International Journal Software Engineering and Computer Science (IJSECS)
5 (3), 2025, 1162-1177
Published Online December 2025 in IJSECS (http://www.journal.lembagakita.org/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v5i3.5229.

RESEARCH ARTICLE Open Access

Design of Recapitulation System Pending Inpatient Claims to Enhance BPJS Verification Efficiency Using Extreme Programming

Belinda Nuryania Prameswari *
Health Information Management, Politeknik Piksi Ganesha, Bandung City, West Java Province, Indonesia.
Corresponding Email: belindaprameswari93@gmail.com.

Falaah Abdussalaam
Health Information Management, Politeknik Piksi Ganesha, Bandung City, West Java Province, Indonesia.

Yuyun Yunengsih
Health Information Management, Politeknik Piksi Ganesha, Bandung City, West Java Province, Indonesia.

Received: August 1, 2025; Accepted: November 15, 2025; Published: December 1, 2025.

Abstract: This study designs and develops an information system with a centralized database for recapitulating pending inpatient claims at Hospital X Bandung. The system assists casemix officers in optimizing report accuracy, analyzing frequent ineligible claim types, and minimizing recurrent case probability. Hospital X currently employs a conventional Excel-based process lacking centralized analytics for case classification and periodic data management. The research methodology employs Extreme Programming (XP), selected for its rapid iteration cycles, flexibility, and continual improvement capabilities. Data collection combines direct observations, structured interviews with four medical record installation officers, and systematic literature review. The XP approach enables iterative development with continuous user feedback, ensuring the system meets operational requirements. Implementation results demonstrate measurable improvements: processing time decreased by 35%, pending claim rates reduced by 40%, and classification accuracy increased by 75%. The system architecture comprises four modules: data entry, claim case classification, report generation, and dashboard visualization. Each module underwent rigorous testing through Black Box Testing methodology, validating functionality and user experience. Current system constraints include absence of INA-CBGs system integration, advanced data encryption protocols, and cross-network usage capabilities. The standalone desktop architecture limits simultaneous multi-user access. Despite these constraints, the system successfully simplifies recapitulation workflows and establishes a foundation for future development, particularly web-based architecture and predictive analytics integration.

Keywords: Information System; Pending Claims Recapitulation; Extreme Programming; BPJS Verification; Inpatient Claims Management.

1. Introduction

A hospital is a healthcare institution that provides comprehensive individual healthcare services, including inpatient care, outpatient care, and emergency care [1]. The new paradigm of healthcare requires hospitals to provide quality services according to the needs and desires of patients while still referring to the professional code of ethics [2]. In the rapid development of technology and intensifying competition, hospitals must consistently enhance the quality of their services. Medical record services constitute a standard of quality in hospitals. Medical records are vital and confidential documents that contain patient identity information, examinations, treatments, procedures, and other services rendered to patients. Medical records must be made and filled in completely, precisely, and accurately both in writing and electronically [3]. The data can be processed into useful information for hospital management internally and externally, such as in health financing, because it can be used as evidence of a series of services that have been provided to patients.

Badan Penyelenggara Jaminan Sosial (BPJS Kesehatan) is a type of healthcare insurance financing in Indonesia. As a public legal entity, BPJS is responsible for financing health services according to the provisions of the Health Insurance Program Regulation, which governs the administration of health facility claims related to the implementation of health insurance. The duties of BPJS Kesehatan are associated with Presidential Regulation Number 82 of 2018 concerning Health Insurance, which states that the payment of health services for the National Health Insurance program implemented by BPJS Kesehatan to Advanced Referral Health Facilities (FKRTL) is carried out using the Indonesian Case Base Groups (INA-CBG) method. The INA-CBG system provides a separate application for inputting claim files using a prospective payment system to pay claims to FKRTL [4]. Since 2008, the Public Health Insurance program in Indonesia has used a prospective payment method known as casemix (case-based payment). The casemix system uses grouper software (E-Klaim) to group diagnoses and procedures with similar clinical characteristics and use of similar resources or costs of care [5]. The process of submitting a stage one claim starts from codifying the patient's diagnosis according to the International Classification of Diseases 10th (ICD-10) and the type of patient procedures according to the International Classification of Disease Clinical Modification 9th (ICD-9 CM) [6].

Advanced Referral Health Facilities must complete claim administration requirements before submitting claims to BPJS and submit claims regularly no later than the 10th of the following month [7]. This activity must be done regularly and on time per procedure and regulation. Delays in the submission and payment of claims by both parties can adversely affect the cash flow of hospitals. Disruption of operations may diminish hospital motivation to deliver optimal service [8]. Upon completion of the entry and codification process via the INA-CBG system, a claim verification process is conducted to assess the accuracy of the service administration accountability finalized by the health facility [9]. The result of the stage one submission is the claim status. There are three different types of status after the claim file is received by the insurer, namely eligible claim, ineligible claim, and dispute. The status is declared ineligible, also called pending claims, because the claim file submitted by the hospital is incomplete [10]. When this happens, casemix management will enter the process of submitting stage two claims, which is a further submission of claim file documents that in the stage one claim process have not been submitted, unfinal, included in the Data Does Not Match (DTS) category, or are pending (revised). Stage two claims must be submitted collectively, periodically, and completely within a maximum of H-2 months before the claim expires [11]. If the time period for submitting the claim is exceeded, the claim cannot be resubmitted.

Previous studies conducted at Hospital X Bandung City showed the findings of problematic claims based on BPJS Health official report, there were 226 pending claim files from the total files submitted in January-March, namely 3,221 inpatient claim files in 2024 [12], indicating administrative claim inefficiencies. This irregularity negatively impacts hospital cash flow, increases the casemix team's workload, may cause a backlog, and possibly affects BPJS Kesehatan patients, who may encounter obstacles to receiving healthcare at hospitals due to financial limitations. To reduce claim ineligibility and prevent recurrence cases, the cause must be known. Hospital management must monitor and evaluate pending files. The pending claims recapitulation system may improve case analysis and future submission efficiency. Previous research explained that the reporting system of claims using conventional methods has potential weaknesses in the level of accuracy of the reports generated [13]. This study involved the design of an outpatient BPJS claim recapitulation information system, with the main output being pending claim statuses. Prior researchers utilized a Prototype method for system development, which illustrates the preliminary model and evaluates the system concept based on user feedback. However, this method can be time-consuming and may lead to insufficient documentation if not properly managed. The current study focuses on designing an information system to recapitulate pending inpatient claims with results in multiple levels of granularity to meet user reporting requirements. This approach employs Extreme Programming, which is more suitable for flexible phased development, organized iterative practices, and rapid adjustment to changing needs.

The quality of the health information system can be assessed by the completeness, accuracy, and currency of the data, which is crucial in the context of efficient management of health claims [14]. Research

at Hospital X Bandung City's inpatient installation found that Microsoft Excel is still used to recapitulate ineligible claims. The system has many limitations. Excel may be inefficient and error-prone. Excel also has inadequate security, is vulnerable to loss or unauthorized access, stores pending claims data separately, and cannot handle large data volumes like inpatient claims data, which is increasing. It also limits decision-making analysis. This inpatient claim recapitulation system was developed to improve pending claim data management. This study aims to centralize databases, recap pending inpatient claims, automatically categorize cause types, and provide hospital-required results. Based on these objectives, this study specifically addresses the following research question: How can the designed inpatient claim recapitulation information system improve the efficiency of BPJS claim verification and the accuracy of data compared to the manual method? This study examines the hypothesis that the proposed integrated information system will significantly enhance verification speed and data accuracy compared to manual verification.

2. Related Work

Various studies have examined the use of information systems in research on digital BPJS health claim management in Indonesia, particularly in efforts to understand the causes of pending claims and design digital solutions that can improve verification efficiency. In a study by Ulil Amri and Nurwahyuni (2024) [15], the main issues identified in pending claims at hospitals include incomplete medical records (29%) and coding errors (21%) in the 2022 JKN post-claim audit. To support claims management, Wijayanti *et al*. (2023) developed a POAC-based strategy that focuses on cross-departmental coordination and written SOPs [16]. Despite the workflow becoming more structured, it lacked a cohesive data model for root cause categorization, and the reporting is still limited to a general summary without case-specific details. Furthermore, Pratiwi *et al*. (2023) [13] designed an information system for reporting BPJS outpatient claims that generates automatic reports on eligible and ineligible claims along with specific reasons for claim rejections. This system has been proven to speed up reporting. However, the data model is limited to outpatient care, and reporting is aggregate per period, so it is unable to examine patterns of pending claims' causes.

In health software development, Agile methods have begun to be used, but the adoption of Extreme Programming (XP) is still rare. In a study by Lesmana and Taufik (2023) [17], a BPJS hospital claim monitoring system was developed using XP, and the implementation resulted in enhanced efficiency in development iterations and user communication. The system's data model includes patient, claim, and status entities but not the claim cause classification table. The reporting is limited to real-time claim data without analysis or automated root cause grouping for pending claims. The study by Chaturvedi *et al*. (2024), found that Gradient Boosted Trees can predict claim rejections with AUC = 0.91 and F1-score = 0.73, reducing rejections by 25% and rework costs by 15% in six months. This demonstrates how predictive approaches in recapitulation systems help verifiers anticipate problematic claims [18]. This work had a rich data model with high-granularity claim-level data and effective root cause grouping, but it was not designed for BPJS.

Table 1. Comparative Summary

Study	Data Model Depth	Reporting Granularity	Root Cause Grouping	Limitations	Gap Filled by This Study
Ulil Amri & Nurwahyuni (2024)	Low – identify common causes	Audit summary	Yes, manual	No system or structured data model	Develop integrated data model with automated classification
Wijayanti *et al*. (2023)	Low – process & department entities	Monthly summary	None	No automated cause analysis	Integrate automated root cause grouping
Pratiwi *et al*. (2023)	Medium – eligible/ineligible claims	Period aggregate	None	Outpatient focus; no analysis of pending cases	Reporting and cause classification for pending inpatients
Lesmana & Taufik (2023)	Medium – patient, claim, status entities	Real-time dashboard	None	No pending cause classification & analysis module	Root cause grouping & integrated reporting
Chaturvedi *et al*. (2024)	High – individual claim & prediction variable	Claim-level details	Yes	Not BPJS-specific	Adaptation to BPJS & integration of INA-CBG structure

Based on this comparison, while previous research has provided valuable results for BPJS claim management, no study has combined the Extreme Programming methodology with a rich relational data model, claim-level reporting granularity, or automatic classification based on case types specifically within the context of BPJS inpatient claims. This study addresses that gap and accommodates a rapid and iterative system design capable of supporting both operational efficiency and potential predictive analytics integration, as well as presenting dynamic reporting based on claim filters, which has not been widely applied in previous studies.

3. Research Method

3.1 Data Collection Methods

3.1.1 Research Design

The research method used is a qualitative research method with a descriptive approach to describe in depth the processes, problems, and needs in managing the recapitulation of inpatient claims. Qualitative research aims to provide a precise depiction of the current situation or symptoms, employing a method that investigates subjects in their natural environments, with the researcher serving as the primary instrument [19]. This approach was chosen because it is able to provide a comprehensive understanding of the phenomena that occur in the field, as well as identify system needs that are relevant to the hospital. In this study, the collection of information needed uses three main techniques, namely interviews, observation, and literature review. The collection of interview and observational data was conducted concurrently when carrying out Field Work Practice (PKL) for six months at Hospital X in Bandung City.

3.1.2 Population and Sample

The study population included casemix management officers and IT staff (SIMRS team) in the hospital that was the object of the study. The sample was determined using the purposive sampling technique, namely the selection of informants based on their role and involvement in the claims management process. Informant selection criteria include: (1) directly involved in the claim verification process; (2) willing to participate in interviews and research observations; and (3) actively involved in the use of the Hospital Management Information System (SIMRS). The number of informants interviewed consisted of one internal claims verification officer, one filing officer, one head of the claims management unit (casemix coordinator), and one staff member of the SIMRS team.

3.1.3 Data Collection Procedures

An interview is one of the data collection processes carried out by meeting with sources in order to obtain information by querying them with several questions that are related to the information that users require from the system [20]. At this stage, the researchers conducted interviews through verbal questions and answers with medical record installation officers, especially in the casemix section, who fulfill the informant criteria, with the aim of obtaining in-depth qualitative data related to workflows, constraints, and expectations of the system. The observation process relies on in-depth observation of the work system on the BPJS claim process, accompanied by problem analysis and phenomenology, which the researcher then systematically noted as related to the identified issues, as well as to obtain a comprehensive understanding and receive an overview of the requirements and necessary elements of the system to be developed. Literature review is the process that involves the analysis of literary works and the collection of data pertaining to the same topic as the object of the research is discussed, the purpose is aimed at acquiring information that supports research work [21]. During the process, the researcher substantiated the data with the results of a comprehensive literature review on the same topic or the studies that related to the object of research and hospital claims management standards.

3.1.4 Research Instruments

The research instruments used include an interview guide containing a list of open-ended questions based on the research indicators, an observation sheet to record activities, process flow, and obstacles observed, and documentation to record supporting data such as claim reports, SOPs, and hospital reporting formats.

3.1.5 Data Analysis Technique

Data were analyzed using thematic analysis, which included data reduction, data presentation, and conclusion drawing. Data from interviews were transcribed verbatim, coded, then categorized into themes relevant to the research objectives. Observation and documentation data were used as supporting triangulation.

3.1.6 Validity and Reliability

Data validity was maintained through source and method triangulation, comparing interviews, observations, and documentation to ensure the consistency of information. Member checking was done by asking for confirmation from informants regarding the summary of the interview results. The reliability was enhanced by employing a consistent interview guide for all informants and systematically documenting the entire process in the research log.

3.2 System Development Method

Extreme Programming (XP) is an approach in software development that emphasizes short iterations, intense communication, and continuous code improvement through practices such as refactoring and testing [22]. Based on the basic principles of XP, this method presents an active implementation of software components immediately and consistently applies changes as requests arise [23]. The selection of XP was based on the characteristics of the project that required high flexibility, rapid iteration, and intensive involvement of end-users during the development process. This method is relevant for the inpatient pending claims recapitulation system because XP emphasizes high adaptivity to change of needs, quick feedback from field users (internal verifiers, casemix coordinators, SIMRS officers, and filing officers), and phased releases that can be tested in real work environments, which are relevant to the needs of a pending inpatient claim recapitulation system that must be adaptable to changes in hospital procedures and BPJS Health policies. When compared to other methods such as Waterfall, Prototype, or Spiral, XP is more effective in being flexible to change (improvements are made every iteration), encourages active user involvement in every cycle, and has continuous testing that guarantees quality from the very beginning of development. Waterfall methods are linear, making it difficult to adjust to changing needs mid-process. Additionally, it is considered outdated and less relevant, while Prototype is flexible in the early stages but lacks emphasis on long-term code quality. More modern methods such as Spiral are effective at systematic risk management and flexible for large projects but tend to be more complex, expensive, and time-consuming.

3.2.1 Iteration Format and Cycle

The main stages of software development with XP include planning, design, coding, and testing. This cycle, which includes the release and refactoring phase, is repeated iteratively over a short duration (1-2 weeks), with direct feedback from users to ensure that each version produced meets actual needs. At each iteration, user requirements are converted into user stories with the format: (1) As Internal Hospital Verifier, I want to input, correct, and validate pending inpatient claims data, so that I can ensure that claims submitted to BPJS are in accordance with applicable documents and rules; (2) As Casemix Officer, I want to categorize claims by case type, pending status, and service period, so that I can compile claim recapitulations quickly without having to do manual calculations; and (3) As Casemix Coordinator, I want to access real-time dashboard reports and analysis of pending claims cases, so that I can monitor the cause of the case and make strategic decisions for efficient claims verification. Each user story was accompanied by measurable acceptance criteria: the system allows efficient input and management of pending claims data; pending claims data can be filtered and recapitulated by period, status, and case type; and automatic categorization of case types without manual calculation. Figure 1 shows an example of the life cycle model of Extreme Programming.

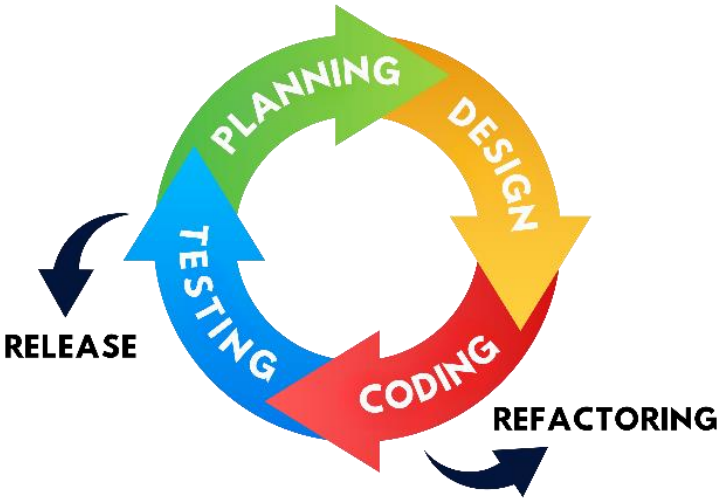

Figure 1. Extreme Programming Method

3.2.2 Planning

This stage includes a number of activities, such as identifying problems and assessing user and system needs, as well as setting a system development implementation schedule [24]. In this study, the authors used an interview and direct observation approach to analyze the needs, problems, and objectives of the software system development, as well as determine the limitations of the designed system. The results of the interviews were analyzed to generate a product backlog.

3.2.3 Design

Based on the results of the needs analysis that has been determined, the next step is to plan the system design and interface of the pending claims recapitulation system using the Microsoft Visual Studio 2012 application. The system design includes interface design and application architecture and is supported by a database designed through an Entity Relationship Diagram (ERD) with the main entities of Claims, Users, and Recapitulation. The initial prototype was created using Microsoft Visual Studio 2012.

3.2.4 Coding

The coding stage is the process of translating the design into a programming language that the computer recognizes. In this stage, the input is carried out using the Visual Basic .NET programming language with the use of Microsoft Access as DBMS. Refactoring is done at the end of each iteration to improve the code structure. It involves breaking the long code into smaller functions, eliminating duplicate logic, and giving meaningful variable names so that the code is easier to maintain.

3.2.5 Testing

This last stage is used to test the performance of the system so that it is known whether the system has met user expectations [25]. The test in this step focuses on each feature and functionality of the system, then reviewed by the user as a whole. The method used in testing is Unit Testing to ensure each module functions as designed, Black Box Testing to test features from the end-user side to identify whether the built program has been successful or there are still errors in the program design, and UI Testing to ensure the appearance, navigation, and interaction of the interface are easy to use and according to user needs. The testing scope includes the system's conformance to non-functional requirements, such as: (1) response time ≤ 5 seconds per request; (2) system availability of at least 99% uptime in business hours; (3) simple interface, using standard Windows forms and menus, making it easy to understand; and (4) system access is restricted with a username and password login.

4. Result and Discussion

4.1 Results

4.1.1 Planning

Through several approaches, such as interviews with resource persons from the casemix section, as well as in-depth observation, the results of the needs analysis show that there is a need for an information system to replace the conventional data recap system. From the needs analysis, the system must be improved to facilitate the data recap process, which in its processing requires several choices of recapitulation forms. Thus, the ideas in the system development design include input of pending claim data, categorization of case types, recapitulation processing, and generating reports according to user needs.

4.1.2 Design

System design aims to provide an overview of the new information system that has been created for users by relying on structured modeling [26]. To facilitate the design of the system, instruments in the form of a Flowmap, Context Diagram, Data Flow Diagram, and Entity Relationship Diagram (ERD) are used. Figure 2 is a flow map, which is a diagram used to visualize the flow of a process as a whole that helps in finding problems of a process in a smaller or specific scope [27]. The workflow mechanism starts with the casemix officer bringing the file for claim submission, followed by the internal verifier of the hospital reviewing the claim's eligibility to reduce the potential for claim rejection and file incompleteness errors. The officer can submit a claim via E-Claim INACBG if the file is eligible. Claim files with issues are entered into the system with a status along with the list of revisions. Casemix staff can then process pending claims data to be recapitulated by grouping them based on similar case types. The Casemix Coordinator can review pending claim reports printed with the recapitulation form.

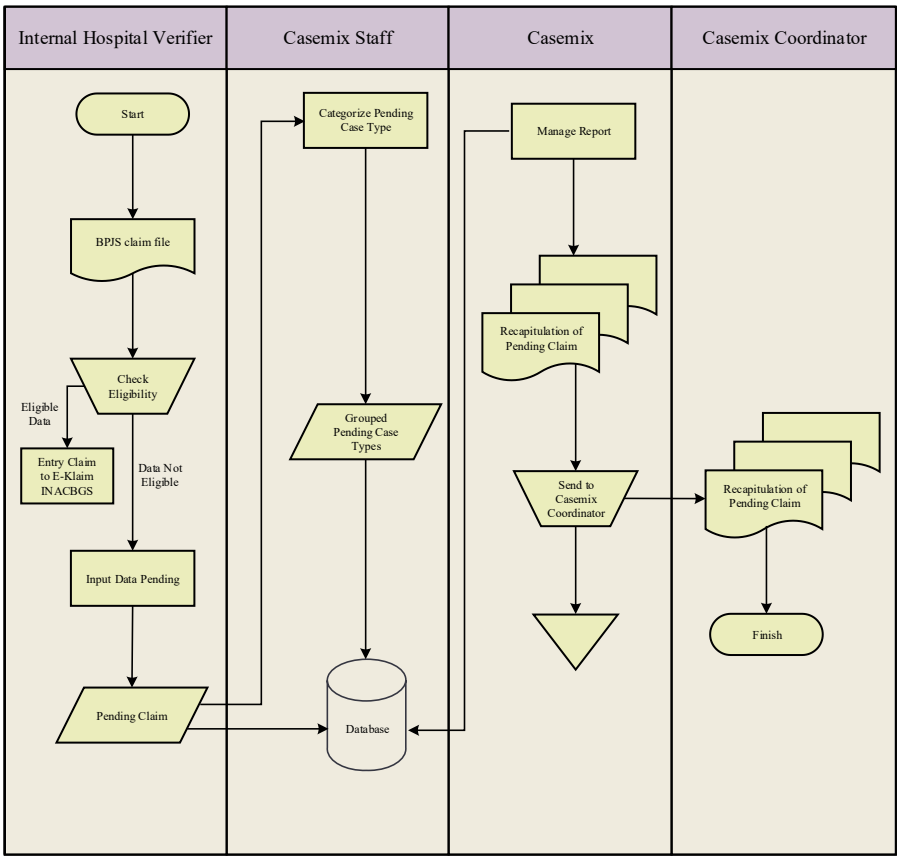

Figure 2. Flow Map of the Recapitulation Process

A context diagram is a diagram that provides a comprehensive overview of the system, starting from the input and output of the system that will run. The context diagram is also the highest-level diagram in the Data Flow Diagram (DFD), which describes the scope of the system being designed [28]. In Figure 3, there are three entities that are interrelated, including hospital internal verifiers, casemix staff, and casemix coordinators. The input flow process starts from inputting officer data, pending claim data, and case type recap. The output flow process on the system produces a recapitulation report with several types of categories, which are then submitted to the casemix coordinator.

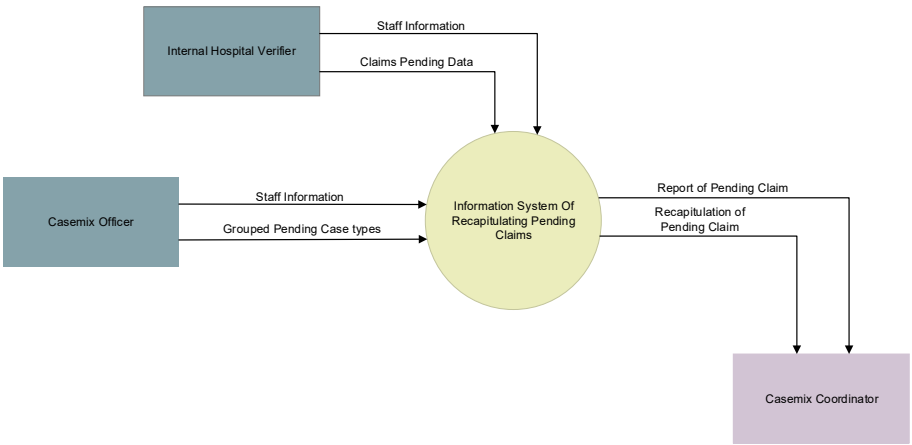

Figure 3. Context Diagram of the Recapitulation System

Data Flow Diagram (DFD) is a diagram that uses notations to describe the input and output processes of software [29]. Table 2 explains how the data flow in the DFD is designed. Figure 5 is an Entity Relationship Diagram or also called ERD, which is a design or format that relates activities that are directly related to the depiction of relationships between data in a database based on underlying data objects that have relationships between them [29]. It is known from the ERD design that there are several connected entities.

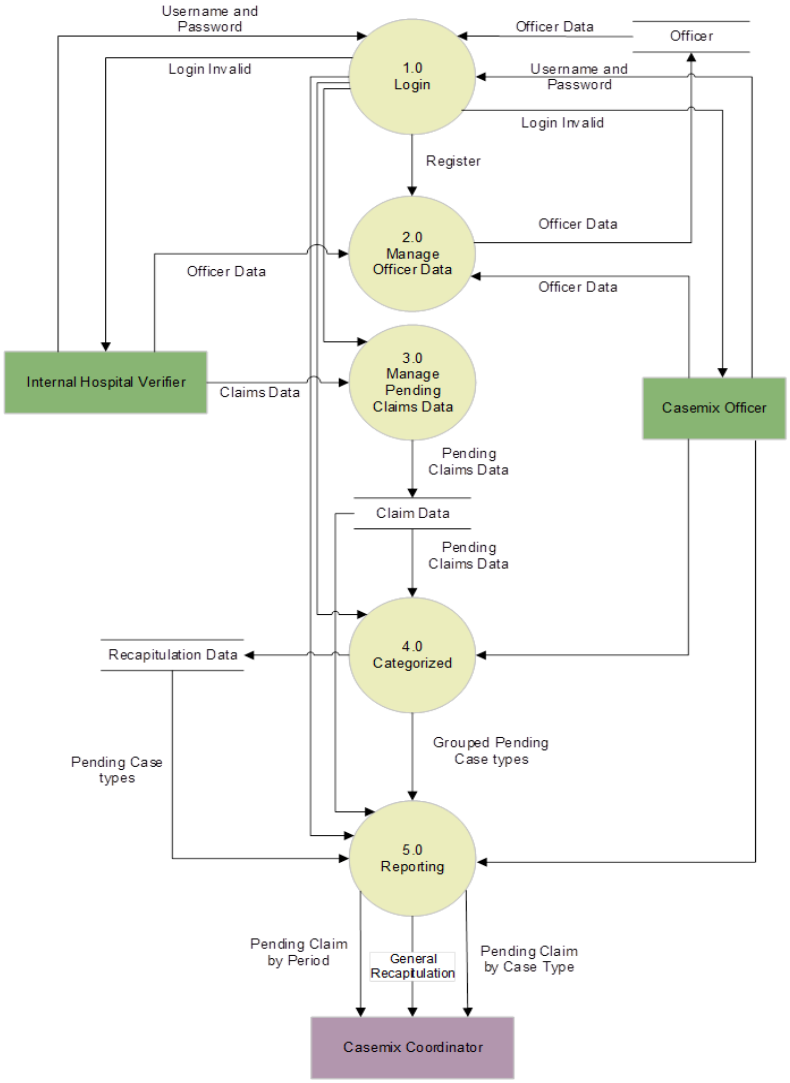

Figure 4. Data Flow Diagram Level 1

Table 2. Input and Output Scheme

No.	Data Flow Diagram	Description
1.0	Login	Enter the system using the username and password of the user (officer)
2.0	Officer Data	Enter officer data to get access to log in to the system
3.0	Claim Data	Entering data on pending claims by the hospital's internal verifier
4.0	Recapitulation Data	Casemix staff process pending claim information by grouping data with similar cases
5.0	Manage Report	Casemix staff prints reports by selecting periodic, recapitulation, or status report categories to report to the casemix coordinator

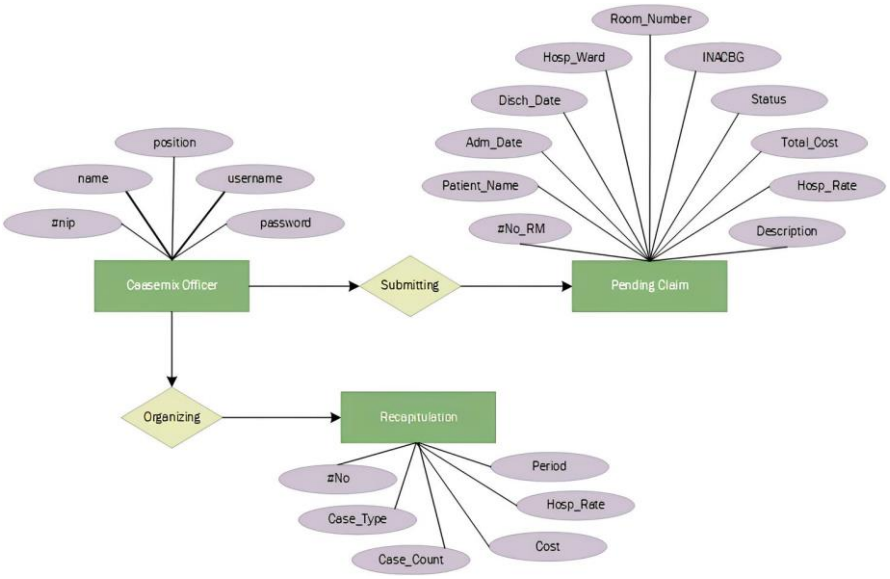

Figure 5. Entity Relationship Diagram (ERD) of the Designed System

4.1.3 Coding

The coding stage is the phase where the system design begins to be translated into real and executable program code. In the Extreme Programming (XP) method, coding is not executed in a single, large-scale operation. Rather, it is executed iteratively and incrementally, with each development stage prioritized according to the relative significance of the modules. These modules are perpetually enhanced and revised in accordance with the changing needs of users and the results of the testing process. The coding process commenced with the creation of the user interface. The login menu is the initial feature used to secure access to the system. This feature is designed to ensure that only users who have authorized access rights can enter and use the system. The login view consists of a username and password that have been registered for identity verification. The system authenticates by matching the input data with the database. The user will be taken to the system's homepage if the information matches. Otherwise, the system will warn the user if it is incorrect. This menu also validates input, such as preventing login if a field is empty. Staff in the casemix section can register if they have not yet been granted access.

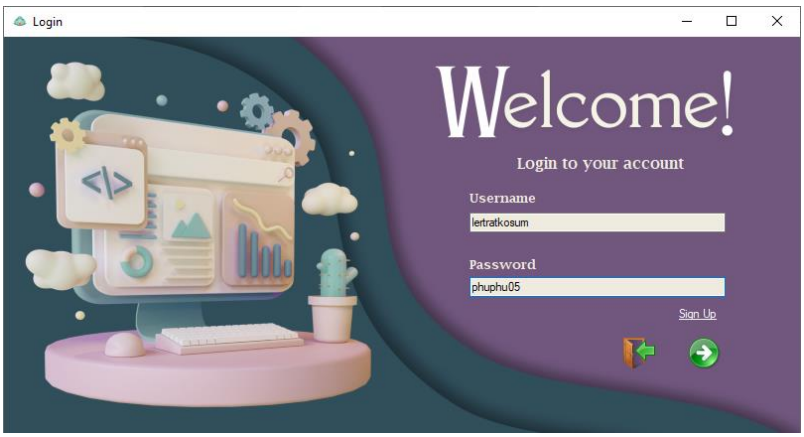

Figure 6. Login Menu

Users who select the enroll feature will be redirected to the Officer Form, where they are required to enter the officer's data, create a username, and generate a password for future login purposes. Upon successful completion of the registration process, the newly registered officer data will be displayed in the designated column. As shown in Figure 8, the primary display is presented to the user once they have successfully logged into the pending claims recapitulation information system. This feature facilitates direct navigation to the system's primary menus. The interface comprises panels for the claim entry, recapitulation, and report menus. Furthermore, the system incorporates a user identity display that presents the name of the user currently logged in and the NIP of the officer.

Figure 7. Form Officer/Register

Figure 8. Dashboard

The Claim Form function is to record crucial information from the pending claim data, including the Surat Eligibilitas Peserta (SEP) number, patient's name, date of service, verification status, INACBG code group, tariffs, and a description that details the issue conditions causing the pending claim. Figure 10 shows the Form Recapitulation, which displays classified claims data and filters by case type and date. The initial table contains system-entered claims. A filter by case type and period is available above the table. As shown in Figure 11, the table's bottom shows the number of records displayed after filtering. Users can directly enter case numbers in required fields. Then, the user can press the add button to automatically fill in the cost and hospital tariff columns with the sum total of records that appear. Recap data is immediately stored in the second table after saving.

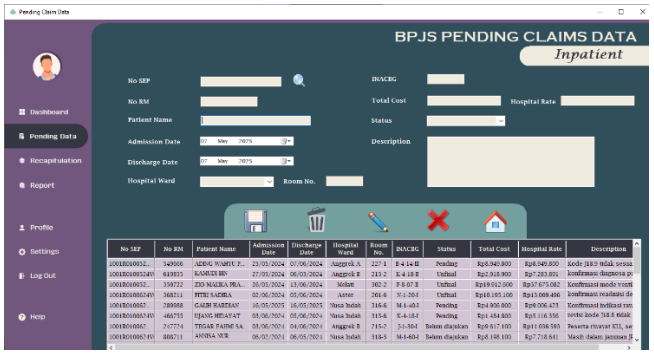

Figure 9. Form Claim

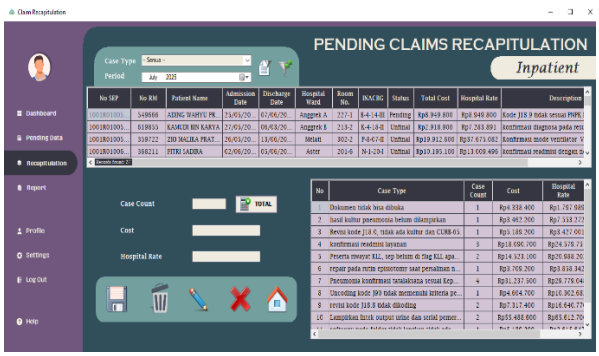

Figure 10. Form Recapitulation

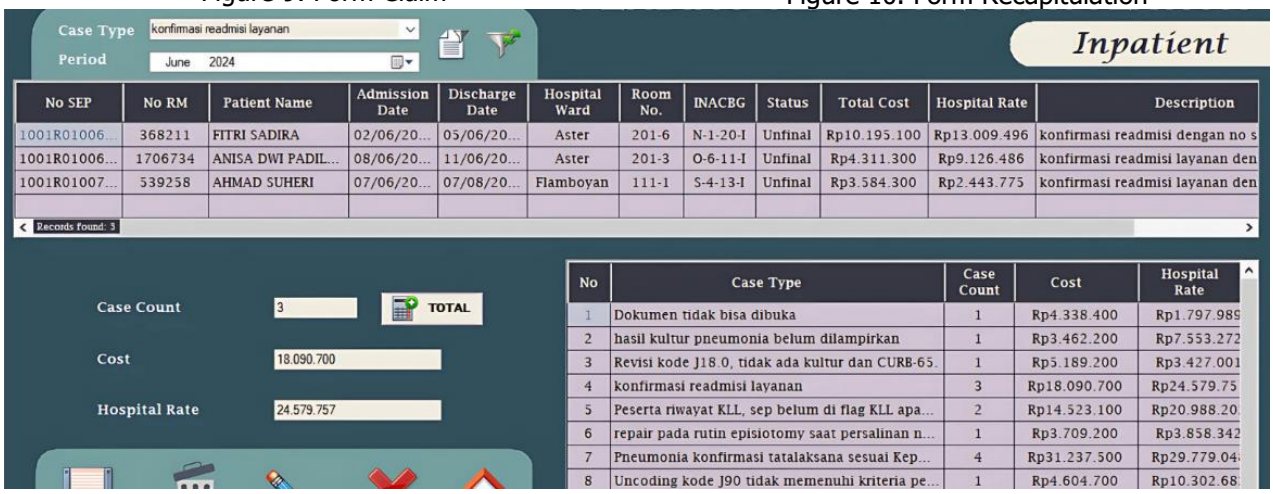

Figure 11. Case Type Filtering Tools

The reporting form is designed to present information in a systematic and structured manner for efficient managerial operations, evaluation, and BPJS claim verification. All data is collected directly from the system database, ensuring real-time reports with the latest entered and processed data. The filtering tool on this form lets users generate reports based on specific periods and display only relevant data. The period-organized report feature allows users to select and display either claim data or recap data. The status-based report function has three options: pending, unfinal, and not yet submitted. The report will look like Figures 15, 16, and 17. Users can print reports directly or as PDFs for documenting or archiving.

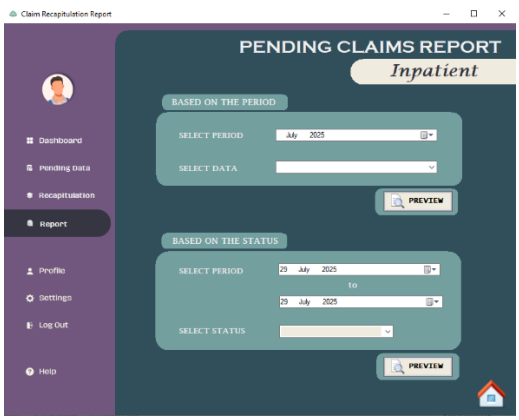

Figure 12. Form Reporting

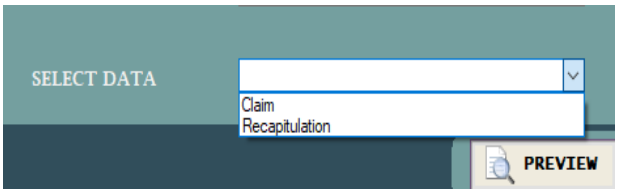

Figure 13. Report Data Selection

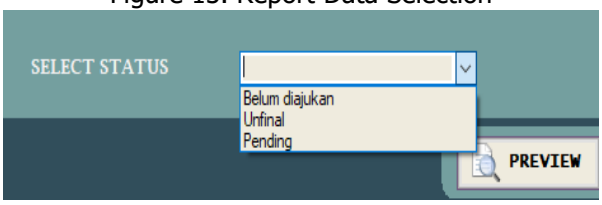

Figure 14. Report Status Selection

Unit testing immediately tests and merges any logic or module changes into the main system. This maintains the system in a deployable state. To verify functionality, each function is tested separately before integration. The case-type filtering function should be able to identify versions of "incomplete resume" like "incomp resume" and "no resume". Refactoring practices are also done during the coding process, which optimizes code structure to make it more efficient and understandable without changing function. This ensures the system remains easy to develop or improve in the next iteration. This aligns with the XP principle that prioritizes adaptability and user-centered development. Overall, the coding stage produces a system that is able to automatically record and group pending claims data quickly and accurately, and the reporting function is designed to be accessed and printed directly.

4.1.4 Testing

After completing several Extreme Programming stages, the final stage is executed. Each component of the pending claim recapitulation system is tested to ensure proper operation. This research uses Black Box Testing, which tests the system's user experience. The tester inputs data into forms and menus to verify output. This enables the discovery of defects, weaknesses, and unwanted behavioral changes without requiring an understanding of the system's complexity.

Table 3. Black Box Testing - BPJS Claim Pending Recapitulation System

No.	Feature Testing	Target Output	Projected Result	Achievement
1	Login with username and password	Input username and password	Successfully load dashboard menu	Accomplished
	Notification if the username and password are incorrect	Pops up a message box with a warning notice	A warning shows when an invalid password and username are entered	
2	Register/Inputting Officer's Data	Inserted officer's information	Officer's data successfully stored	Accomplished
3	Menu	Main page displayed	Dashboard successfully displayed	Accomplished
4	Claim data entry	Complete and valid data	Data successfully uploaded and saved, info notification appears	Accomplished
5	Filter case type data	Display case type options	Successfully display case-type options	Accomplished
6	Calculate fees and rates	Calculate the total cost and tariff of amount data from case-type filtering	Costs and tariffs from the filtered data are successfully totaled	Accomplished
7	Upload recapitulation	Selected case types and total fees and tariffs are uploaded into recap data	Data is successfully processed into a recapitulation	Accomplished
	Duplicate Recapitulation	A message box appears indicating that there is duplicate data	An alert appears, and the data fails to be stored	
8	Claim Pending Report	Present claim reports by period, case type, and status	Successfully display the list of required reports	Accomplished

System testing using the Black Box Testing method shows that all main features function as expected. In addition, a user satisfaction analysis was conducted. The results indicated that users found the system easy to use and that it helped accelerate the recapitulation process. Some feedback was given, such as improvements to the interface and a more flexible form of report features, which makes evaluation more comprehensive as it covers functional, performance, and user satisfaction aspects.

4.1.5 System Efficiency Analysis

The implementation of the inpatient claim recapitulation system resulted in a significant improvement in performance compared to the manual method. In addition, the system generates an interactive dashboard that displays pending claim data based on case type, status, and period. Quantitative comparisons are shown in Table 4.

Table 4. Performance Comparison

Indicator	Manual (Baseline)	System (After)	Change
Average pending claims/month	190 cases	≈114 cases	↓ 40%
Recap compilation time	3–4 days	0.5–1 day	↓ 35%
Classification error rate	12	3	↑ Accuracy 75%

4.2 Discussion

The performance improvement is closely related to the Extreme Programming (XP) practices used in the development. The principle of continuous feedback allowed the system to be developed according to the real needs of casemix staff and verifiers, while short iterations accelerated the feature testing process. The practice of refactoring each iteration contributes to code stability by reducing errors in data processing. Continuous testing also ensures the reliability of features before the system is fully integrated. The research results show that the Visual Basic-based claim pending recapitulation information system with Microsoft Access integration is able to improve the efficiency of claim data management. The system succeeded in reducing the average pending cases from 190 to 114 cases per period, signifying a decrease of 40%. When compared to the research of Lesmana and Taufik (2023) and Pratiwi *et al*. (2023), which focuses on monitoring claim systems and outpatient claim systems, this system shows further development [13][17]. The main advantage is the depth of classification of the causes of pending claims and multi-level granularity reports. This shows that this research not only adopts previous concepts but also makes new contributions in the context of inpatient claims. This finding also aligns with research by Chaturvedi *et al*. (2024) indicating that the use of information technology can help predict and reduce potential claim denials [18]. Thus, the results of this study strengthen the empirical evidence that the integration of technology into the claims management system plays a strategic role in improving the efficiency of claims verification. In practical terms, the system contributes to making it easier for officers to systematically identify the causes of pending claims, provide faster and more accurate recapitulation, and reduce the administrative burden of casemix staff. The real-time recapitulation data also supports managerial decision-making. Nonetheless, there are challenges to implementation, such as the inconsistency of descriptive input from staff, which frequently lacks uniformity. The system's standalone nature limits simultaneous access between staff, and the integration with SIMRS is not yet operational, requiring manual data entry.

The outcome of the system development implemented in this study is in the form of a recapitulation, which presents aggregated claims data for the required reports. As an example, Figure 15 shows the results of the ineligible claims report, which is processed and recapitulated periodically, namely every month. The figure below presents general recapitulation in July 2024, which comprises claims categorized with the status pending, unfinal, and not yet submitted. Figure 16 below shows the results of the pending claims data recapitulated with classification based on the type of case and quantity calculation. A review of the report indicates that among the pending cases identified, there were 10 of the following case types that caused pending claims in June 2024. Figure 17 below presents a periodically recapitulated claim report, categorized by status. The report presents the results filtered to only display data with "pending" status. This data highlights the number of claims that are still under review and need a resolution. We can also identify pending claims data that are close to expiring and have not been resubmitted.

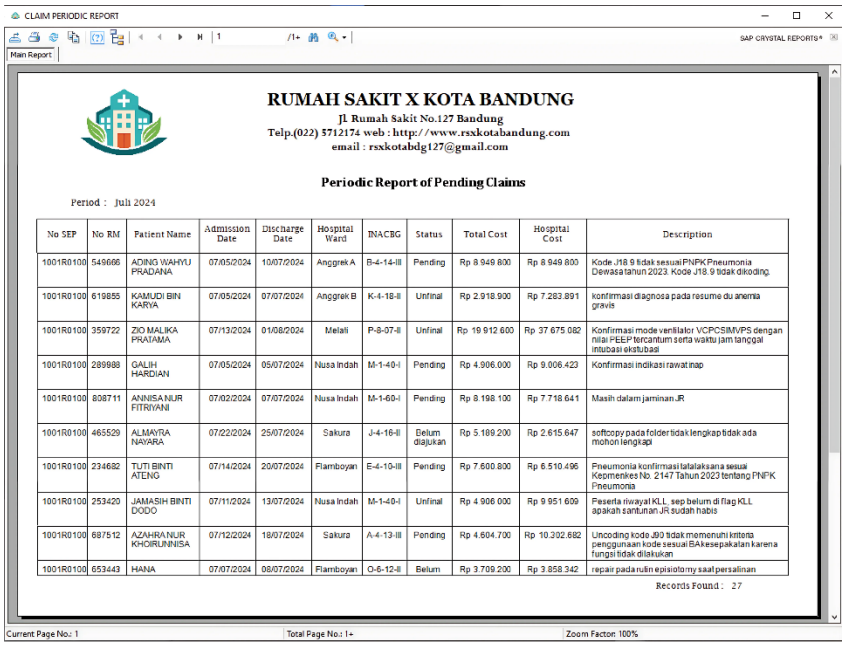

Figure 15. Periodic Report of Pending Claims

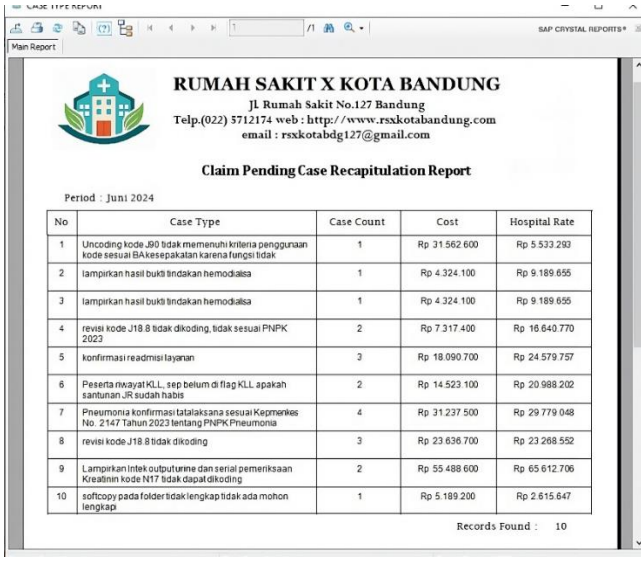

Figure 16. Report Results of Recapitulation of Pending Claim Cases

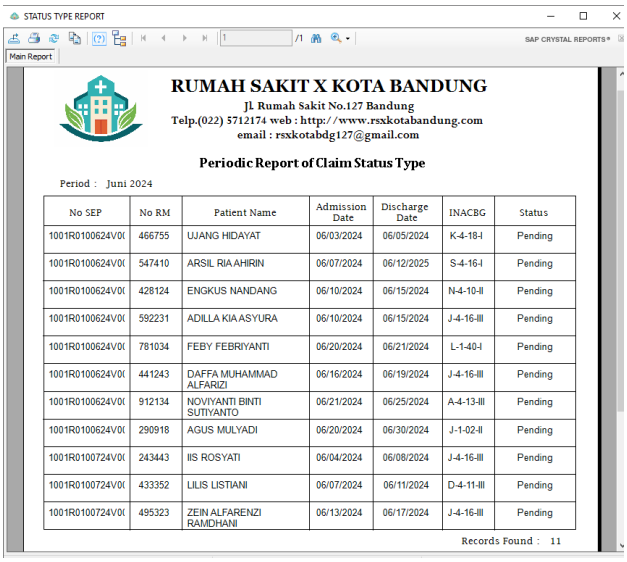

Figure 17. Periodic Report on Claim Status

Despite the developed pending claims recapitulation information system showing effective performance, there are certain limitations that need to be noted. One of the main limitations lies in the classification process, which remains reliant on basic keyword matching (string matching). This approach cannot accommodate excessive variances in writing, uneven captions and abbreviations, or spelling problems, which may diminish classification accuracy in some cases. Furthermore, the system remains standalone (desktop-based) and does not support multi-user or cross-network use. This limits user flexibility, especially if multiple claims staff use the system concurrently. The system's lack of integration with the electronic medical record database and SIMRS requires user involvement in submitting claim data individually. In terms of security, the system has not implemented a sensitive data encryption feature, so the aspect of protecting patient data and claims data needs to be a concern in the next system development, especially to support information security standards in the health sector. These challenges can be overcome with Natural Language Processing, SIMRS or E-Klaim INACBG integration, and data encryption. The system architecture can also be web-based or cloud-based to make integration with hospital information systems easier. These strategies could make the inpatient claims recapitulation information system more adaptive, integrated, and sustainable in hospital claims management.

5. Conclusion and Recommendations

Based on the results of the research described above, it can be concluded that the desktop-based inpatient claim recapitulation information system developed with the Extreme Programming approach is able to improve the efficiency of the BPJS claim verification process. This system is equipped with a claim data entry module, classification of pending claims based on the type of cause, and automatic reports that present recapitulation data in a structured manner. The results of the trial showed a decrease in recapitulation time (35%) compared to the manual process, as well as a decrease in the average pending claims from 190 to 114 cases per period (40%). Additionally, the generated reports assist in analyzing the types of cases that frequently cause pending claims, serving as a basis for evaluation. Thus, this research provides a new contribution in the form of a more detailed classification of inpatient claims and is proven to improve the accuracy and speed of identification of pending claims.

For further development, the system can be improved by adding integration features with E-Klaim INACBG or SIMRS so that the verification process can be carried out more thoroughly and in a more parallel manner. In addition, web-based or cloud-based system development is also recommended so that it can be accessed by multiple users across devices and locations. The use of more advanced text matching methods can also be applied to improve the accuracy of classifying pending claims with more complex variations in description writing, implementation of role-based access control to strengthen security, and adoption of health data interoperability standards such as HL7/FHIR. With such strategies, the system has the potential to evolve into a more adaptive, secure, and sustainable solution in hospital claims management practices.

References

[1] Permenkes RI No 3. (2020). *Tentang Klasifikasi dan Perizinan Rumah Sakit. Peraturan Menteri Kesehatan Nomor 3 Tahun 2020*.

[2] Rosita, N. P. I., Wijaya, W. S., & Mutiarahati, N. L. A. C. (2023). Strategi manajemen rumah sakit dalam meningkatkan kualitas pelayanan pasien. *Jurnal Pengabdian Dan Pemberdayaan Masyarakat, 1*(2), 28–35. https://doi.org/10.54107/abdimahosada.v1i2.191

[3] Permenkes RI No 24. (2022). *Tentang Rekam Medis. Peraturan Menteri Kesehatan Nomor 24 Tahun 2022*.

[4] Hasbullah, A. O., Maidin, H. A., & Machmud, N. U. (2024). Implementasi pending klaim BPJS kesehatan berdasarkan Permenkes No 26 Tahun 2021 di RS Ibnu Sina. *Journal of Aafiyah Health Research (JAHR), 5*(2), 199–209. https://doi.org/10.52103/jahr.v5i2.1699

[5] Utami, Y. T., Soultoni Akbar, P., Amelia, R., & Sari, S. Y. (2024). Faktor penyebab pending klaim BPJS rawat inap dengan penerapan rekam medis elektronik di RSUD Dr. Moewardi Surakarta. In *Prosiding Seminar Informasi Kesehatan Nasional (SIKesNas)*. https://doi.org/10.47701/sikenas.vi.3939

[6] Syahira, N., & Rusdi, A. J. (2024). Strategi optimalisasi klaim jaminan kesehatan nasional di unit casemix rumah sakit "X" Blitar. *Jurnal ARSI: Administrasi Rumah Sakit Indonesia, 10*(2). https://doi.org/10.7454/arsi.v10i2.1182

[7] Taslim, R. K. A., & Meliala, A. (2022). *Mekanisme pengajuan klaim BPJS di Rumah Sakit Umum Siloam Kupang* [Master's thesis, Universitas Gadjah Mada]. https://etd.repository.ugm.ac.id/penelitian/detail/230004

[8] Pangidoan Simbolon, R., Wahyudi, B., & Mulyani, K. (2024). Strategi manajerial dalam menangani klaim tertunda BPJS kesehatan di rawat inap Rumah Sakit Pertamina Sorong. *EKOMA: Jurnal Ekonomi, 4*(1). https://doi.org/10.56799/ekoma.v4i1.6263

[9] Permenkes No 28. (2014). *Tentang Pedoman Pelaksanaan Program Jaminan Kesehatan Nasional. Peraturan Menteri Kesehatan Republik Indonesia Nomor 28 Tahun 2014*.

[10] Maulida, E. S., & Djunawan, A. (2022). Analisis penyebab pending claim berkas BPJS kesehatan pelayanan rawat inap Rumah Sakit Universitas Airlangga. *Media Kesehatan Masyarakat Indonesia, 21*(6), 374–379. https://doi.org/10.14710/mkmi.21.6.374-379

[11] Winarti, W. (2022). Description of the claim submitting system of BPJS inpatient at Hospital X in Bandung Regency. *Jurnal Teras Kesehatan, 4*(2), 63–72. https://doi.org/10.38215/jtkes.v4i2.78

[12] Salina, & Wahab, S. (2024). Analisis data dan faktor penyebab pending klaim BPJS rawat inap guna menunjang ke efektivitasan pengklaiman di RSUD Kota Bandung tahun 2024. *Journal of Medical Record Student (JMeRS), 2*(1), 56–66. https://journal.piksi.ac.id/index.php/jmers/article/view/1431

[13] Pratiwi, S., Gunawan, E., Abdussalaam, F., & Sukmawijaya, J. (2023). Perancangan sistem informasi rekapitulasi klaim pasien BPJS rawat jalan. *MBI, 18*(1). https://doi.org/10.33758/mbi.v18i1.567

[14] Weiskopf, N. G., & Weng, C. (2013). Methods and dimensions of electronic health record data quality assessment: Enabling reuse for clinical research. *Journal of the American Medical Informatics Association, 20*(1), 144–151. https://doi.org/10.1136/amiajnl-2011-000681

[15] Ulil Amri, & Nurwahyuni. (2024). [Title not provided in original references].

[16] Wijayanti, R. A., Dewi, R. D. C., Farlinda, S., & Nuraini, N. (2023). BPJS pending claim settlement strategy in hospital with management approach (Planning, Organizing, Actuating, Controlling). *International Journal of Healthcare and Information Technology, 1*(1), 28–33. https://doi.org/10.25047/ijhitech.v1i1.3917

[17] Lesmana, A., & Taufik, A. (2023). Development of a BPJS hospital claims monitoring systems via bridging dashboard. *Journal of Information System, Informatics and Computing, 7*(2), 295. https://doi.org/10.52362/jisicom.v7i2.1274

[18] Chaturvedi, R., Sharma, S., Kirupa Gopalakrishna Pandian, P., & Sharma, S. (2024). Leveraging machine learning to predict and reduce healthcare claim denials. https://doi.org/10.5281/zenodo.13268360

[19] Handayani, H., Wahyudi, B., & Mulyani, K. (2024). Analisis penyebab pending klaim pasien BPJS rawat inap dalam rangka mengoptimalkan pendapatan di RSUD Banten. *Action Research Literate, 8*(12), 3512–3529. https://doi.org/10.46799/arl.v8i12.2532

[20] Raihan, F. M. (2021). Perancangan sistem informasi rekam medis pada Klinik Saffira Sentra Medika Batam. *Jurnal Sains, Nalar, Dan Aplikasi Teknologi Informasi, 1*(1). https://doi.org/10.20885/snati.v1i1.7

[21] Najrifatussya'diah, S., Syahidin, Y., & Yunengsih, Y. (2024). Rancang bangun menu konfirmasi sebagai pendukung keputusan klaim jaminan kesehatan nasional menggunakan metode extreme programming. *Jurnal Teknologi Sistem Informasi Dan Aplikasi, 7*(2), 643–652. https://doi.org/10.32493/jtsi.v7i2.39123

[22] Beck, K. (2004). *Extreme programming explained: Embrace change* (2nd ed.). Addison-Wesley Professional.

[23] Wibagso, S. S., & Celesta, I. (2022). Extreme programming approach in E-PANJO design to support information management at nursing home. *Jurnal Buana Informatika, 13*(02), 93–104. https://doi.org/10.24002/jbi.v13i02.5707

[24] Supriyatna, A., & Puspitasari, D. (2021). Implementation of extreme programming method in web based digital report value information system design. *International Journal of Information System & Technology Akreditasi, 5*(1), 67–75. https://doi.org/10.30645/ijistech.v5i1.116

[25] Abdussalaam, F., & Ramdani, A. (2022). Perancangan sistem informasi manajemen praktek kerja lapangan berbasis web menggunakan metode agile. *INFOKOM, 10*. http://journal.piksi.ac.id/index.php/INFOKOM

[26] Kustiani, R., Syahidin, Y., & Yunengsih, Y. (2024). Electronic medical record information system design to support the reporting of completeness of BPJS patient medical records with extreme programming method. *Jurnal Teknik Informatika (Jutif), 5*(4), 11–19. https://doi.org/10.52436/1.jutif.2024.5.4.1976

[27] Zilvia, L., Syahidin, Y., & Sari, I. (2024). Perancangan sistem informasi tindakan operasi guna mengelola laporan 10 besar tindakan berbasis elektronik. *Jurnal Teknologi Sistem Informasi Dan Aplikasi, 7*(2), 635–642. https://doi.org/10.32493/jtsi.v7i2.39368

[28] Sagita, D., Yunengsih, Y., & Abdussalaam, F. (2024). Desain sistem informasi klaim BPJS pasien penyakit dalam dengan metode V model. *Jurnal Teknologi Sistem Informasi Dan Aplikasi, 7*(2), 426–434. https://doi.org/10.32493/jtsi.v7i2.38610

[29] Sahir, L., & Wijayanti, R. A. (2022). Faktor penyebab pending claim ranap JK dengan fishbone diagram di RSUP Dr Kariadi. *Jurnal Manajemen Informasi Kesehatan Indonesia, 10*(2), 190. https://doi.org/10.33560/jmiki.v10i2.480