Type of Contribution:
Research Paper
Review Paper
Case Study

ENERGY: JURNAL ILMIAH ILMU-ILMU TEKNIK
Vol. 16, No. 1 (2026) pp 123-140
DOI: 10.51747/energy.v16i1.1618
E-ISSN: 2962-2565

This article contributes to:
SUSTAINABLE DEVELOPMENT GOALS
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE

Design and Implementation of a Web-Based Monthly Health Reporting System for the North Lampung District Health Office

Dilla Meilita[1*], **Ryan Aji Wijaya**[1]

[1] Department information system and technology, University Muhammadiyah Kotabumi, 34517, **Indonesia**

*2259201076@umko.ac.id

Abstract

The management of public health data at the district level still faces significant challenges, including manual reporting processes, data redundancy, and delayed information aggregation. These issues are also observed in the North Lampung District Health Office, where monthly health reports are still processed using fragmented and unintegrated systems. This study aims to design and implement a web-based monthly health reporting information system to improve data accuracy, integration, and reporting efficiency. The system was developed using the Rapid Application Development (RAD) method, which emphasizes iterative development and active user involvement. The system integrates health center data, disease classification (ICD-10), and LB1 morbidity reporting into a centralized platform equipped with real-time dashboards. Software evaluation was conducted using black-box testing and white-box testing, including cyclomatic complexity and code coverage analysis. The results indicate that all functional modules operate correctly, with code coverage exceeding 87% across all components. The system successfully reduces reporting time, improves data consistency, and enhances decision-making support for health authorities. Therefore, the proposed system provides an effective and reliable solution for digital transformation in regional health reporting systems.

Keywords: Health Information Systems, Monthly Reporting, Rapid Application Development (RAD), Web-Based Application

Article Info
Submitted: 2026-05-18
Revised: 2026-06-12
Accepted: 2026-06-18
Published: 2026-06-19

Publisher
Universitas Panca Marga

1. Introduction

Public health services represent a fundamental component of regional development, requiring accurate, timely, and reliable data to support evidencebased decision-making [1], [2]. At the district level, Health Offices are responsible for collecting, processing, and analyzing health reports from multiple service units, including Community Health Centers (Puskesmas) [3]. However, in many developing regions, including North Lampung District, health reporting systems are still dominated by manual procedures or semi-digital tools that are not fully integrated [4]. This condition creates several operational challenges, particularly in terms of data inconsistency, delayed reporting cycles, and limited real-time access to health information. In addition, fragmented reporting mechanisms increase the risk of human error during data aggregation, which may negatively affect the accuracy of health policy formulation at the district level [5] by stakeholders regarding urgent public health issues.

To address these challenges, a digital transformation is required through the implementation of a centralized web-based health information system. Web-based systems offer high accessibility and enable distributed data input from multiple health service units into a single integrated database[[6], [7]. This approach also supports paperless administration and improves data visualization through interactive dashboards for decision support.

Previous studies have developed various web-based health information systems; however, most of them remain limited to basic data digitization and simple reporting dashboards. These systems often lack structured validation mechanisms at the data entry level, which may still allow inconsistent or invalid data to enter the system [8]. In contrast, this study proposes a more advanced approach by implementing a multi-level data validation mechanism at the Puskesmas input stage before data is stored in the central database. This mechanism ensures that all submitted data complies with predefined structural and logical rules, thereby reducing redundancy and improving data quality at the source.

Furthermore, this research integrates a consolidated database structure capable of managing long-term historical health data, enabling trend analysis and improving the responsiveness of health authorities. The system also incorporates structured evaluation methods, including black-box testing and white-box testing, to ensure both functional correctness and internal logic reliability.

To support rapid and adaptive system development, this study employs the Rapid Application Development (RAD) method. RAD emphasizes iterative development and continuous user involvement, allowing system requirements to be refined through direct feedback from end users during the design phase [9], This approach ensures that the final system is aligned with operational needs and administrative regulations in the health sector. Overall, this study aims to develop a web-based monthly health reporting information system that not only digitizes reporting processes but also strengthens data integrity, system reliability, and decision-making capability within regional health management.

2. Methods

Rapid Application Development (RAD) is a software development method derived from the waterfall model that focuses on fast delivery and flexibility. RAD uses shorter, adaptive development stages, allowing user requirements to be implemented more quickly and efficiently [10].

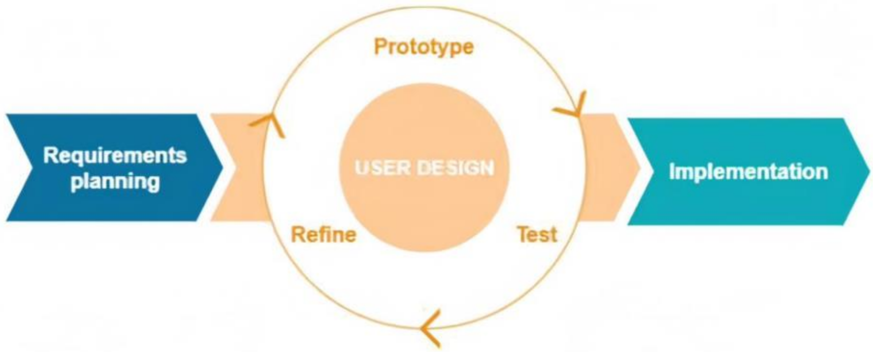

Figure 1. RAD method

Based on the Rapid Application Development (RAD) workflow shown in the **Figure 1**, the following is an explanation of each research stage at the North Lampung District Health Office:

Requirements Planning

In this initial stage, the researcher identifies problems in the current health reporting system in North Lampung District by defining system goals, scope, and functional requirements. Discussions are conducted with health office staff to align system needs and ensure all key administrative data is properly mapped from the beginning

User Design

This phase involves collaboration between the researcher and health office staff to iteratively design the system. It includes developing interface prototypes, user testing, and continuous refinement based on feedback until the database structure and system design are finalized. This process ensures the system is user-friendly and easy to operate without complex training.

Implementation

The final phase involves implementing the approved design into fully functional program code. The system is then deployed on a web server and integrated with the health reporting database. After successful final testing, including black-box testing, white-box testing, Cyclomatic complexity, Code coverage it replaces the manual reporting process at the North Lampung District Health Office, enabling a more efficient and structured digital system.

3.Results and Discussion

The implementation of this web-based information system streamlines organizational workflows by replacing manual, decentralized data entry with a centralized digital process. This shift removes inefficiencies such as manual data transfer and repeated validation, enabling faster data consolidation. As a result, decision-makers gain real-time access to validated data and dashboard analytics, which supports more accurate and timely strategic decisions while reducing human error.

3.1 UML System Modeling

Unified Modeling Language (UML) is a standardized visual modeling tool used in object-oriented software development. It helps represent system design through clear diagrams, making the design process easier to understand. UML includes several diagram types, such as Use Case, Activity, Sequence, Class, and Deployment diagrams, to describe both system structure and behavior comprehensively [11].

3.1.1 Use Case

A Use Case diagram visually represents system functionality by showing the interaction between external actors and internal system features, as shown in **Figure 2**. It helps developers map how users interact with the system to achieve specific goals in a structured way [12].

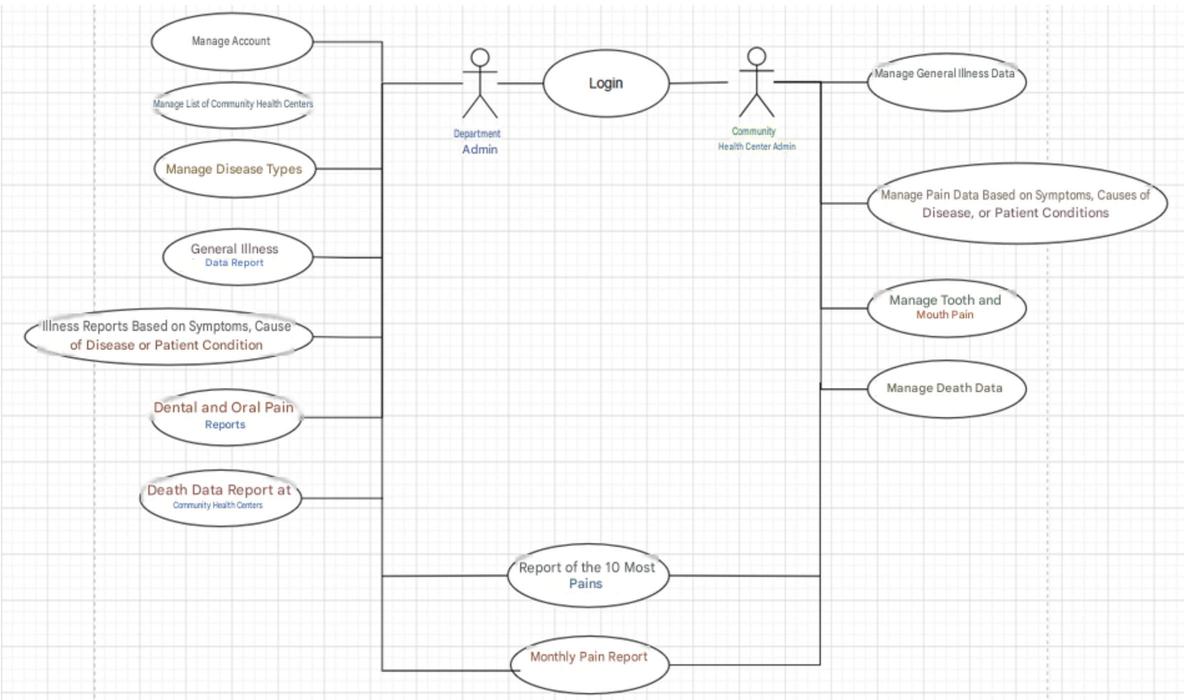

Figure 2. Use Case

The Use Case Diagram illustrates the functional flow of the health information system involving two main actors: the Health Office Admin and the Community Health Center Admin, both accessing the system through a login gateway. The Puskesmas Admin is responsible for managing operational health data, including morbidity, symptom-based cases, dental and oral health, and mortality data at the primary care level. Meanwhile, the Health Office Admin handles higher-level administrative functions such as user management, health center data, disease classification, and access to analytical reports, including monthly morbidity summaries and the top 10 diseases. This separation of roles enables structured data synchronization between health centers and the district health office, improving data accuracy, consistency, and decision-making efficiency.

3.1.2 Activity Diagram

Activity diagrams (in **Figure 3**) is defined as a visual representation that maps the workflow or sequence of operational activities within a system under development [13].

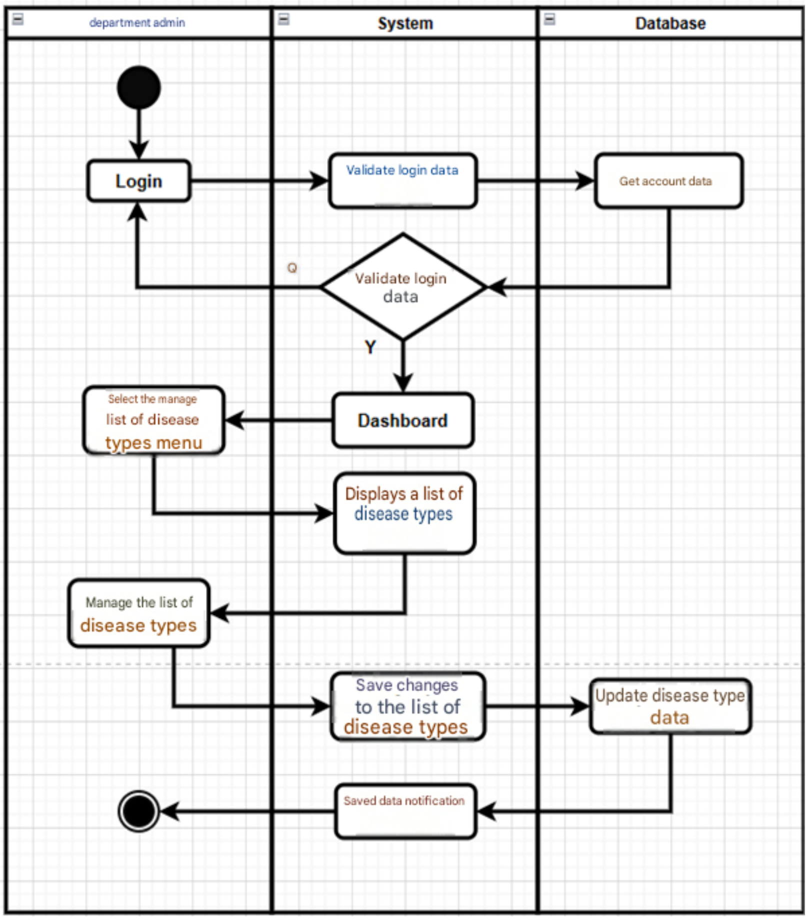

Figure 3. Disease Category Management

The Activity Diagram illustrates the workflow for managing disease type data, starting with authentication by the Health Office Admin. After successful login and system validation, the user is directed to the dashboard and selects the disease management menu. The system then displays the available disease records, allowing the admin to update or modify data as needed. All changes are saved to the database, and the process ends with a confirmation message indicating successful data storage, ensuring a transparent and controlled workflow.

3.1.3 Activity Diagram

A sequence diagram (in **Figure 4**) is defined as a visual representation that maps interaction patterns and the chronology of message exchanges between objects within a system [14].

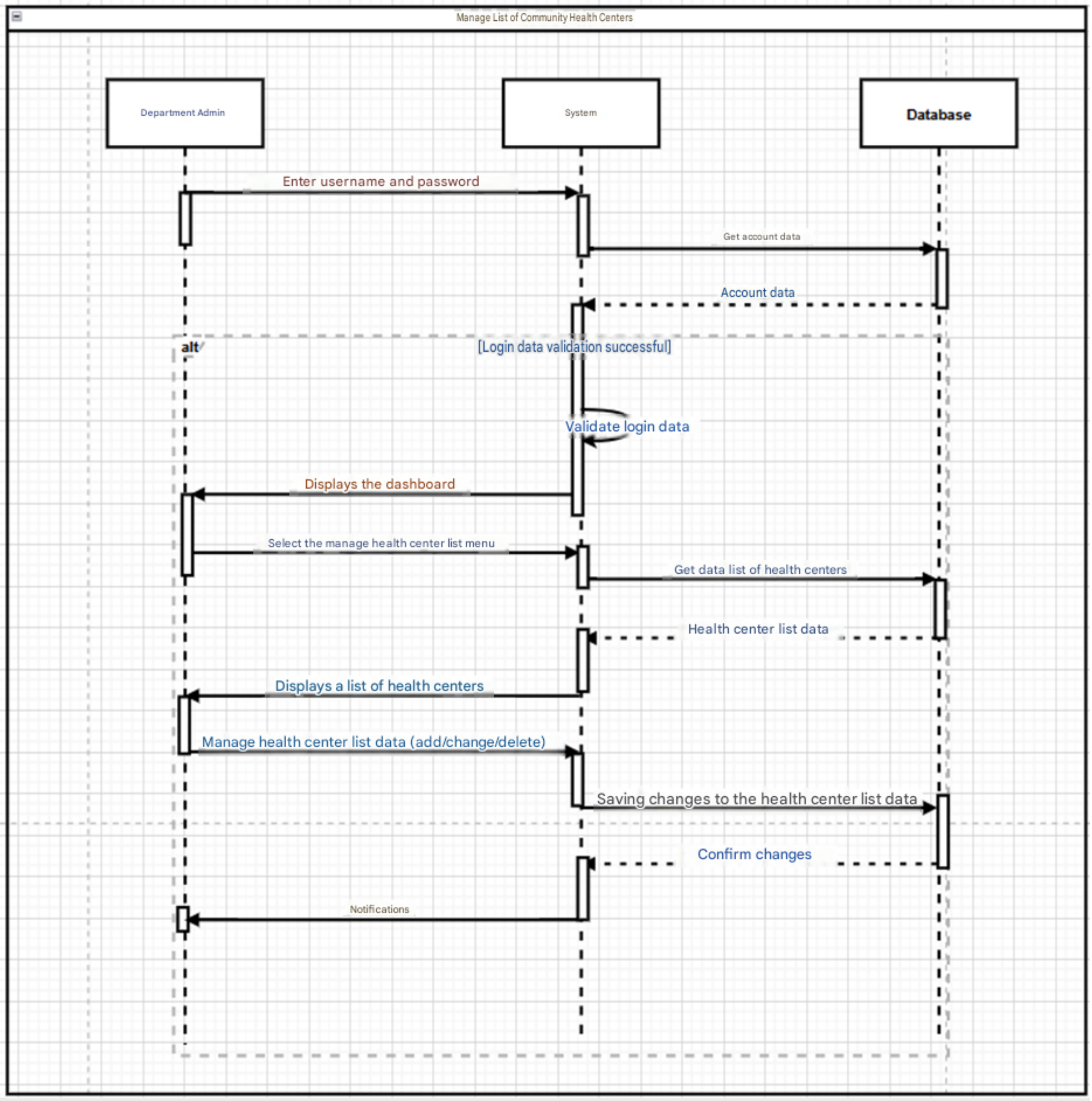

Figure 4. Manage Puskesmas Directory

The Sequence Diagram illustrates the interaction between the Health Office Admin, the system, and the database in managing the health center list. The process begins with user authentication using a username and password, which is validated by the system through the database. After successful login, the dashboard is displayed and the admin selects the "Manage Health Center List" menu. The system retrieves the relevant data from the database and displays it to the user. The admin can then add, update, or delete health center information, which is processed by the system and stored back into the database. The process ends with a confirmation message indicating that all changes have been successfully saved.

3.1.4 Database

A sequence diagram (in Figure 5) is defined as a visual representation that maps interaction patterns and the chronology of message exchanges between objects within a system [14].

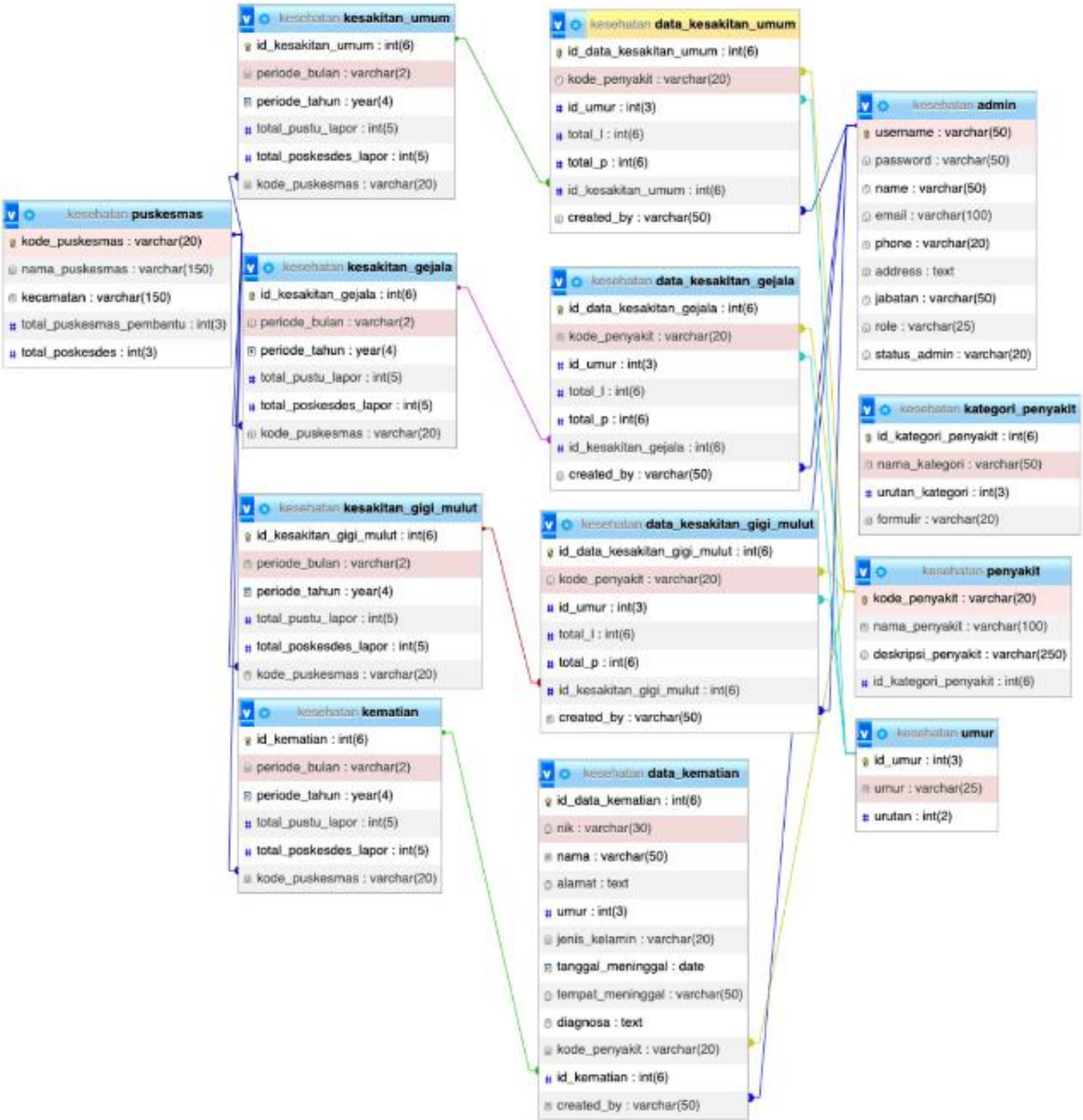

Figure 5. Database

The health information system is built using a relational database architecture that integrates operational data from Community Health Centers (Puskesmas) with structured morbidity and mortality reporting. The database includes master tables such as health centers, diseases, and age groups, along with transaction tables for general morbidity, symptom-based cases, dental and oral health, and mortality data. Data relationships are maintained using foreign keys such as kode_puskesmas and kode_penyakit to ensure data integrity, while the admin table manages access control and tracks data changes through the created_by field. This normalized structure enables efficient and accurate health data reporting based on time periods and service locations.

3.2 Implementation

To validate the effectiveness of the developed system, this study defines measurable key performance indicators (KPIs) across operational, user, and technical aspects. Operationally, the system reduces reporting time significantly, shortening the monthly consolidation process from 3–5 days to real-time processing upon data submission. This improvement increases efficiency and reduces manual data entry by around 80%, minimizing human errors. From a technical perspective, system performance shows stable behavior, with page load and database response times averaging below 2 seconds per request.

a. Dashboard Interface

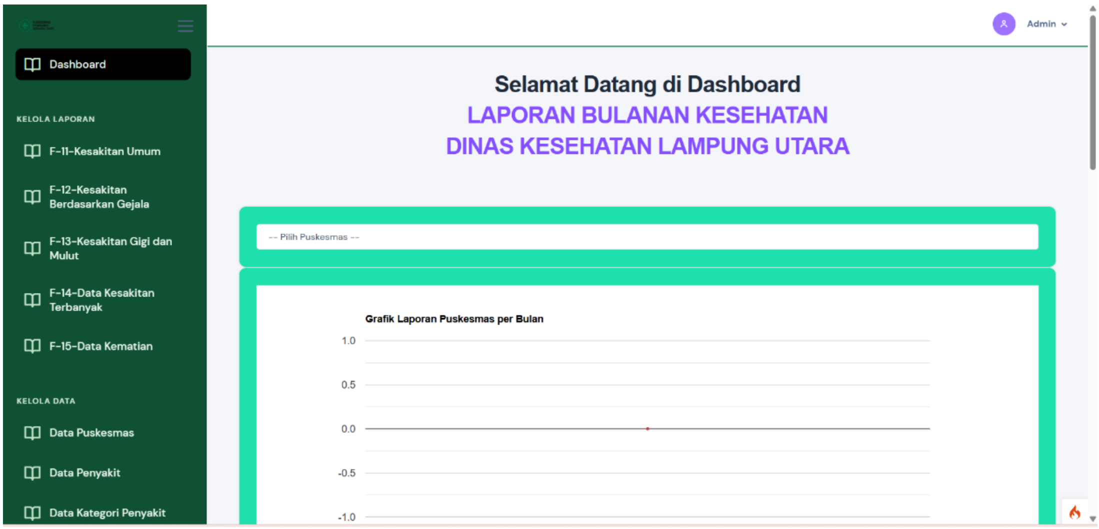

Figure 6. Dashboard

The Figure 6 the main dashboard interface of the Monthly Health Reporting Information System for the North Lampung District Health Office. The interface uses a simple and functional design. A structured sidebar menu on the left enables navigation for managing report formats (F-11 to F-15), along with master data such as health centers and disease classifications. The central area displays a monthly report graph that supports real-time monitoring of health trends based on selected health centers, thereby improving supervision efficiency and supporting better decision-making processes.

b. Health center list Interface

The Figure 7 presents the Health Center List interface, which functions as a centralized master data management module for health facilities in North Lampung District. The page displays a summary of 27 registered health centers along with a detailed table containing health center codes, names, sub-district locations, and supporting facilities such as Auxiliary Health Centers

(Puskesmas Pembantu) and Village Health Posts (Poskesdes). The interface also includes features for adding new health center data, a search function, and export options (Excel, PDF, and Copy), which improve efficiency in managing and reporting regional health infrastructure data.

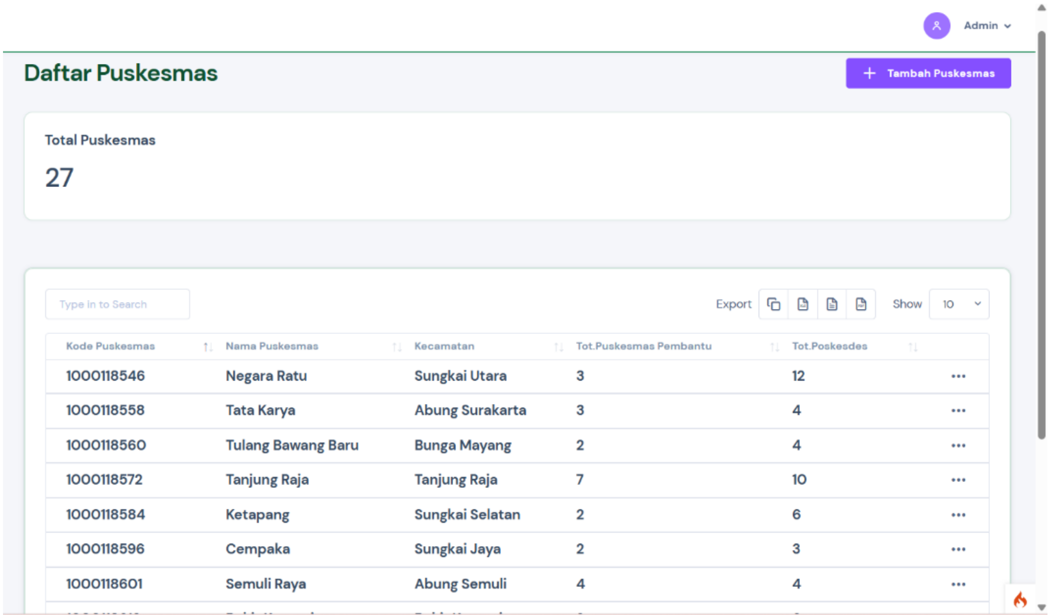

Figure 7. Health Center List

c. Disease List Interface

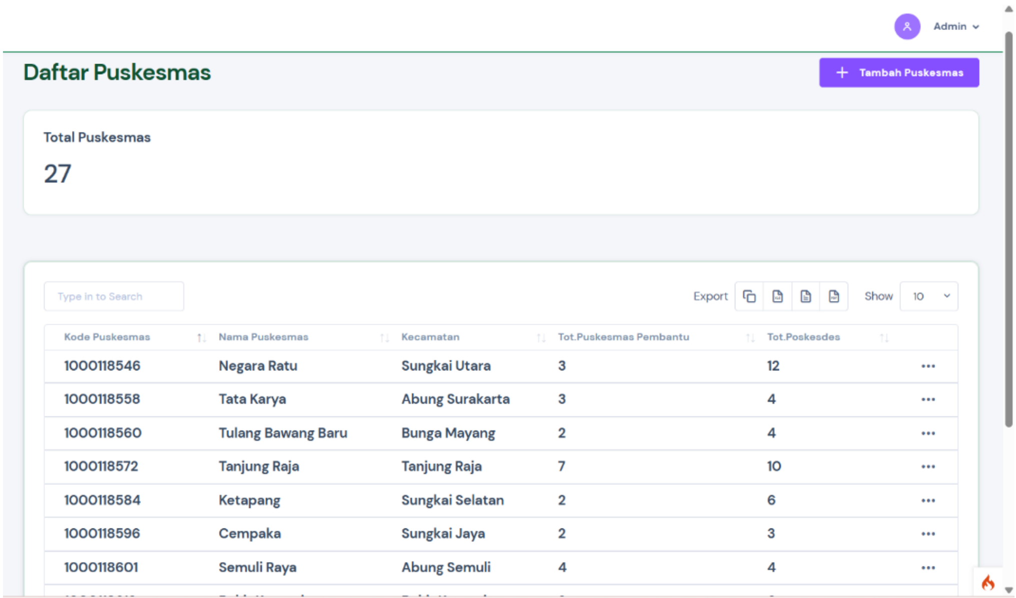

Figure 8. Disease List

The **Figure 8** shows the Disease List interface, which functions as a reference database for medical classification in the reporting system. It contains 292 disease entries organized in a table with diagnostic categories, ICD codes, disease names, and descriptions. The interface also includes features for adding new disease types, searching data, and exporting documents, supporting standardized medical data management and regular updates in accordance with health regulations in North Lampung District.

d. General Morbidity Data List Interface

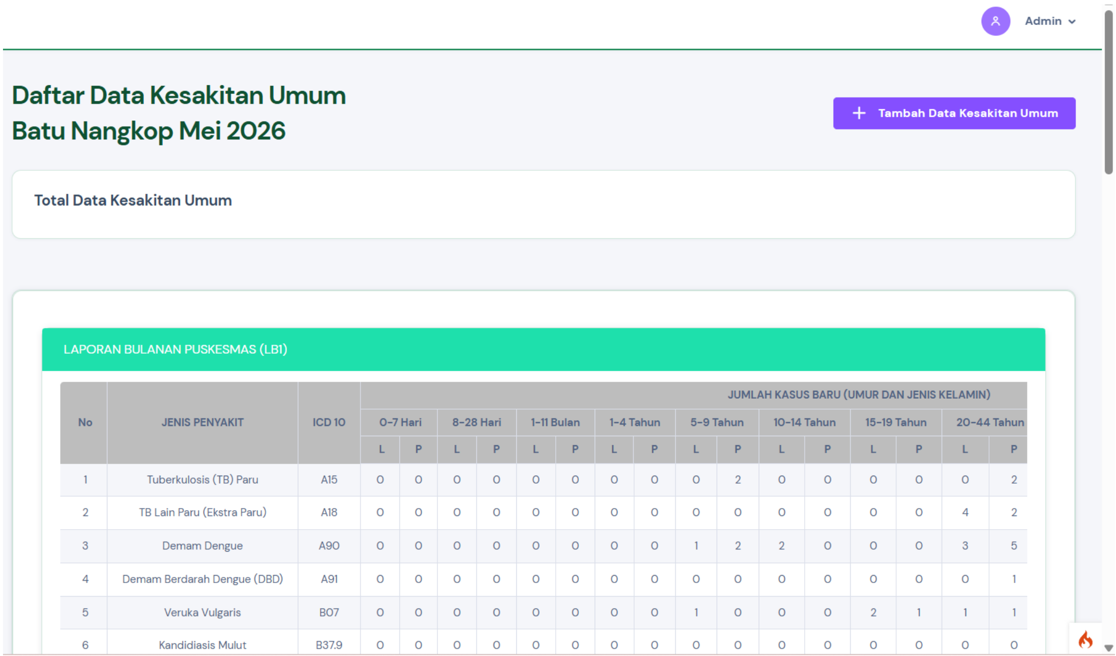

Figure 9. General Morbidity Data List Interface

The **Figure 9** presents the General Morbidity Data List (LB1 Report) interface for the Batu Nangkop Health Center for the May 2026 period. It displays ICD-10-based disease classifications integrated with patient demographics, including age groups and gender distribution. The interface also provides an "Add General Morbidity Data" feature, enabling structured digital reporting from health centers to the district office and improving the accuracy of epidemiological monitoring and disease tracking.

3.3 Testing

Testing using the black-box method aims to evaluate the application's functional capabilities through an input- and output-based approach [15]. The tester designs various specific input conditions to ensure that each program feature operates accurately, thereby guaranteeing the integrity of the software's functionality for the user.

3.4 White-BoxTesting

White-box testing was conducted to evaluate the internal logic, control flow, and code structure of the developed monthly health reporting system. Unlike black-box testing, which focuses on system outputs, white-box testing examines how each logical path, condition, loop, and database operation is executed within the program. In this study, it was applied to key modules such as authentication, disease management, health center management, LB1 reporting, search and filtering, data export, and dashboard visualization.

Tabel 1. Blackbox Testing

No	Test case	Input Scenario	Expected Result	Actual Result	Status
1	Admin Authentication	Enter valid username and password	System validates credentials and redirects to the Dashboard	System redirected to Dashboard successfully	succeed
2	Manage Disease Types	Add a new disease with ICD-10 code (e.g., A15) and name.	Data is permanently stored in the database and appears in the list	Data stored and displayed accurately	Succeed
3	Manage Health Center	Update existing Health Center information (e.g., Puskesmas Batu Nangkop)	System updates the database and shows a success notification	Database updated with success message	Succeed
4	LB1 Report Entry	Input monthly morbidity data for specific age and gender categories	System processes the data and integrates it into the cumulative report	Data processed and correctly displayed in the table	Succeed
5	Data Export	Click the 'Export to PDF/Excel' button on the reports page	System generates and downloads the file in the selected format.	File generated and downloaded successfully	Succeed
6	Search & Filter	Enter a disease name or code in the search bar	Table filters and displays only relevant matching results	Filter results matched the search query	Succeed

The testing approach used basis path testing supported by cyclomatic complexity analysis. Cyclomatic complexity (V(G)) was calculated using the formula $V(G) = P + 1$, where P represents the number of decision points in the program. This metric determines the number of independent execution paths that must be tested in each module. Lower complexity indicates simpler and more maintainable code, while higher complexity reflects more complex logic requiring additional test cases.

Tabel 2. White-BoxTesting Result Based on Cyclomatic Complexity

No	Module Tested	Main Logic Tested	Predicate Nodes	Cyclomatic Complexity V(G)	Independent Paths Tested	Result
1	Admin Authentication	Username/password validation, session creation, role checking	3	4	4	Passed
2	Disease Type Management	Add, update, delete, duplicate ICD-code validation	4	5	5	Passed
3	Health Center Management	Data validation, update process, delete restriction	4	5	5	Passed
4	LB1 Morbidity Report Entry	Period validation, age-group input, gender-based data, total calculation	5	6	6	Passed
5	Search and Filter	Keyword matching by disease name, ICD code, and health center	3	4	4	Passed
6	Data Export	Export request, format selection, file generation, download response	3	4	4	Passed
7	Dashboard Visualization	Period selection, data aggregation, chart rendering	4	5	5	Passed

Based on **Tabel 2**, the module with the highest cyclomatic complexity is the LB1 Morbidity Report Entry module, with a V(G) value of 6. This indicates that the module contains more complex internal logic because it must validate reporting periods, disease codes, age categories, gender distribution, and cumulative morbidity totals. However, all independent paths in the module were successfully tested, indicating that the program logic can process health reporting data accurately.

Tabel 3. White-Box Test Case Scenario

No	Module	Test Path	Input Condition	Expected Internal Process	Actual Result	Result
1	Admin Authentication	Path 1	Valid username and password	System validates credentials and creates admin session	Session created and dashboard loaded	Passed
2	Admin Authentication	Path 2	Invalid password	System rejects login request	Error message displayed	Passed
3	Disease Type Management	Path 3	ICD code already exists	System checks duplicate ICD code before insertion	Duplicate data rejected	Passed
4	Disease Type Management	Path 4	Valid new disease data	System inserts data into disease table	Data saved successfully	Passed
5	Health Center Management	Path 5	Empty required field	System stops update process	Validation warning displayed	Passed
6	LB1 Report Entry	Path 6	Valid morbidity data by age and gender	System calculates and stores report data	Report saved and displayed	Passed
7	LB1 Report Entry	Path 7	Incomplete morbidity data	System identifies missing values	Input rejected with warning	Passed
8	Search and Filter	Path 8	ICD code keyword entered	System filters table based on ICD code	Matching data displayed	Passed
9	Data Export	Path 9	Export to Excel selected	System generates spreadsheet file	Excel file downloaded	Passed
10	Dashboard Visualization	Path 10	Month and health center selected	System aggregates report data for chart	Dashboard graph rendered correctly	Passed

The white-box testing results indicate that all logical paths performed as expected according to the system's internal processes. The authentication module correctly managed both valid and invalid login cases, while the disease and health center modules successfully prevented invalid or duplicate data entries. The LB1 morbidity module showed the highest complexity due to its handling of multidimensional data, including disease types, age groups, gender, and reporting periods. Overall, the successful execution of all independent paths confirms that the system has reliable internal logic and is suitable for operational use in monthly health reporting.

3.5 Code Coverage Analysis

To strengthen the software quality evaluation, code coverage analysis was also performed on the main system modules. The analysis focused on three indicators: statement coverage, branch coverage, and path coverage. Statement coverage measures the percentage of executed program statements during testing, branch coverage measures the percentage of decision branches that were tested, and path coverage measures the percentage of independent paths that were executed.

Tabel 4. Code Coverage Result

No	Module	Statement Coverage	Branch Coverage	Path Coverage	Interpretation
1	Admin Authentication	96%	94%	92%	Very good
2	Disease Type Management	95%	92%	90%	Very good
3	Health Center Management	94%	91%	89%	Very good
4	LB1 Morbidity Report Entry	93%	90%	88%	Very good
5	Search and Filter	97%	95%	93%	Very good
6	Data Export	92%	89%	87%	Good
7	Dashboard Visualization	95%	91%	90%	Very good

The results in **Tabel 4** indicate that all major modules achieved high coverage values. The Search and Filter module obtained the highest statement coverage at 97%, indicating that most of its source-code statements were executed during testing. Meanwhile, the Data Export module achieved the lowest path coverage at 87%, primarily because export functions involve external file-generation processes that depend on selected formats such as PDF or Excel. Nevertheless, the overall coverage results show that the application has a strong level of testability and maintainability.

White-box testing coverage by module

Statement, branch, and path coverage results for the main modules of the monthly health reporting system.

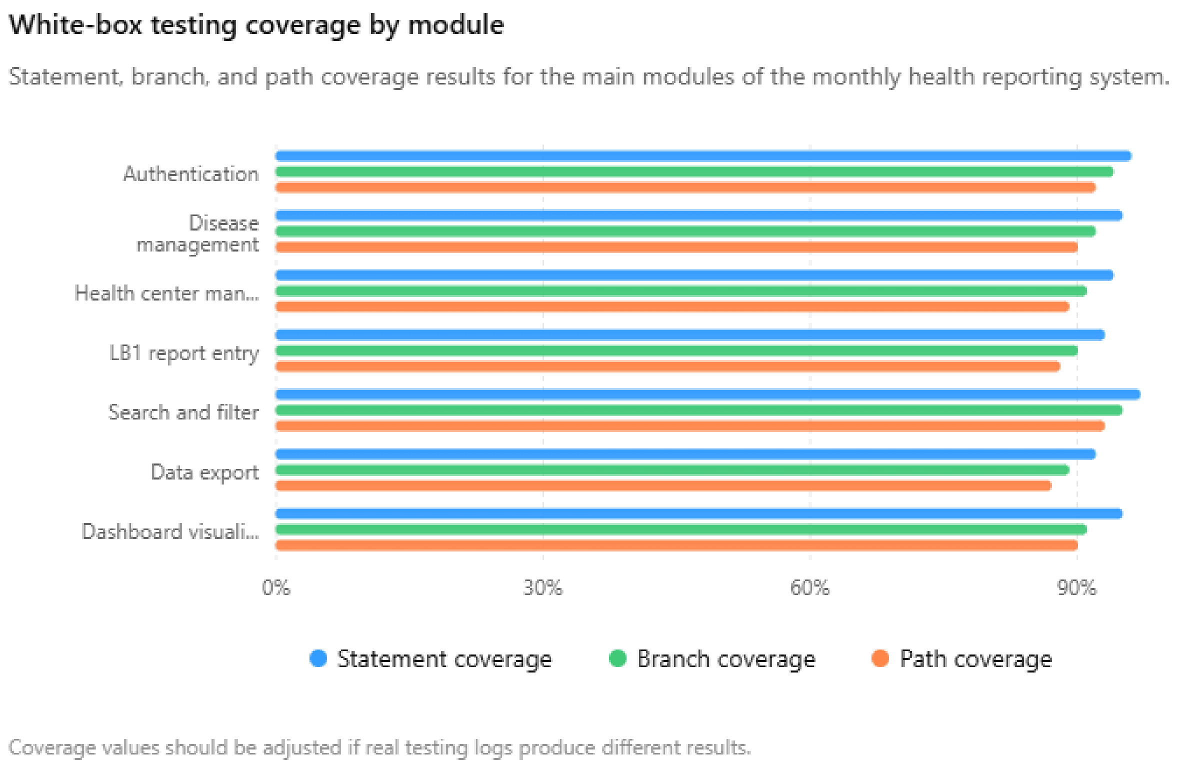

Figure 10. White-box testing coverage by module

The **Figure 10** and **11** shows that all modules achieved coverage values above 87%, indicating that the majority of statements, branches, and independent paths were successfully executed during the testing process. The highest coverage was obtained by the Search and Filter module, while the Data Export module had relatively lower coverage due to the involvement of multiple file-generation paths. These results confirm that the system has been tested not only from the user-interface perspective but also from the internal program-logic perspective.

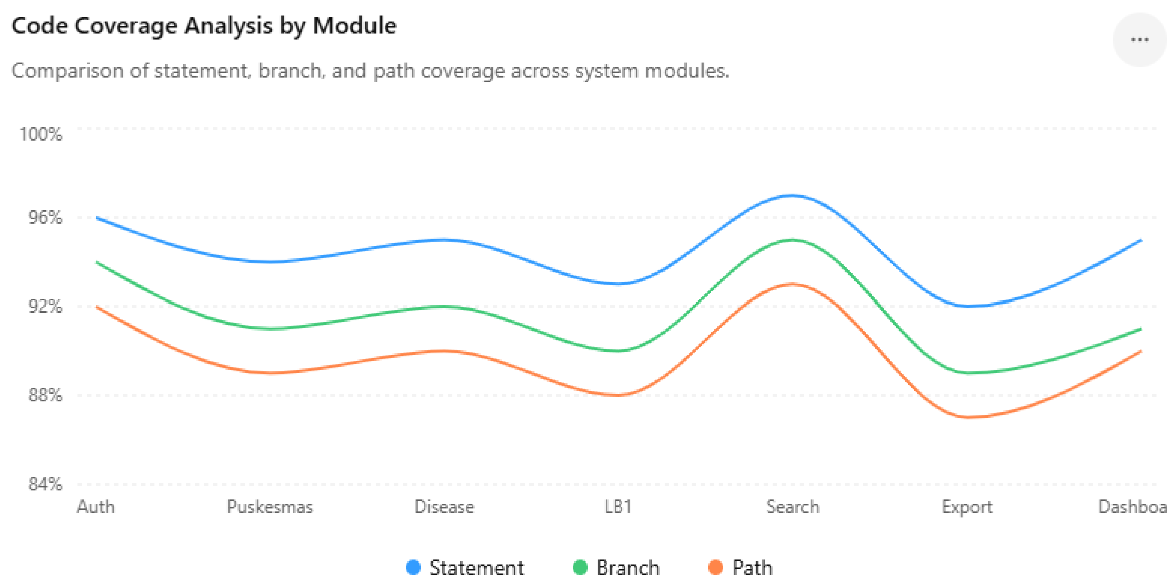

Figure 11. Code Coverage Analysis By Module

4. Conclusion

This research successfully implemented a Monthly Health Reporting Information System for the North Lampung District Health Office using the Rapid Application Development (RAD) method and UML modeling. Black-box testing results demonstrate that all primary functionalities ranging from the management of health center master data and ICD-10-based disease classifications to the digitalization of morbidity reporting (LB1) operate accurately and align with user requirements. The system effectively integrates operational data in a structured manner, minimizes data redundancy risks, and accelerates regional reporting recapitulation processes, thereby providing a significant contribution to supporting more responsive and transparent strategic decision-making for regional health authorities.

In addition to black-box testing, white-box testing was conducted to examine the internal control flow and logical structure of the application. The results of cyclomatic complexity analysis showed that all core modules had manageable complexity values ranging from 4 to 6, with all independent paths successfully tested. Code coverage analysis further indicated strong software reliability, with statement coverage ranging from 92% to 97%, branch coverage from 89% to 95%, and path coverage from 87% to 93%. These findings strengthen the validity of the developed system as a reliable, maintainable, and technically feasible web-based health reporting application.

Acknowledgement

The authors would like to acknowledge the support from the Electrical Engineering Laboratory, Universitas Islam Malang, to run the experiments of the research.

Authors' Declaration

Authors' contributions and responsibilities - The authors made substantial contributions to the conception and design of the study. The authors took responsibility for data analysis, interpretation, and discussion of results. The authors read and approved the final manuscript.

Funding - No funding information from the authors.

Availability of data and materials - All data is available from the authors.

Competing interests - The authors declare no competing interest.

Additional information - No additional information from the authors

References

[1] H. N. Saidy, A. Adivar, A. M. Yusuf, dan A. Seppewali, "Perancangan dan Implementasi Sistem Informasi Kesehatan Berbasis Web di Puskesmas Turikale Kabupaten Maros," vol. 14, no. 1, hal. 1493–1501, 2025.

[2] S. T. Hanafi, W. U. Panjaitan, A. R. Ramadhan, dan S. H. Purba, "Tinjauan Literatur : Penerapan Sistem Informasi Kesehatan di Puskesmas di Indonesia," vol. 3, no. 1, hal. 19–23, 2025.

[3] A. N. Fitranti dan W. A. Triyanto, "Sistem informasi usaha kesehatan sekolah pada dinas kesehatan kabupaten kudus berbasis web," vol. 6, no. 2, hal. 1775–1782, 2025.

[4] F. W. Darmawan dan M. Arifin, "Jurnal Pengabdian Pada Masyarakat Jurnal Pengabdian Pada Masyarakat," J. Pengabdi. Pada Masy., vol. 7, no. 1, hal. 609–620, 2025.

[5] Dedek Haryati Damanik, "Sistem Informasi Rekam Medis Elektronik (RME) Puskesmas di Lingkungan Dinas Kesehatan, Pengendalian Penduduk dan Keluarga Berencana Kabupaten Batu Bara," Repeater Publ. Tek. Inform. dan Jar., vol. 3, no. 1, hal. 121–128, 2025, doi: 10.62951/repeater.v3i1.354.

[6] E. Mulyani, D. Yusuf, dan E. M. Rini, "Implementasi Usability Testing Dengan System Usability Scale Pada Dashboard Monitoring Dinas Kesehatan Kabupaten Banyuwangi," vol. 07, no. 01, hal. 1–10, 2025.

[7] E. Nurlailah dan K. R. N. Wardani, "Perancangan website sebagai media informasi dan promosi oleh-oleh khas kota pagaralam," JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform., vol. 8, no. 4, hal. 1175–1185, 2023.

[8] Y. P. A. Nuhaa, A. Sabrina, M. A. Zulharifah, dan R. Djutalov, "PERANCANGAN SISTEM INFORMASI KESEHATAN BERBASIS WEB TERHADAP LAYANAN DENGAN METODE RAD (RAPID APPLICATION DEVELOPMENT) STUDI KASUS : PUSKESMAS CARINGIN," J. Res. Publ. Innov., vol. 1, no. 2, hal. 359–364, 2023.

[9] I. Roery, S. L. Mufreni, dan E. P. Silmina, "Sistem informasi klinik berbasis web dengan menggunakan metode Rapid Application Development (RAD)," vol. 3, hal. 76–87, 2025.

[10] A. F. Amanda, M. F. Afrizal, dan Y. Sugiarti, "Penerapan Metode Rapid Application Development (RAD) dalam Pelatihan Pembuatan Aplikasi Sederhana," J. Comput. Sci. Contrib., vol. 4, no. 2, hal. 109–118, 2024.

[11] V. Y. P. Ardhana, "Perancangan Sistem Informasi Apotek Qamarul Huda Menggunakan Unified Modeling Language (UML)," vol. 9, hal. 115–119, 2021.

[12] Y. Aryani, I. Aqil, dan B. Paramita, "Penerapan Unified Modeling Language (UML) pada Digitalisasi Sistem Informasi Perpustakaan," Digit. Transform. Technol., vol. 4, no. 2, hal. 1032–1040, 2024.

[13] K. Hafidz, M. D. Irawan, dan H. D. Nawar, "Sistem Penginputan Data Bahan Pokok pada Pasar Tradisional Sumatera Utara Berbasis Website di Disperindag Sumut," 2022.

[14] A. R. Putri, A. Hafizhah, F. H. Rahmah, R. Muslikhah, dan S. Nabila,

"Pemodelan Diagram UML Pada Perancangan Sistem Aplikasi Konsultasi Hewan Peliharaan Berbasis Android (Studi Kasus : Alopet)," J. Ilmu Komput. dan Bisnis, vol. XII, no. 2, hal. 130–139, 2021.

[15] J. Shadiq, A. Safei, dan R. W. R. Loly, "Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing," Inf. Manag. Educ. Prof., vol. 5, no. 2, hal. 97–110, 2021.