

Bug Tracking Monitoring Development and Management Application

Ade Sutedi^{1*}, Ayu Latifah², Hasbi Pauji³

^{1,2}Institut Teknologi Garut, Indonesia

*email: adesutedi@itg.ac.id

Article Info

Submitted: 13 May 2026

Received: 25 May 2026

Published: 31 May 2026

Keywords:

Bug Tracker;

Monitoring;

Rational Unified Process;

Unified Modeling Language.

ABSTRACT

Software development projects frequently encounter issues during the development process. System failures and malfunctioning features are often caused by minor errors, commonly referred to as *bugs*. Effective bug management is a crucial aspect of software development to ensure that issues are resolved promptly, thereby supporting the continuity of users' business processes. The objective of this study is to develop a bug tracking and management application by incorporating deadline management, reminder notifications, and bug resolution history features. Specifically, the study aims to enhance the productivity of development teams by providing an application capable of managing, tracking, and resolving software bugs efficiently. In the long term, this research seeks to develop an effective and efficient bug management system that contributes to the production of higher-quality software. The methodology employed in this study is the Rational Unified Process (RUP), which consists of four phases: Inception, Elaboration, Construction, and Transition. Unified Modeling Language (UML) is utilized as the modeling tool for system visualization, including four types of diagrams: Use Case Diagram, Activity Diagram, Sequence Diagram, and Class Diagram. The result of this research is a bug tracking and management application that enables users to monitor and manage software bugs effectively. The application provides a deadline feature for tracking bug resolution schedules, a reminder feature for notifying users of pending tasks, and a bug resolution history feature that records detailed information regarding the handling and resolution of reported bugs.

1. INTRODUCTION

Application development is a series of processes carried out by a development team or an individual developer, starting with initial planning and design, prototype creation, implementation using programming languages, maintenance and testing, and continuing until the application is completed and ready for use. Throughout the application development process, development teams may encounter various issues, such as system failures, malfunctioning features, and other technical problems. One of the most common issues faced by developers is software bugs [1]. A bug is a defect or technical issue that causes software to function improperly or fail to meet its intended requirements. Developers play a crucial role in detecting and resolving bugs within a project to ensure that project objectives and targets are achieved [2]. Monitoring within the bug-tracking process is an essential aspect of software development, as it helps ensure timely issue resolution and supports the continuity of users' business processes [3]. A bug tracking and management application can assist development teams and individual developers in tracking software defects within a system, enabling project members to report, manage, and analyze bug reports efficiently [4].

Previous studies have discussed various aspects of bug tracking and management systems. The first study aimed to design and develop a bug-tracking application using the Rational Unified Process (RUP) methodology to help software development teams evaluate, correct, track, and control bugs in their projects [1]. The second study implemented a Bug Tracking System for software development teams and reported that 78.72% of survey respondents perceived improvements in bug-resolution efficiency, resulting in a lower recurrence of similar errors [5]. The third study developed a Bug Tracking and Reporting System (BTS) capable of uploading, searching, and storing bug details. The system also ensured that all developer errors were documented as references for future projects through records of handled and resolved bugs [6]. The fourth study focused on developing a decision-support system to determine bug priorities using the Simple Additive Weighting (SAW) method in the software testing process [4]. The fifth study designed and developed a web-based monitoring system for tracking student disciplinary violations. The monitoring system was intended to assist counseling staff in controlling, supervising, and evaluating student activities and violations [7].

Based on the findings of previous studies, this research aims to develop a web-based bug-tracking and management application. The proposed application is designed to facilitate tracking software defects by incorporating additional features such as deadline management, reminder notifications, and a bug-resolution history. The bug resolution history serves as a facility for recording and storing data related to completed tasks, thereby providing comprehensive documentation of bug-handling activities and supporting future maintenance and development efforts.

2. RESEARCH METHODOLOGY

The Rational Unified Process (RUP) was employed as the research methodology in this study. The use of RUP aims to facilitate the development of the bug-tracking and management application by providing a structured, detailed framework throughout the development process [8]. Furthermore, RUP enables developers to effectively plan project schedules and estimate software development costs while maintaining high software quality and minimizing overall development expenses [9].

2.1. Research Framework

The research framework is a diagram that illustrates the study's overall structure and provides an overview of the research process's logical flow. The research framework used in this study is presented below.

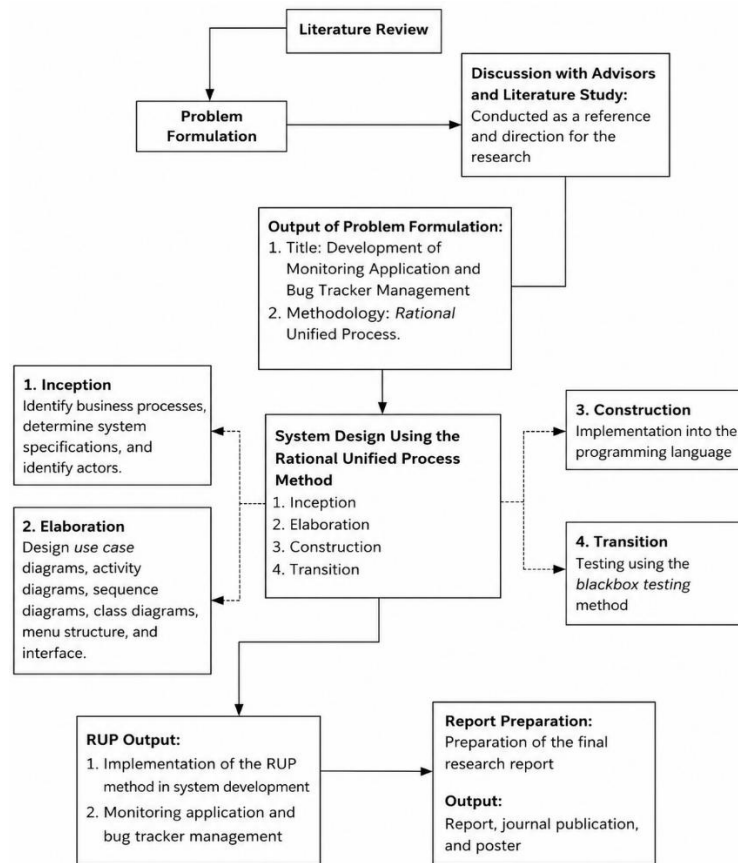


Figure 1. Conceptual Framework

2.2 Work Breakdown Structure

A Work Breakdown Structure (WBS) is a method used to simplify project management by dividing a project into smaller and more manageable components. The WBS used in this study is presented below.

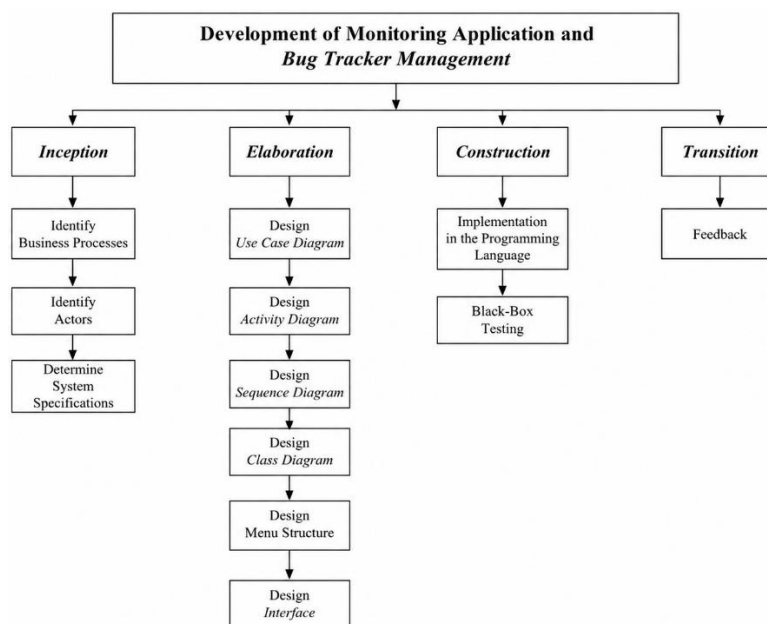


Figure 2. Work Breakdown Structure

The explanation of the WBS diagram shown in Figure 2 is as follows:

1. Inception

The first phase is Inception, which consists of three main activities: actor identification, system specification definition, and business process identification. These activities are carried out based on the results of the literature review.

2. Elaboration

The second phase is Elaboration. The activities performed during this phase include designing the Use Case Diagram, Activity Diagram, Sequence Diagram, Class Diagram, menu structure, and user interface. These artifacts serve as part of the Unified Modeling Language (UML) modeling process.

3. Construction

This phase focuses on implementing the designed system using a programming language. In addition, Black Box Testing is employed to verify that the system functions correctly according to its requirements [8].

4. Transition

The Transition phase focuses on obtaining user feedback, which serves as direct input. This feedback includes responses on the user interface, ease of use, and additional features users require. Feedback can be collected through various methods, such as surveys, interviews, and usability testing [10].

3. RESULT AND DISCUSSION

3.1. Results

The result of this study is a web-based Bug Tracking Monitoring, and Management Application equipped with several key features, including a deadline feature to display task completion schedules, a reminder feature to notify users involved in the project, and a history feature that allows users to view detailed information regarding projects handled by developers from initiation to completion.

3.2. Discussion

3.2.1. Inception

The activities conducted during this phase include business process identification, actor identification, and system specification definition. These outputs were obtained through a literature review of journals and references related to this study. The business process of the proposed system is illustrated in Figure 3.

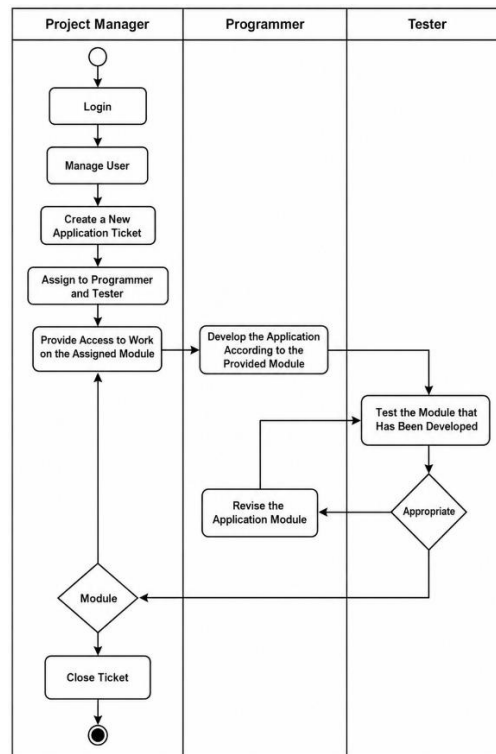


Figure 3. Business Process

The process begins with the Project Manager logging into the system. Subsequently, the Project Manager manages the users involved in the project. Afterward, the Project Manager can create a new application ticket and assign programmers and testers to the ticket. Assigned programmers and testers can then access the ticket and work on bug resolution tasks within a specified deadline until completion.

3.2.2. Elaboration

The Elaboration phase focuses on designing the system architecture using Unified Modeling Language (UML). The UML models developed in this study include the Use Case Diagram, Activity Diagram, Sequence Diagram, and Class Diagram. The application design based on UML is described as follows [1].

1. Use Case Diagram Design

The Use Case Diagram is used to identify the functions available within the Bug Tracking, Monitoring, and Management Application and the actors who interact with those functions [1]. The Use Case Diagram of the application is presented in Figure 4.

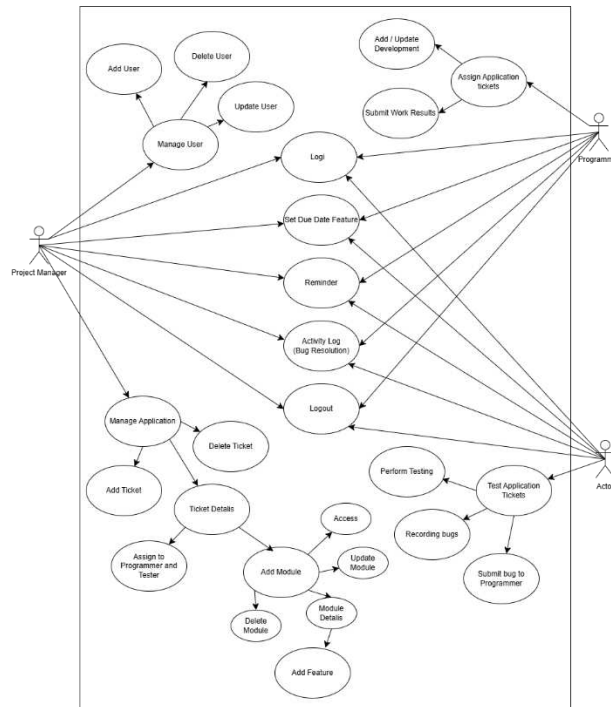


Figure 4. Use Case Diagram of the Bug Tracking Monitoring, and Management Application

2. Activity Diagram Design

This stage develops Activity Diagrams based on the Use Case Diagram to model the workflows of the Bug Tracking, Monitoring, and Management Application. One of the designed Activity Diagrams is shown below.

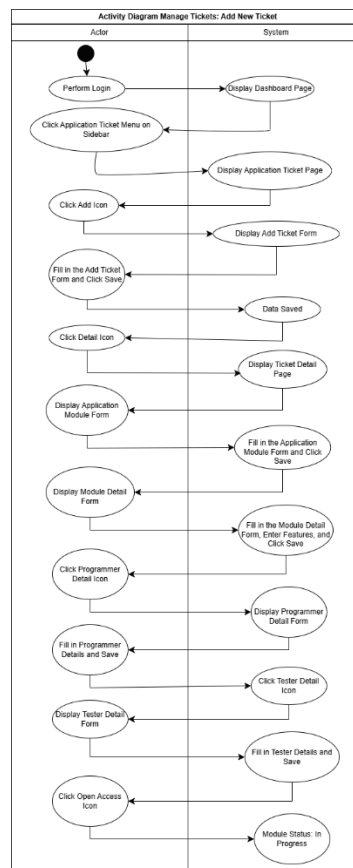


Figure 5. Activity Diagram for Ticket Management: Add Ticket

As illustrated in Figure 5, the system performs numerous activities to accommodate user requirements. These activities are designed according to the functional needs identified during system analysis.

3. Sequence Diagram Design

The Sequence Diagram illustrates the interactions between actors and the system. The Sequence Diagram for this application is presented below.

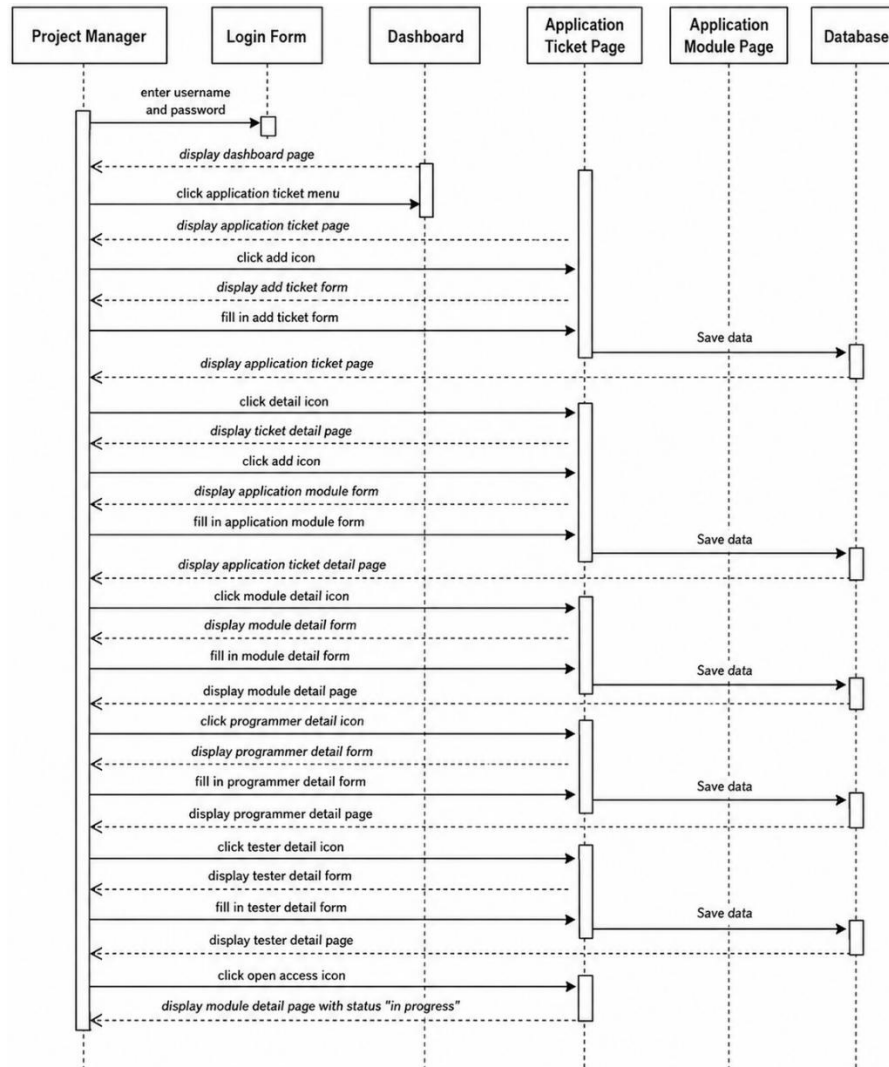


Figure 6. Sequence Diagram for Ticket Management: Add Ticket

4. Class Diagram Design

The Class Diagram describes the system structure, database entities, and relationships among classes within the application. The resulting Class Diagram for the web-based Bug Tracking, Monitoring, and Management Application is shown in Figure 7.

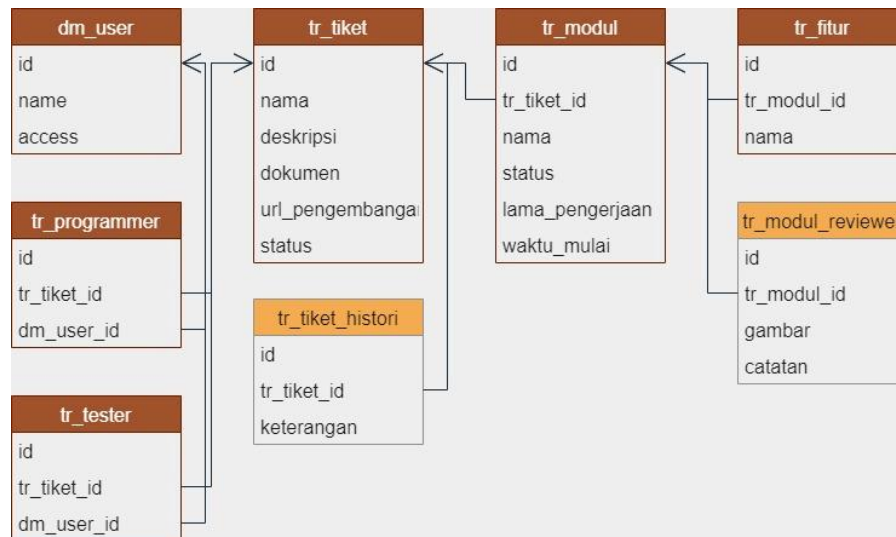


Figure 7. Class Diagram

5. User Interface Design

One of the designed user interfaces for the Bug Tracking Monitoring and Management Application is presented below.

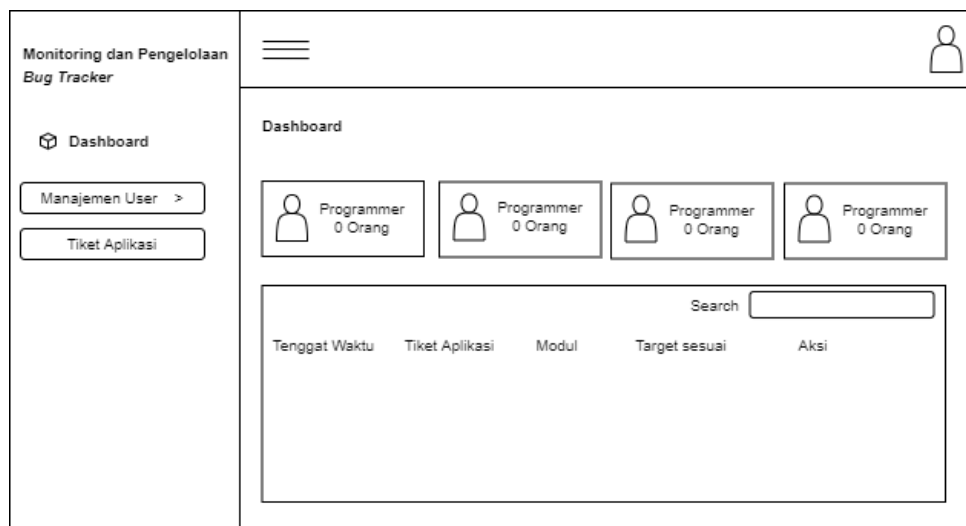


Figure 8. Project Manager Dashboard

The Project Manager Dashboard displays menus for user management and application ticket management in the sidebar. In addition, the dashboard provides information on the number of programmers and testers, completed and pending tickets, and their assigned deadlines.

3.2.3. Construction

The Construction phase focuses on implementing the application and conducting system testing using the Black Box Testing method.

1. System Implementation

The system was developed using the Laravel framework in PHP, with Apache serving as the web server. MySQL was used as the database management system, while Visual Studio Code was utilized as the source code editor. The implementation result is shown below.

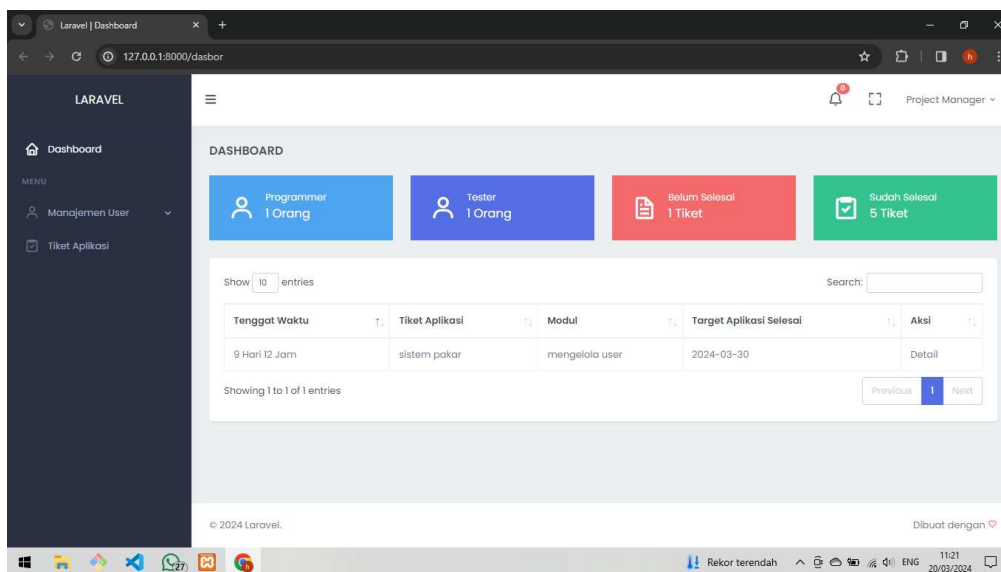


Figure 9. Project Manager Dashboard

2. Testing

This phase focuses on testing activities to ensure that the application functions properly and can be effectively used by users. System testing was conducted using the Black Box Testing method [1]. Black Box Testing evaluates software functionality by examining its inputs and outputs without considering its internal structure [11].

Table 1. Functional Testing for Project Manager

No.	Activity Name	Expected Result	Validity	
			Y	T
1	Login	Verify login credentials and redirect to the dashboard	Y	
2	Logout	Log out from the account	Y	
3	User Management: Programmer	Add, edit, and delete programmer data	Y	
4	User Management: Tester	Add, edit, and delete tester data	Y	
5	Application Ticket Menu	Add, edit, and delete application tickets	Y	
6	Ticket Detail Form	Add, edit, and delete ticket details	Y	
7	Module Detail Form	Add, edit, and delete module details	Y	
8	Programmer Detail Form	Add programmer information	Y	
9	Tester Detail Form	Add tester information	Y	
10	History Menu	View activity history	Y	

3. Transition

The Transition phase focuses on gathering user feedback. Respondents were asked to complete a questionnaire distributed through Google Forms. The evaluation employed a five-point Likert scale, where:

- 1 = Strongly Disagree
- 2 = Disagree
- 3 = Neutral
- 4 = Agree
- 5 = Strongly Agree

This measurement technique is commonly used to evaluate perceptions, attitudes, and opinions regarding social phenomena or events [10].

Table 2. Questionnaire Results

No.	Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	Does the Bug Tracker Monitoring and Management Application have an interface that is easy to understand?			0	11	9
2	Does this application provide features that meet user needs?			0	14	6
3	Is this application easy to use and operate?			0	13	7
4	Does this application help you manage bugs effectively?			1	14	5
5	How likely are you to recommend this application to others?			1	12	7
Total Responses		0	0	2	64	34
Score Total		0	0	6	256	170
Σ Score						432

Based on the questionnaire results, the total score obtained was 432, while the maximum expected score was 500. The feasibility percentage was calculated as follows:

$$\text{Feasibility Percentage} = \left(\frac{\text{Questionnaire Score}}{\text{Expected Score}} \right) \times 100\% \quad (1)$$

$$= (432 / 500) \times 100\% \\ = 86.4\%$$

The questionnaire results indicate that the application achieved a feasibility score of 86.4%, which falls within the “Highly Feasible” category according to the established feasibility criteria. These findings demonstrate that the Bug Tracking Monitoring and Management Application is well accepted by users and effectively supports bug management activities through its deadline, reminder, and bug history features.

4. CONCLUSION

Based on the findings of this study, the research successfully developed a Bug Tracking, Monitoring, and Management Application with three key enhancements: a deadline feature, a reminder feature, and a bug resolution history feature. The implementation of these features is expected to improve the efficiency of bug-handling processes, enabling development teams to track and resolve bugs more quickly and effectively than conventional manual methods. The contribution of this research lies in its ability to support the management and resolution of software bugs within development projects. Using the Bug Tracking, Monitoring, and Management Application can reduce the time required to manage bug reports, thereby improving development teams' productivity. Furthermore, the application helps ensure that software products contain fewer defects, resulting in more reliable, stable, and higher-quality software systems.

One limitation of this study is that user feedback and evaluation were collected from only a limited number of programmers. Therefore, the findings may not fully represent the perspectives of all potential users involved in software development projects.

For future research, it is recommended to incorporate a bug-ticket prioritization feature that enables bugs to be handled based on their urgency and complexity. Another recommendation is to add a Frequently Asked Questions (FAQ) feature, which would help users resolve issues related to module development and bug-resolution processes more efficiently.

REFERENCES

- [1] S. I. Mandala, L. Fitriani, and A. Sutedi, "Rancang Bangun Aplikasi Pengelolaan Bug Tracker Berbasis Web pada Pengembangan Software," *Jurnal Algoritma*, vol. 19, no. 1, pp. 189–200, 2022.
- [2] E. Tuna, V. Kovalenko, and E. Tüzün, "Bug tracking process smells in practice," in *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice*, 2022, pp. 77–86.
- [3] S. Monalisa and B. Kurniadi, "Sistem Informasi Monitoring Perkembangan Skripsi dengan Reminder System untuk Mahasiswa," *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, vol. 10, no. 1, pp. 23–32, 2019, doi: 10.31849/digitalzone.v10i1.2340.
- [4] S. H. Wibowo and M. Akbar, "Sistem Penunjang Keputusan Prioritas Bug Dalam Tahap Pengujian dengan Simple Additive Weighting," *JURNAL TEKNIK INFORMATIKA (JUTEKIN)*, vol. 11, no. 1, 2023.
- [5] Y. Hartanto and Suwarno, "Implementasi Bug Tracking System dengan Metodologi Scrum dan Algoritma Cosine Similarity," *Jurnal Riset Komputer*, vol. 9, no. 1, pp. 2407–389, 2022, doi: 10.30865/jurikom.v9i1.3861.
- [6] S. Liang and Y. Hartanto, "Implementasi Bug Tracking System dengan Metodologi Scrum dan Algoritma Cosine Similarity," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 1, pp. 24–32, 2022.
- [7] A. H. Pratama, "Rancang Bangun Aplikasi Monitoring dan Evaluasi Pelanggaran Siswa pada SMP Negeri 20 Bandar Lampung Berbasis Web," *Jurnal Teknologi dan Sistem Informasi*, vol. 3, no. 4, pp. 63–68, 2022.
- [8] R. Perwitasari, R. Afawani, and S. E. Anjarwani, "Penerapan metode rational unified process (rup) dalam pengembangan sistem informasi medical check up pada citra medical centre," *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTIIKA)*, vol. 2, no. 1, pp. 76–88, 2020.
- [9] A. D. Supriatna, D. D. S. Fatimah, and I. Nurrohman, "Perancangan Aplikasi Pelayanan Data Penjualan Toko Optik Berbasis Web dengan Menggunakan Metodologi Rational Unified Process," *Jurnal Algoritma*, vol. 19, no. 1, pp. 23–30, 2022.
- [10] L. Luthfiana, J. C. Young, and A. Rusli, "Implementasi Algoritma Support Vector Machine dan Chi Square untuk Analisis Sentimen User Feedback Aplikasi," *Ultimatics: Jurnal Teknik Informatika*, vol. 12, no. 2, pp. 125–126, 2020.
- [11] M. N. Ichsanudin, M. Yusuf, and S. Suraya, "Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula," *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 1, no. 2, pp. 1–8, 2022.
- [12] V. H. Pranatawijaya, W. Widiatry, R. Priskila, and P. B. A. A. Putra, "Penerapan skala Likert dan skala dikotomi pada kuesioner online," *Jurnal Sains Dan Informatika*, vol. 5, no. 2, pp. 128–137, 2019.