DOI: https://doi.org/10.70656/ijcse.v2i01.495

Design of a Medical Record Management Information System Based on Long-Term Storage for Hospital Efficiency Using V-Model

1st Yuda Syahidin
Health Information Management Study Program
Piksi Ganesha Polytechnic
Bandung, Indonesia
mail4yudasy@gmail.com

2nd Rena Novia Dewi
Health Information Management Study Program
Piksi Ganesha Polytechnic
Bandung, Indonesia
renalucu1@gmail.com

3rd Sali Setiatin
Medical Records and Health Information
Piksi Ganesha Polytechnic
Bandung, Indonesia
salisetiatin@gmail.com

Abstract*—This study aims to design an information system for the long-term storage of outpatient medical records in hospitals, focusing on improving data management efficiency, especially in retrieving inactive records. The research method used is descriptive with a qualitative approach. Data collection techniques include interviews, observations, and literature reviews. The software development process adopts the V-Model method, which emphasizes verification and validation at each development stage. The main problem identified in the long-term storage process is the manual classification of active and inactive records, which is done by checking the patient's last visit date one by one. This manual process is considered inefficient, time-consuming, and potentially disruptive to effective medical record management. To address this issue, the study proposes the design of an application that systematically manages inactive medical records in a computerized manner. This system is expected to help hospitals optimize time and resources, as well as support digital-based data storage policies. By implementing this information system, medical record management can become more structured, efficient, and accurate, ultimately enhancing the quality of health information services in hospitals.

Keywords- Medical Records, Information System, Long-Term Storage, Outpatient, V-Model

I. INTRODUCTION

Hospitals are healthcare service facilities that provide comprehensive individual health services, including inpatient care, outpatient care, and emergency medical services. This provision aligns with the Indonesian Law of the Republic of Indonesia Number 44 of 2009 on Hospitals, which stipulates that hospitals have a primary function in delivering comprehensive healthcare services to individuals. In addition, hospitals hold responsibility in supporting the national healthcare system, particularly as secondary and tertiary service providers that serve as referral centers for primary healthcare facilities and other related institutions [1], [2].

One of the essential components in the provision of healthcare services in hospitals is medical records. Medical records are documents that contain patient identification, anamnesis results, physical and supporting examinations, diagnosis, treatment, and medical procedures provided to the patient. These records are prepared by healthcare professionals, such as doctors or dentists, and must adhere to completeness and clarity standards in accordance with applicable regulations. Medical records can be in manual or electronic form and serve as legal and administrative documentation, as well as a basis for clinical decision-making in subsequent healthcare services. [3].

In general, medical records encompass all health service information received by a patient from admission to discharge, whether as an inpatient, outpatient, or emergency patient. This document serves as the primary reference when the patient returns for further treatment and must be managed in accordance with legal and administrative regulations. According to storage guidelines, medical records of inpatient patients must be kept for a minimum of five years from the date of the patient's last visit or discharge from the hospital or health center. After this period, the medical records may be destroyed, with the exception of discharge summaries and consent forms for medical procedures, which must be retained [1].

Research on medical record retention information systems has demonstrated that unstructured management of medical records can hinder the efficiency of healthcare services in hospitals. A study by Hikmah Ramadhanti et al. (2021) at Islamic Hospital Arafah Jambi revealed that the retention system was still carried out manually, with the separation of active and inactive files done individually, and data entry into computers was performed without system automation. Issues such as limited storage space and delays in the retention process have become major obstacles, directly affecting the quality of health information services. To address these challenges, the researchers designed a desktop-based retention information system using the Waterfall approach, leveraging Microsoft Visual Studio, with the aim of making the retention management more systematic, efficient, and in line with medical record management standards in hospitals [4].

A similar study conducted by Putri, Syahidin, and Hidayati (2022) at RSIA X also highlighted the limitations in the retention process, which still relied on Microsoft Excel and manual procedures for checking and classifying active and inactive medical records. This issue resulted in slow and inaccurate retention processes, especially due to the lack of a dedicated information system that integrates the retention workflow through to reporting. The system developed in this study was designed using a Waterfall approach, leveraging Microsoft Visual Studio and Microsoft Access as technical

platforms, and incorporating Data Flow Diagrams (DFD), context diagrams, and Entity Relationship Diagrams (ERD) in the design phase. The implementation of the system demonstrated an improvement in the effectiveness of retention and reporting processes, as well as enhanced ease of searching for archived medical records. [5].

At Jasa Kartini Hospital in Tasikmalaya, the retention system is still managed conventionally, resulting in prolonged classification processes and a high risk of record-keeping errors. The research conducted there developed a desktop-based information system using Visual Studio 2010 and Microsoft Access, which is equipped with reporting features and undergoes black-box testing to ensure optimal system functionality.

A key issue faced by many hospitals is the limited storage space for medical records. Over time, the number of medical records continues to increase, leading to document accumulation on storage shelves. Medical records are crucial archives containing data and information about patients who have received healthcare services. Therefore, their proper storage and organization are essential for the continuity of administrative operations and service delivery at hospitals and public health centers. Without an effective storage system, hospital administration cannot function as intended[6].

To address these limitations, hospitals need to implement retention activities or the reduction of inactive medical record files from active storage spaces. Retention is carried out by transferring files to other storage areas or converting them into other formats, such as microfilm. For retention to be effective, it is essential to establish a retention schedule and an integrated, documented system for managing inactive medical records. However, in practice, many hospitals still use manual retention systems and lack standardized retention schedules, resulting in file accumulation, difficulties in document retrieval, and a shortage of space for new files.

Therefore, a technology-based solution is needed to assist hospitals in the retention and long-term management of medical record files. The designed information system should be capable of supporting automatic classification between active and inactive files, organizing retention schedules, and providing quick and accurate document search functions. The implementation of this system will not only enhance space and time efficiency but also improve file organization and support the digital transformation of health information management in healthcare facilities.

II. METHODOLOGY

A. Software Methodology

The Software Development Life Cycle (SDLC) is a systematic process used in the development and modification of software systems. This process employs specific models and approaches that have been tested and proven to produce high-quality software systems. It adheres to methods that have demonstrated effectiveness through established best practices [7].

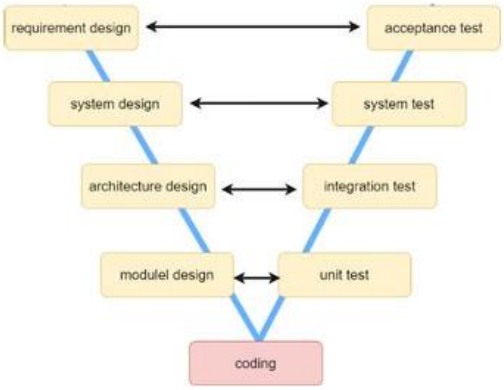

Figure 1. V-Model[7]

Phases of the V-Model

1. Requirement Analysis

This phase is the initial process for understanding and identifying the user's needs for the system to be developed. Information is gathered through interviews, observations, or document studies to identify both functional and non-functional problems and requirements. The result of this phase is the user requirements document, which serves as the foundation for the further system design.

2. System Design

In this phase, the overall system structure is designed based on the results of the requirement analysis. The development team determines how the system will operate, what modules are needed, and how system components will be integrated. This phase also defines the system boundaries, data specifications, and the main interfaces between the system and the user.

3. High-Level Design

High-level design focuses on identifying the main system modules, the data flow between modules, and the primary functions of each component. The goal is to create a framework for the system so that all components can interact effectively. Diagrams such as DFD Level 0 and ERD are commonly used to visually represent this design.

4. Low-Level Design

Low-level design provides technical details for each module, such as algorithms, data structures, programming logic, and user interface design. The goal is to ensure that programmers can implement the system accurately according to the design, and to facilitate the debugging and maintenance of the system in the future.

5. Coding / Implementation

This phase involves the actual process of building the system based on the previously established designs. Each module is developed using a specific programming language and is tested independently. The output of this phase is an initial system or application that can be technically and functionally tested.

6. Unit Testing

Unit testing is conducted to ensure that each module or component works as intended on an individual basis. This testing is performed by the programmer or development team to detect errors early before the modules are integrated into the larger system.

7. Integration Testing

After all modules are tested individually, the next phase is to test the integration between modules. This testing ensures that the data flow and communication between parts of the system function smoothly and as intended, as designed during the high-level design phase.

8. System Testing

System testing is performed on the entire integrated system to ensure that all components work according to the system's specifications. This process includes testing the system's functionality, performance, security, and overall reliability before it is delivered to the users.

9. Acceptance Testing

The final phase in the V-Model is acceptance testing, where end users test the system to verify that it meets all the agreed-upon requirements. This testing serves as the basis for the users to either accept or reject the system, and it is often used as a benchmark for the successful implementation of the system.

The selection of the V-Model methodology in this research is based on its advantages as a derivative of the Waterfall model, which emphasizes software quality through a systematic approach. This method supports communication, modeling, and construction processes from the beginning of the development cycle, allowing for parallel validation and verification at each stage. Thus, the V-Model is an optimal choice for ensuring the accuracy, completeness, and reliability of the system from the initial phase to the final implementation [7],[8].

III. Result And Discussion

A. System Process Analysis

The system analysis phase aims to identify and outline the specifications of the processes within the information system currently in use. This process is visualized using a use case diagram, as referenced in [9], [10], [11] The system design is a follow-up to the analysis that has been conducted, describing how a system is developed to become a solution to the issues identified during the analysis of the existing system [7], [8], [15], Figure 4 below illustrates the visual representation of the class diagram design for the developed system.

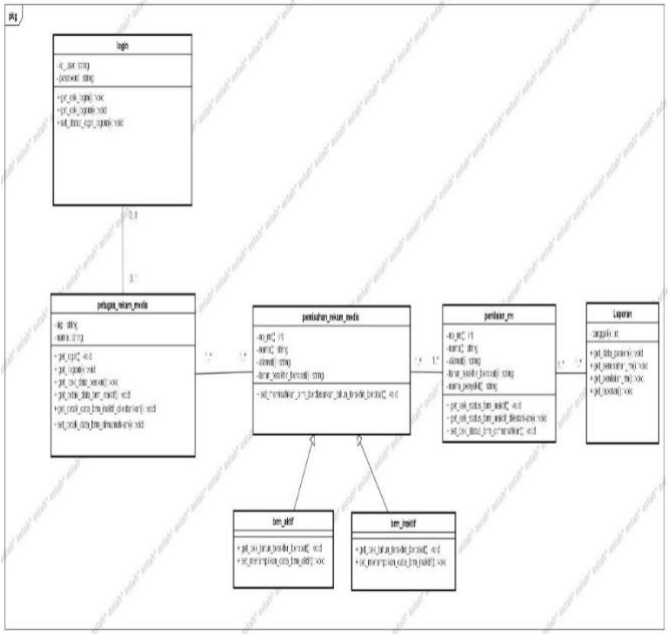

Figure 2. Use Case of the Retention Information System

Table 1. Class Descriptions in the Class Diagram for the Outpatient Medical Record Retention Information System

Class Name	Description
Login	Represents the user authentication process based on the login use case scenario. It is used to validate user access to the application system.
Medical Record Officer	Represents the entity responsible for managing medical record data. It is used to summarize information from the staff data.
Medical Record Separation	Responsible for classifying medical record files based on their active or inactive status for retention purposes.
Medical Record Assessment	Functions to assess the feasibility of inactive files, determining whether they still hold value for storage or should be destroyed.
Report	Used to generate retention data summaries in the form of reports, based on information from the previous processes

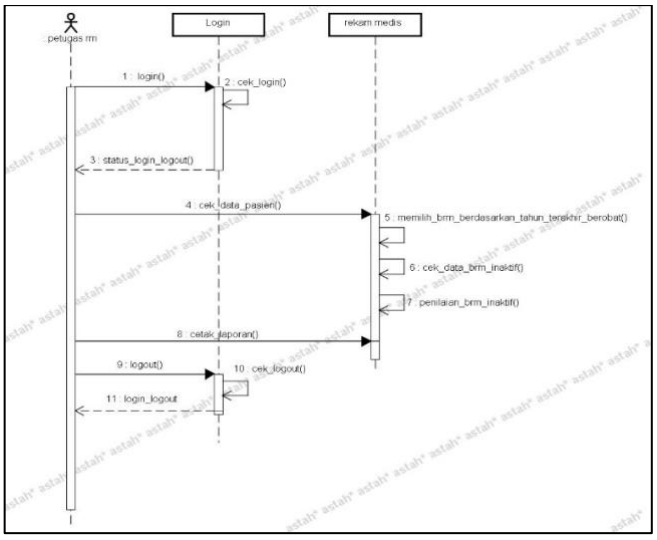

Figure 5. Sequence Diagram of the Long-Term Storage Information System

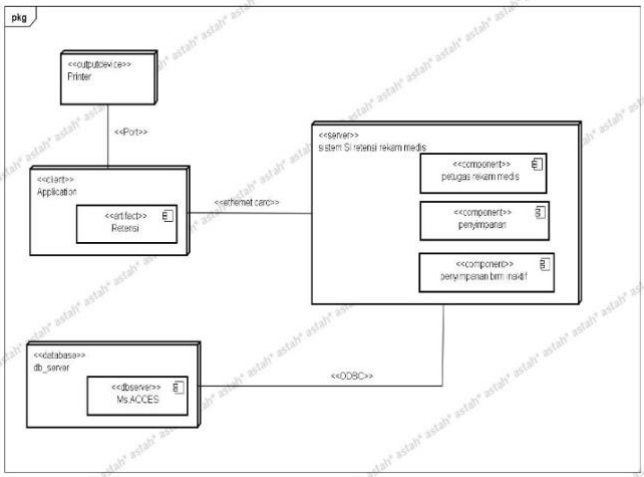

Figure 6. Deployment Diagram of the Long-Term Storage Information System

10. Table Mapping in System Design

The database design for the outpatient medical record retention information system consists of several primary tables that represent key entities within the system. The Staff Table stores information regarding the identities of the staff responsible for managing medical records, with the Employee Identification Number (NIP) serving as the primary key. This table includes attributes such as name, position, phone number, username, and password, all of which are of the text data type.

The Patient Registration Table serves to record patient data at their first visit, with the medical record number (No_rm) as the primary key. This table includes attributes such as name, gender, date of birth, age, address, occupation, payment method, responsible person, clinic, doctor's name, and admission date. Most of the data in this table are of the text data type, while the date of birth and admission date are stored as date and time data types.

Next, the Medical Record Table (RM) stores data regarding patient visits, including the medical record number, patient name, visit date, final diagnosis, attending doctor, file storage duration, and the medical record status (active/inactive). The primary key for this table is also No_rm.

The final table is the Assessment Table, which contains information regarding the evaluation of medical record files. The structure of this table is like the RM Table, but it includes an additional remarks column to provide supplementary notes on the retention assessment results. Like the others, No_rm is used as the primary key. This design supports the system's needs by efficiently managing data related to staff, patients, visit processes, and the assessment of the medical record status in a structured manner.

B. User Interface Design

This section presents the results of the implementation of the interface design previously created in the form of display layouts. The following screenshots showcase the user interface of the outpatient medical record retention information system as a visualization of the functionality that has been designed.

1. Login

Figure 7. Login

The image above displays the Login Form interface of the outpatient medical record retention information system. This interface serves as the entry point for users who wish to access the system, providing two main input fields: Username and Password. After the user enters their credentials, the Login button is used to verify the information and proceed to the system's main page, while the Cancel button functions to abort the login process. The visual design of the background, featuring a digital padlock icon, reflects a strong emphasis on data security and restricted user access.

2. Main Menu

The image above illustrates the main page (dashboard) of the Medical Record Retention Information System application. This interface serves as the central navigation hub for users upon successful login to the system. At the top, the primary menu includes options such as Master Data, Reports, and Log Out, allowing users to access the system's key features, including staff and patient data management, retention assessment, and report generation. The visual design, featuring a health-related logo in the top-left corner,

establishes the system's identity as an application used within healthcare facilities. This interface is designed to be simple yet informative, ensuring that users can operate the system efficiently and effectively.

Figure 8. Main Menu

3. The Patient Information Input Interface

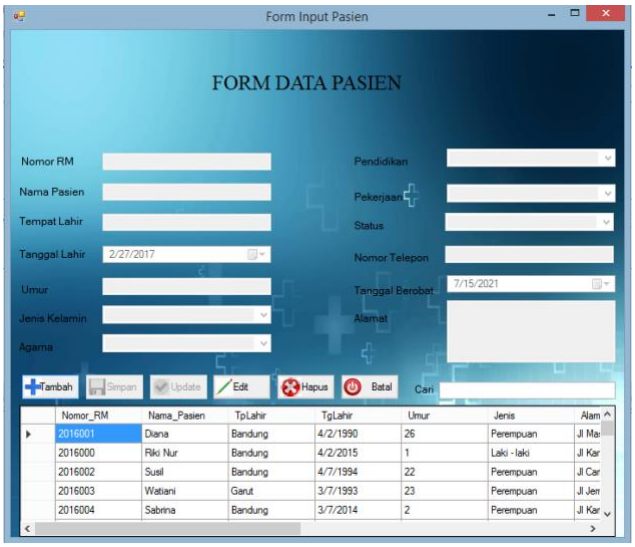

Figure 9. Patient Information

The image above depicts the Patient Input Form in the outpatient medical record retention information system. This form is designed to record and manage patient data, which includes information such as Medical Record Number, Patient's Name, Place and Date of Birth, Gender, Age, Religion, Education, Occupation, Marital Status, Phone Number, Date of Visit, and Address. At the bottom, there is a table displaying previously entered patient data. Users can perform actions such as adding, saving, updating, deleting, canceling, or searching for data using the available control buttons. This interface aims to streamline the management of patient data, making it more systematic and efficient for the staff.

4. Outpatient Medical Record Retention Form Display

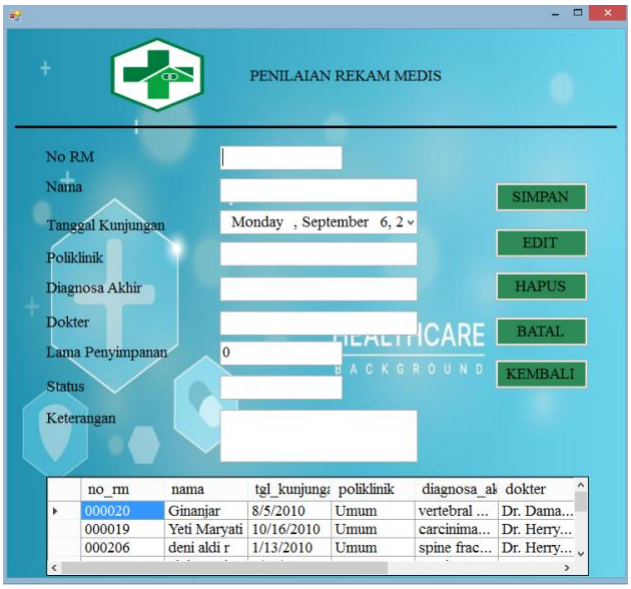

Figure 10. Evaluation of Outpatient Medical Records

The image above illustrates the Medical Record Assessment Form in the medical record retention information system for outpatient services. This form is used to evaluate the feasibility of storing patient medical records based on visit data. The information inputted includes the Medical Record Number, Patient Name, Visit Date, Polyclinic, Final Diagnosis, Doctor, Storage Duration, Medical Record Status (active or inactive), and Additional Notes. On the right side, control buttons such as Save, Edit, Delete, Cancel, and Back are provided to facilitate data management. Meanwhile, the bottom section displays a table of previously assessed patient data, making it easier for staff to review or update existing information. This interface supports the retention process in a systematic, efficient, and structured manner.

5. Long-Term Storage Reporting

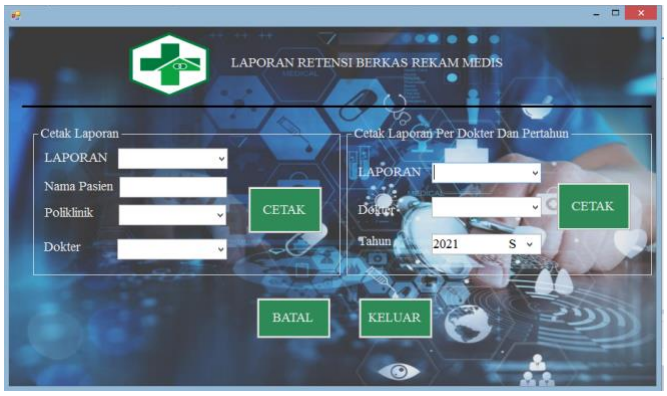

Figure 11. Reporting of Outpatient Medical Record Retention

The image above illustrates the Retention Report Form interface for medical record retention in the outpatient information system. This interface offers two generation options: one based on the patient's name, polyclinic, and doctor, and the other based on the doctor and the year of service. Users can select the desired report type via the dropdown menu, then click the PRINT button to generate the report according to the specified criteria. The CANCEL button is used to clear the input fields, while the EXIT button closes the form. This interface design is intended to streamline

the process for staff, enabling them to quickly, accurately, and systematically produce retention reports based on the selected parameters.

C. Software Performance Testing

The software testing of the outpatient medical record retention information system was conducted using the black box testing method, which focuses on evaluating the system's functionality without considering the internal structure of the software. The primary objective of this test was to ensure that each feature of the system operates as expected. Several components were tested, including the login process, data input in the officer form, patient form, doctor form, and polyclinic form. The test results indicated that when the input data was valid, the system successfully displayed a confirmation dialog and stored the data in the database. Conversely, when the data was invalid, the system displayed an error message, and the data was not saved, confirming that the validation feature is functioning as intended. Overall, all tests yielded acceptable results, leading to the conclusion that the system is functioning correctly according to the designed functional specifications.

Table 2. Login

Test	Purpose	Description
Login	Access the main menu	Succesfull
Add	Access the officer data form	Succesfull
Cancel	Cancel login	Succesfull
Exit	Exit from the login form	Succesfull

Table 3. Menu

Test	Purpose	Description
Master Data	Display form options	Succesfull
• Officer Data	Display officer data	Succesfull
• Patient Registration	Display patient registration	Succesfull
• Doctor Data	Display doctor data	Succesfull
• Visit Data	Display visit data	Succesfull

IV. CONCLUSION

This study has resulted in the development of an outpatient medical record retention information system designed to assist in the process of separating and evaluating inactive medical record files in a more structured manner. Utilizing the V-Model methodology in its development, the system was successfully built through a systematic series of phases, resulting in features that meet the specified requirements, including login, patient data input, staff and doctor management, as well as retention processes and reporting. Testing conducted using the black-box method indicates that the system functions properly, handling both correct and erroneous input effectively.

With the implementation of this system, the management of inactive medical records has become more efficient and easier to monitor. In the future, the system can be further developed, for instance, by adding an automatic search feature based on retention schedules, integrating with the hospital's electronic medical records (EMR) system, and incorporating cloud storage to support long-term digital storage. Additionally, direct field trials should be conducted to assess the system's effectiveness and user acceptance.

REFERENCES

[1] D. R. D. YanMed, *Pedoman Pengelolaan Rekam Medis Rumah Sakit di Indonesia, Jakarta : DepKes,*. Indonesia, 2006.

[2] K. R. Indonesia, "PERATURAN MENTERI KESEHATAN REPUBLIK INDONESIA NOMOR 24 TAHUN 2022 TENTANG REKAM MEDIS," 2022. doi: 10.1097/00006231-199705000-00098.

[3] Menteri Kesehatan RI, "Peraturan Menteri Kesehatan RI Nomor 269/MENKES/PER/III/2008, tentang Rekam Medis," 2008.

[4] H. Ramadhanti, Y. Syahidin, and D. Sonia, "Disain Sistem Informasi Retensi Rekam Medis Pasien Rawat Inap," *Explor. Sist. Inf. dan Telemat.*, vol. 12, no. 2, p. 90, 2021, doi: 10.36448/jsit.v12i2.2044.

[5] R. K. Putri, Y. Syahidin, and M. Hidayati, "Rancang Bangun Sistem Informasi Retensi Rekam Medis Di Rsia X," *J. SAINTEKOM*, vol. 12, no. 1, pp. 33–44, 2022, doi: 10.33020/saintekom.v12i1.226.

[6] R. Yusnia, S. Setiatin, W. Nadiroh, and C. Mecca Sufyana, "Perancangan Sistem Informasi Retensi Rekam Medis Pasien Rawat Inap Menggunakan Visual Studio 2010 di Rumah Sakit Jasa Kartini Tasikmalaya," 2021. doi: 10.46799/jhs.v2i8.252.

[7] R. Pressman, *Software Engineering - A Practitioners Approach*. New York: McGraw-Hill Education, 2015.

[8] Ian Gorton, *Essential Software Architecture Second Edition*. Springer-Verlag Berlin Heidelberg, 2011.

[9] K. Barclay and J. Savage, "Object-Oriented Design with UML and Java," p. 414, 2004.

[10] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, 2nd ed. Boston, MA: Addison-Wesley, 2005.

[11] B. Bruegge and A. H. Dutoit, *Object-Oriented Software Engineering Using UML, Patterns, and Java*, 3rd ed. Upper Saddle River, NJ: Pearson, 2009.

[12] bin A.-B. Ladjamudin, "Analisis dan Desain Sistem Informasi Edisi Kedua, Graha Ilmu, Yogyakarta.," 2013.

[13] J. Arlow and I. Neustadt, *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design*, 2nd ed. Boston, MA: Addison-Wesley, 2005.

[14] M. Blaha and J. Rumbaugh, *Object-Oriented Modeling and Design with UML*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2005.

[15] A. Pyster, "Guide to the Systems Engineering Body of Knowledge (SEBoK)," 2012, *The Trustees of the Stevens Institute of Technology*. [Online]. Available: https://www.sebokwiki.org/