirements
1	Upload billing data	Web-based application.
2	Monitoring dashboard	Ease of use
3	Anomaly detection	Fast response time
4	Data filtering	Browser compatibility
5	Customer search	Ease of system maintenance

6	Customer details
7	Chart visualization
8	Export Excel report.

The requirement analysis results became the basis for the software design process.

2.2.3. System Design

The design stage was carried out using the Unified Modeling Language (UML). Several diagrams were used, including:

a. Use Case Diagram

The Use Case Diagram is used to describe the interaction between the user and the system[17]. This diagram identifies the main functions that can be performed by the Employee and the Division Head, including uploading data, viewing the dashboard, filtering anomaly data, viewing customer details, searching for customers, viewing interactive charts, and downloading reports.

b. Activity Diagram

The Activity Diagram is used to describe the business process flow of each system feature. The diagrams developed include Upload Data, Dashboard, Anomaly Filter, Customer Details, Customer Search, Download Report, and Interactive Chart. These diagrams help ensure that the entire business process has been modeled systematically before implementation.

c. Sequence Diagram

The Sequence Diagram is used to model the communication between objects during the system process. This diagram describes the interaction between the View, Controller, and Database in each main system process, thereby facilitating implementation using the MVC architecture.

d. Class Diagram

The Class Diagram is used to define the software structure. This diagram consists of several main classes, namely Upload Data, DataRekening, Pelanggan, Dashboard, Anomali, ProsesDeteksi, Grafik, Laporan, and Filter. The relationships between the classes serve as the basis for implementing the program code structure.

2.2.4. System Implementation

The implementation stage was carried out based on the UML design results. The system was developed using several technologies, as follows:

Table 2. Technologies used

Component	Technology
Programming Language	Python 3.x
Framework	Flask
Data Processing	Pandas
Numerical Computing	NumPy
Export Excel	OpenPyXL
IDE	Visual Studio Code
Version Control	Git & GitHub

The application architecture applies the Model–View–Controller (MVC) pattern provided by the Flask framework, so that the separation of business logic, interface display, and data management can be carried out effectively[18]. In the implementation stage, several main modules were developed, namely

Dashboard Monitoring, Upload Data, Anomaly Detection Engine, Customer Details, Data Filter, Customer Search, Report, and Chart Visualization. All modules were integrated into a web application accessible through a browser, making it easier for users to monitor customer data.

2.2.5. Anomaly Detection Implementation

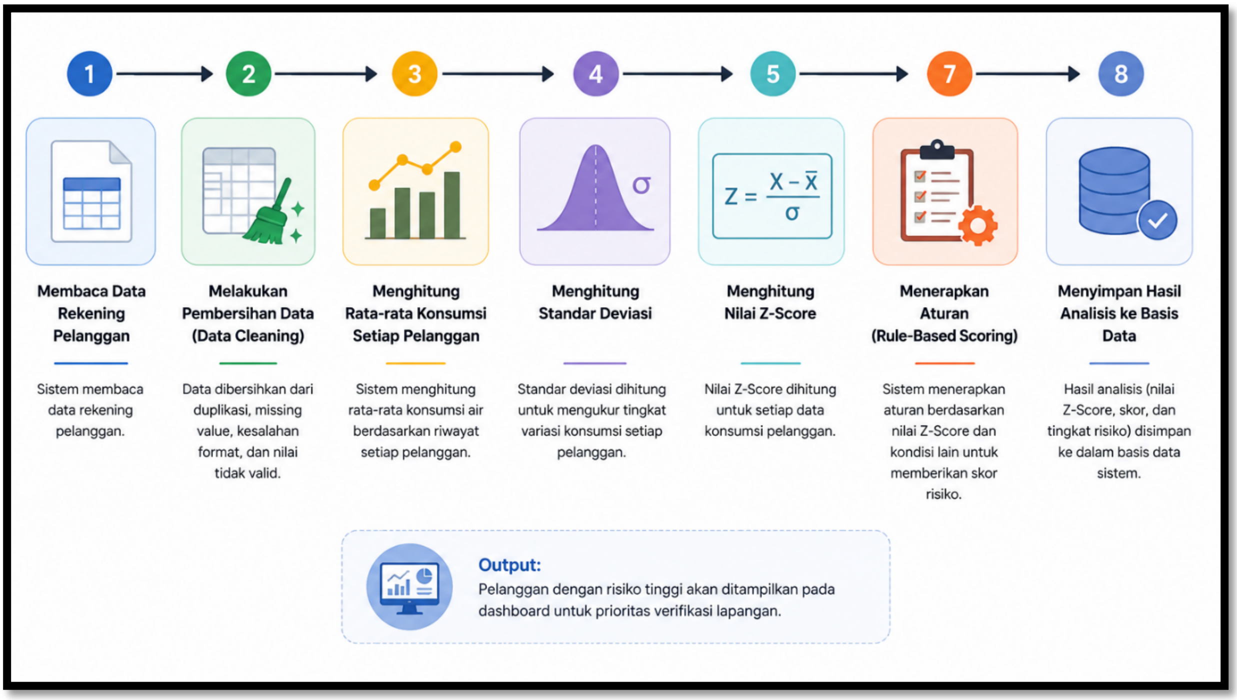


Figure 3. Anomaly Detection Implementation

Anomaly detection is performed using a combination of Z-Score and Rule-Based Scoring[19],[20]. The combination of these two methods allows the system to identify customers whose consumption patterns differ significantly from their historical patterns, while also considering the operational rules applicable at the PDAM.

2.2.6. System Testing

The testing stage was carried out using the Black Box Testing. method. Testing focused on the conformity of the system functions to user needs without considering the program code structure. The system is declared successful if all functions produce output that matches user needs.

2.2.7. System Evaluation

The evaluation was carried out using customer billing data for the period September 2025 to February 2026 consisting of 3,996 transaction records from 666 customers. This dataset was used as test data to evaluate the system's ability to detect water consumption anomalies and to support the PDAM operational monitoring process. The evaluation results were analyzed based on the number of anomalies successfully identified, the risk category, and the ease with which users could utilize the information presented by the system for decision making.

3. Results and Discussion

3.1. System Requirement Analysis Results

The initial stage of SIPADMA development began with requirement analysis through observation and interviews with employees of PDAM Tirta Moedal, East Semarang Branch. The analysis results show that the process of identifying irregular water consumption is still carried out manually by examining customer billing data every month. This condition causes the monitoring process to take a relatively long time, especially as the number of customers continues to grow. In addition, the identification process depends heavily on the officer's experience, which may cause delays in detecting indications of pipe leakage, meter recording errors, and deviations in customer consumption patterns. Based on the analysis, two main actors who will use the system were obtained, namely the Employee and the Division Head. Both actors have functional requirements that include uploading billing data, dashboard monitoring, anomaly detection, customer search, consumption history visualization, filtering data by risk level and customer category, and automatic report generation. These requirements then became the basis for the software design process.

3.2. System Design Results

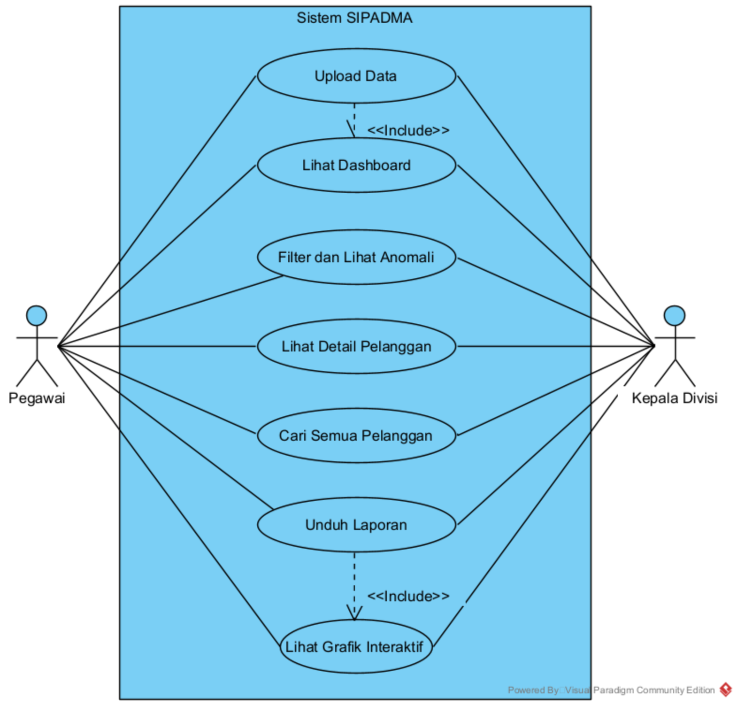


Figure 4. Use Case Diagram

The system design began with the preparation of a Use Case Diagram to describe the interaction between the user and the system. The diagram shows that both the Employee and the Division Head have access to all main SIPADMA functions, namely uploading data, viewing the dashboard, filtering anomaly data, viewing customer details, searching for customers, viewing interactive charts, and downloading reports. Some functions have an *include* relationship, for example the dashboard that always displays chart visualization, and the report download process that includes a summary of the analysis results. This modeling helps ensure that all user needs have been accommodated before implementation. The next design stage was to create the Activity Diagram, used to model the workflow of each main process in the system. The diagrams developed include the data upload process, dashboard access, anomaly data filtering, viewing customer details, customer search, report downloading, and interactive chart visualization. The modeling of activities provides a clear overview of the sequence of business processes, from user interaction to system response. With this documentation, developers can minimize implementation errors while facilitating software maintenance in the future. The Sequence Diagram is used to describe the communication between objects in each system process. The diagram shows how the interface component (*view*), the controller (*controller*), and the database interact with one another in processing user requests. This communication pattern shows that the system has applied a separation of responsibilities between components, so that the business logic is not mixed with the user interface. This approach improves code readability, facilitates development, and supports application scalability. The Class Diagram describes the static structure of the software, which consists of several main classes, namely UploadData, DataRekening, Pelanggan, Dashboard, ProsesDeteksi, Anomali, Grafik, Filter, and Laporan. The relationships between classes show that the system has been designed modularly, with functions separated according to the responsibility of each component. This modular design provides advantages such as ease of maintenance, development of new features, and improvement of software quality, because changes to one module do not directly affect other modules.

3.3. *System Implementation Results*

Figure 5. Dashboard Page 1

The Anomaly Monitoring Dashboard is the main interface of SIPADMA (Sistem Deteksi Anomali Pemakaian Air Pelanggan). This system applies an approach based on *Z-Score* and rule-based scoring to identify patterns that deviate from normal conditions in the billing accounts of PDAM Tirta Moedal, East Semarang Branch customers. Based on six months of processed billing data, the dashboard presents four main performance indicators, namely: a total of 666 customers, 163 accounts detected as anomalous or equivalent to 4.1% of the total 3,996 billing data rows, and 147 at-risk customers requiring further field verification. The visual display on the dashboard includes two pie charts. The first chart presents the distribution of risk levels, with the medium category being the most dominant, followed by low and high risk. The second chart classifies the anomaly types, with the "Irregular" category being the most frequently found, followed by "Drastic Decrease" and "Spike".

Figure 6. Dashboard Page 2

The next dashboard page displays two bar charts. The Anomalies per Tariff Category chart shows the distribution of anomalies based on the customer's tariff category. Category RT 4 dominates with about 98 cases, followed by RT 5, RT 3, and Commercial. Categories RT 2, RT 3, RT 4, and RT 5 are the Household customer classifications set by the PDAM based on water usage capacity, where a higher category number indicates a higher usage capacity, while Commercial refers to customers who use water for commercial or business purposes. This condition indicates RT 4 as the highest-risk group. The Anomalies per Month chart illustrates the monthly trend from September 2025 to February 2026. The number of accounts is relatively stable at 650–670 per month, while anomalies remain low and consistent. At the bottom of the dashboard, an Upload Billing Data feature is available to upload .xlsx, .xls, or .csv files. After uploading, the user runs the analysis with the "Process & Update" button to automatically update the database and the anomaly results.

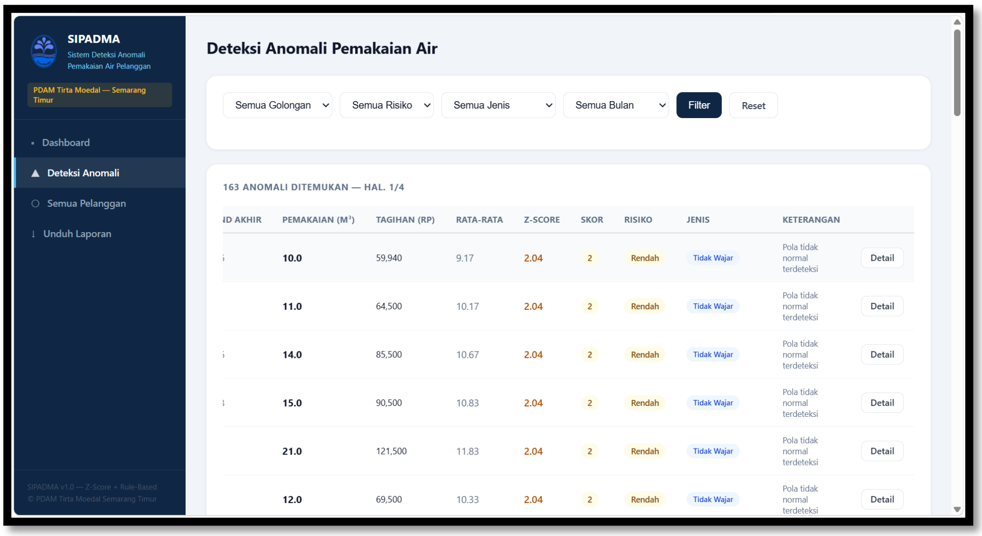


Figure 7. Anomaly Detection Page 1

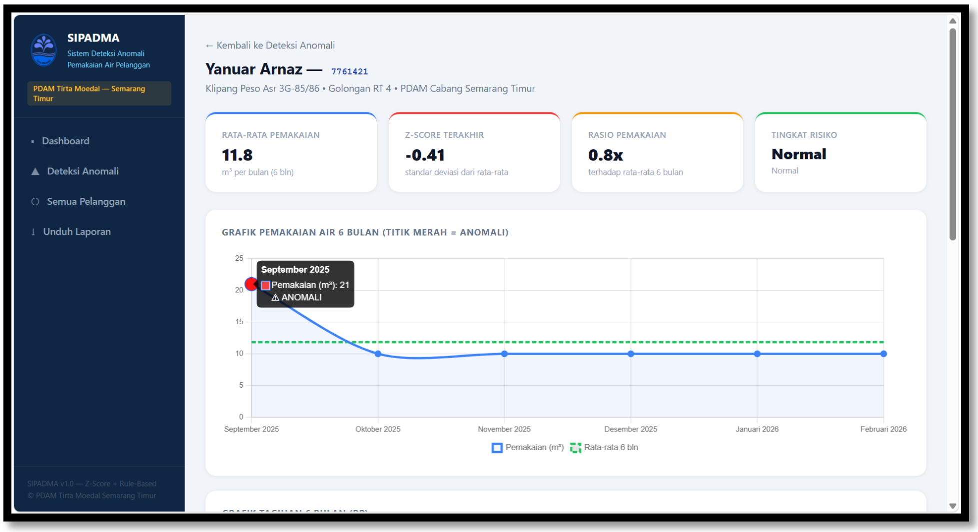


Figure 8. Anomaly Detection Detail Page 1

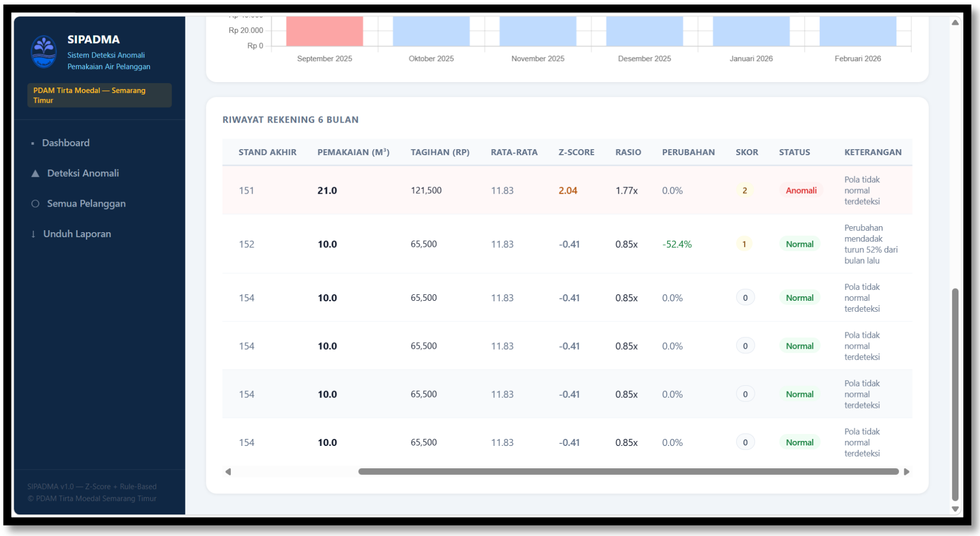


Figure 9. Anomaly Detection Detail Page 2

The Water Consumption Anomaly Detection page presents a comprehensive list of all anomalous accounts. Of the total data analyzed, 163 anomalies were recorded, presented in a paginated table. Each entry displays the customer ID, usage volume (m³), bill (Rp), six-month average usage, *Z-Score* value, rule-based score, risk level, anomaly type, and cause description. A multi-criteria filtering feature is available at the top of the page based on tariff category, risk level, anomaly type, and month period. This feature narrows down and prioritizes the data according to handling needs. The "Filter" button applies the filtering, while "Reset" restores the initial view. The "Detail" button on each row accesses an individual page with four main indicators: average usage, the latest Z-Score, usage ratio, and risk level. Two interactive charts present six-month usage and billing with anomaly points marked in red. The history table displays all attributes per period, including the final meter reading, Z-Score, ratio, month-to-month change, score, and status, enabling the tracing of historical patterns.

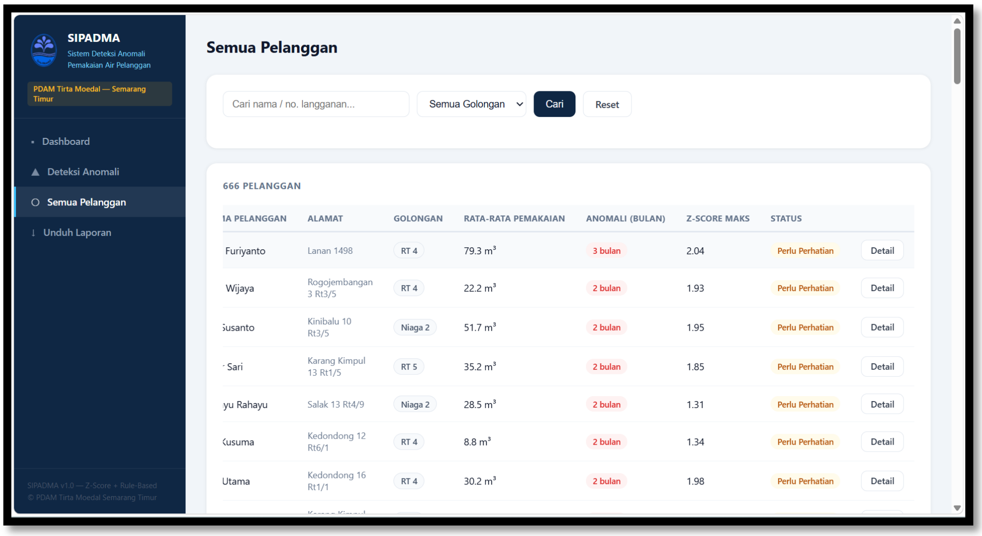


Figure 10. All Customers Page

The All Customers page displays 666 customers with a status summary. The system sorts them by the highest anomaly frequency to facilitate handling prioritization. Each row shows the name, address, tariff category, average usage, number of anomaly months, maximum Z-Score, and priority status. Search and filter features are available for efficient browsing.

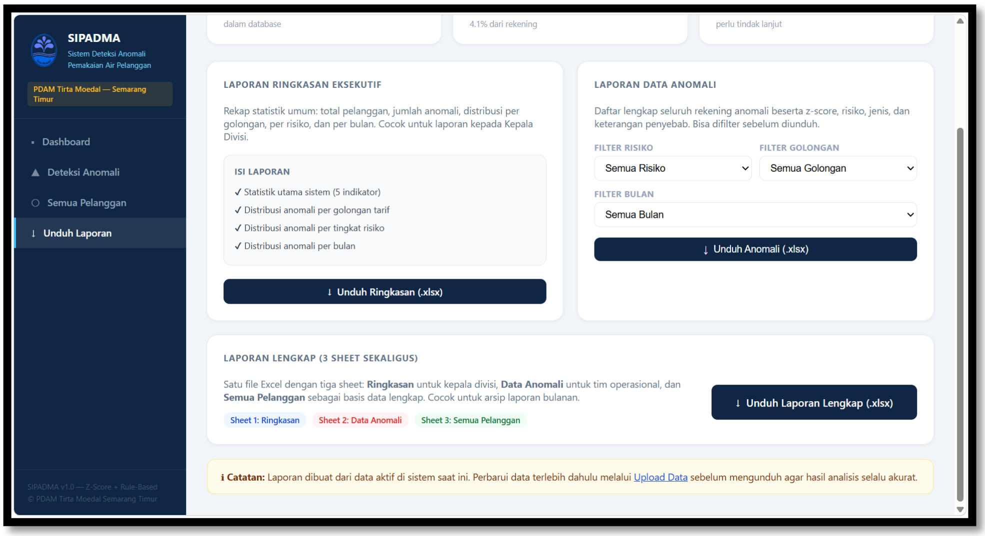
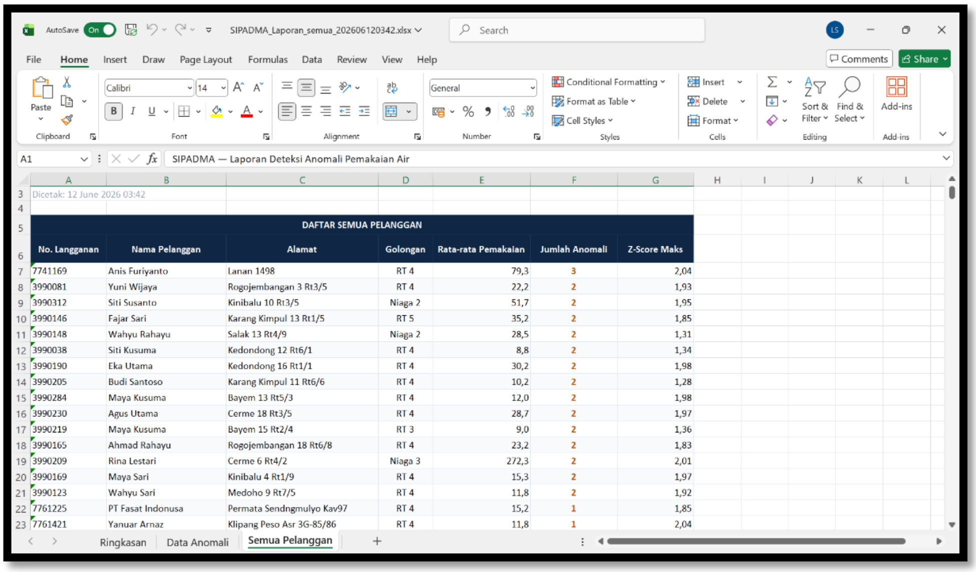


Figure 1. Download Report Page

The Download Report feature provides three .xlsx export options. First, the Executive Summary Report for the Division Head contains the main statistics and the distribution of anomalies by tariff, risk, and month. Second, the Anomaly Data Report for the field team can be filtered by risk, tariff, and period. Third, the Complete Report integrates the Summary, Anomaly Data, and All Customers into a single file. The export results show that the system is able to produce ready-to-use formal documents without further processing.


SIPADMA — Laporan Deteksi Anomali Pemakaian Air

Dicetak: 12 June 2026 03:42

DAFTAR SEMUA PELANGGAN						
No. Langganan	Nama Pelanggan	Alamat	Golongan	Rata-rata Pemakaian	Jumlah Anomali	Z-Score Maks
7741169	Anis Furiyanto	Lanan 1498	RT 4	79,3	3	2,04
3990081	Yuni Wijaya	Rogojembangan 3 Rt3/5	RT 4	22,2	2	1,93
3990312	Siti Susanto	Kinibalu 10 Rt3/5	Niaga 2	51,7	2	1,95
3990146	Fajar Sari	Karang Kimpul 13 Rt1/5	RT 5	35,2	2	1,85
3990148	Wahyu Rahayu	Salak 13 Rt4/9	Niaga 2	28,5	2	1,31
3990038	Siti Kusuma	Kedondong 12 Rt6/1	RT 4	8,8	2	1,34
3990190	Eka Utama	Kedondong 16 Rt1/1	RT 4	30,2	2	1,98
3990205	Budi Santoso	Karang Kimpul 11 Rt6/6	RT 4	10,2	2	1,28
3990284	Maya Kusuma	Bayem 13 Rt5/3	RT 4	12,0	2	1,98
3990230	Agus Utama	Cerme 18 Rt3/5	RT 4	28,7	2	1,97
3990219	Maya Kusuma	Bayem 15 Rt2/4	RT 3	9,0	2	1,36
3990165	Ahmad Rahayu	Rogojembangan 18 Rt6/8	RT 4	23,2	2	1,83
3990209	Rina Lestari	Cerme 6 Rt4/2	Niaga 3	272,3	2	2,01
3990169	Maya Sari	Kinibalu 4 Rt1/9	RT 4	15,3	2	1,97
3990123	Wahyu Sari	Medoho 9 Rt7/5	RT 4	11,8	2	1,92
7761225	PT Fasat Indonusa	Permata Sendngmulyo Kav97	RT 4	15,2	1	1,85
7761421	Yanuar Arnaz	Klipang Peso Asr 3G-85/86	RT 4	11,8	1	2,04

Figure 2. Excel Report Export Results

The report export results show that the system successfully produced a document in Excel format with three worksheets: Summary, Anomaly Data, and All Customers. The document contains the columns Account No., Name, Address, Category, Average Usage, Number of Anomalies, and Max Z-Score, with colored headers and cells with high anomalies automatically marked. This result shows that the system is able to produce formal documents ready for institutional use.

3.4. Implementation Evaluation

System testing was carried out using customer billing data from Perumdam Tirta Moedal, East Semarang Branch, for the period September 2025 to February 2026, consisting of 3,996 billing records from 666 customers. The implementation results show that the system successfully identified 163 accounts indicated as anomalous, or about 4.1% of the entire dataset. Of that number, 147 accounts were categorized as customers with a risk level requiring further verification by field officers.

3.5. Discussion

The research results show that applying the Software Engineering approach through the SDLC Waterfall model is able to produce a structured information system that meets the operational needs of users. All functional requirements obtained in the analysis stage were successfully implemented into the main features of SIPADMA, from data upload, anomaly detection, dashboard visualization, and customer search to automatic report generation. The presence of UML documentation also provides an advantage in the development process because it facilitates communication between developers and users and serves as a reference for software implementation.

From the software architecture perspective, the use of the Flask framework with the Model–View–Controller (MVC) pattern allows a clear separation of business logic, user interface, and data management[21]. This approach improves the maintainability (*maintainability*) of the system because changes to one component do not directly affect other components. In addition, the applied modular structure allows the development of new features, such as the integration of Internet of Things (IoT) sensors, web services (*web service*), and *machine learning* algorithms that are more complex in future research.

Based on the implementation results, SIPADMA is able to automate the analysis of customer billing data that was previously done manually. The interactive dashboard, data filtering feature, and consumption history visualization provide better support for users in decision making. This result shows that the success of system development is influenced not only by the capability of the anomaly detection algorithm but also by the quality of the software engineering that is able to integrate business processes, data processing, and the user interface into a single easy-to-use platform.

4. Conclusion

This study successfully developed SIPADMA (Sistem Deteksi Anomali Pemakaian Air) as a web-based information system designed to support the early detection of irregular customer water consumption at PDAM Tirta Moedal, East Semarang Branch. The system was developed using the Software Development Life Cycle (SDLC) Waterfall **model,** which includes the stages of requirement analysis, system design using Unified Modeling Language (UML), application implementation using the Flask framework, and functional testing to ensure that all features run according to user needs. This approach produced software that is well documented, structured, and easy to maintain.

The implementation results show that SIPADMA is able to automate the analysis of customer billing data through the integration of the Z-Score and Rule-Based Scoring methods. The system provides various main features, such as customer data upload, an interactive dashboard, anomaly data filtering, customer consumption history visualization, data search, and report generation in Excel format. Testing using 3,996 billing records from 666 customers successfully identified 163 accounts indicated as anomalous, with 147 accounts categorized as having a risk level that requires follow-up through field verification. This result shows that the system can help identify at-risk customers more quickly and in a more structured way than the manual process.

From the perspective of Software Engineering, this study shows that applying a systematic software development process can produce a system that not only meets users' functional needs but also has modular characteristics, is easy to develop, and supports data-driven decision making. The design documentation using UML and the application of a Flask-based architecture provide a good foundation for further development, such as integration with Internet of Things (IoT) sensors, the implementation of more adaptive *machine learning* algorithms, and the provision of services based on *Application Programming Interface* (API). Thus, SIPADMA functions not only as an anomaly detection application but also as a foundation for developing a smarter and more sustainable water distribution monitoring system.

Declaration of AI and AI assisted technologies in the writing process

During the preparation of this work the author(s) used Claude.Ai in order to Helps translate from Indonesian to English. It also helps in developing the background of the issue. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] B. R. . Pressman, R. S., & Maxim, *Software Engineering: A Practitioner's Approach (9th ed.)*. McGraw-Hill., 2020.
[2] I. Sommerville, *Software Engineering (10th ed.)*. Pearson, 2016.
[3] T. S. Farley M, *Losses in Water Distribution Networks*. London: IWA Publishing, 2005.
[4] W. A. Liemberger R, "Quantifying the global non-revenue water problem.," *Water Supply*, vol. 19(3), pp. 831–837, 2019, doi: https://doi.org/10.2166/ws.2018.129.
[5] Rogers D, "Leaking water networks: an economic and environmental disaster," *Procedia Eng*, vol. 70, pp. 1421–1429, 2014, doi: https://doi.org/10.1016/j.proeng.2014.02.157.
[6] American Water Works Association, "M36 Water Audits and Loss Control Programs. 4th ed," 2016.

[7] C. J. Kim GH, Trimi S, “Big-data applications in the government sector,” *Commun ACM*, vol. 57(3), pp. 78-85., 2014, doi: https://doi.org/10.1145/2500873.
[8] K. V. Chandola V, Banerjee A, “Anomaly detection: A survey,” *ACM Comput Surv*, vol. 41(3), pp. 1–58, 2009, doi: https://doi.org/10.1145/1541880.1541882.
[9] S. J. L. Breunig MM, Kriegel HP, Ng RT, “Identifying density-based local outliers,” *SIGMOD Rec*, vol. 29(2), pp. 93–104, 2000, doi: https://doi.org/10.1145/335191.335388.
[10] Z. Z. Liu FT, Ting KM, “Isolation Forest,” *IEEE Int Conf Data Min.*, pp. 413–422, 2008, doi: https://doi.org/10.1109/ICDM.2008.17.
[11] H. Y. Wu ZY, “Time series data decomposition-based anomaly detection and evaluation framework for operational management of smart water grid,” *J Water Resour Plan Manag*, vol. 147(9), 2021, doi: https://doi.org/10.1061/(ASCE)WR.1943-5452.0001433.
[12] X. Y. Jian C, Gao J, “Anomaly detection and classification in water distribution networks integrated with hourly nodal water demand forecasting models and feature extraction technique,” *Water Resour Plan Manag*, vol. 148(11), 2022, doi: https://doi.org/10.1061/(ASCE)WR.1943-5452.0001616.
[13] K. D. Nicholaus IT, Park JR, Jung K, Lee JS, “Anomaly detection of water level using deep autoencoder,” *Sensors*, vol. 21(19), 2021, doi: https://doi.org/10.3390/s21196679.
[14] A. S, *Information systems: Foundation of e-business. 4th ed*. Pearson Education, 2019.
[15] C. D. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, “Fundamental algorithms for scientific computing in Python,” *Nat Methods*, vol. 17, pp. 261–272, 2020, doi: https://doi.org/10.1038/s41592-019-0686-2.
[16] urii D. Oleg Melnikov and V. Degeler, “SIAM: Statistical Identification of Anomalies via Mahalanobis metrics: an open-source framework for identifying anomalies in water distribution networks,” *Softw. Impacts*, 2026, doi: https://doi.org/10.1016/j.simpa.2026.100842.
[17] OMG, *Unified Modeling Language (UML) Version 2.5.1*. Object Management Group, 2017.
[18] K. R. Bass L, Clements P, *Software Architecture in Practice. 4th ed*. Boston: Addison-Wesley, 2022.
[19] Montgomery DC, *Introduction to Statistical Quality Control. 8th ed*. Hoboken: Wiley, 2020.
[20] McKinney W, “Data structures for statistical computing in Python,” *Proc Python Sci Conf*, pp. 56–61, 2010, doi: https://doi.org/10.25080/Majora-92bf1922-00a.
[21] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, “Array programming with NumPy,” *Nature*, pp. 357–362, 2020, doi: https://doi.org/10.1038/s41586-020-2649-2.
[22] W. U. M. Rahmadoni J, Akbar R, “Web-Based Cooperation Information System at The Science Techno Park Technology Business Development Center,” *J Appl Eng Technol Sci*, vol. 3(2), pp. 156–167, 2022, doi: https://doi.org/10.37385/jaets.v3i2.806.
[23] R. K. Rikiti MZA, Kusmayanto E, “Design and Development of Help Patient Data Management Information System at Sinergi Atap Negeri Foundation Web-Based Using Codeigniter,” *J Appl Eng Technol Sci*, vol. 3(2), pp. 235–245, 2022, doi: https://doi.org/10.37385/jaets.v3i2.792.
[24] M. N. Kanyama, F. B. Shava, A. M. Gamundani, and A. Hartmann, “Machine learning applications for anomaly detection in Smart Water Metering Networks: A systematic review,” *Phys Chem Earth*, vol. 134, 2024, doi: https://doi.org/10.1016/j.pce.2024.103558.
[25] M. N. Kanyama, F. B. Shava, A. M. Gamundani, and A. Hartmann, “AI-driven anomaly detection in smart water metering systems using ensemble learning,” *Water*, vol. 17(13), 2025, doi: https://doi.org/10.3390/w17131933.