Type of Contribution:
Research Paper
Review Paper
Case Study

ENERGY: JURNAL ILMIAH ILMU-ILMU TEKNIK
Vol. 16, No. 1 (2026) pp 106-122
DOI: 10.51747/energy.v16i1.1617

E-ISSN: 2962-2565

This article contributes to:

Design and Development of a Web-Based SBMPTN Try-Out Application: A Case Study at SMK N 2 Kotabumi

Maryana[1*], **Ryan Aji Wijaya**[1]

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

*2259201072@umko.ac.id

Abstract

This research is motivated by the suboptimal state of exam simulation media for students at SMK N 2 Kotabumi in preparing for the Joint Entrance Examination for State Universities (SBMPTN), where conventional tryout processes result in slow and inaccurate data processing. As a solution to these issues, this study designs a web-based tryout application developed using the Waterfall method through the stages of requirement analysis, design, coding, and systematic testing. The application is designed to simulate the Computer Assisted Test (CAT) mechanism in real-time, thereby providing instant and transparent assessment feedback for both students and the school. The results of this research are expected to provide an effective self-simulation platform to enhance academic readiness and the accountability of student learning evaluations at SMK N 2 Kotabumi, ultimately achieving a more competitive graduation quality at the national level.

Keywords: Computer Assisted Test (CAT), SBMPTN, SMK N 2 Kotabumi, Waterfall Method, Web-Based Application

Article Info
Submitted: 2026-05-11
Revised: 2026-06-12
Accepted: 2026-06-18
Published: 2026-06-19

Publisher
Universitas Panca Marga

1. Introduction

Vocational secondary education currently faces significant challenges in preparing graduates for state university entrance examinations, particularly through the Joint Entrance Examination for State Universities (SBMPTN) track. As an educational institution focused on job readiness, SMK N 2 Kotabumi maintains a strategic responsibility to facilitate students aspiring to pursue higher education. However, the exam preparation process or tryouts conducted thus far are often hindered by conventional mechanisms that are inefficient in terms of time and cost.

In fact, the integration of information technology in schools is crucial, where information systems defined as a combination of computer technology and organized procedures for processing data into functional information have been proven to provide faster and more accurate evaluation results compared to manual methods [1], [2].

The primary issue at SMK N 2 Kotabumi is the lack of exam simulation media that align with Computer Assisted Test (CAT) standards. As a digital-based evaluation platform, CAT offers advantages such as automated question randomization and real-time result presentation [3]. Without this technology, paper-based tryouts not only impose an operational burden but also delay assessment feedback. Such delays in learning evaluation information which should ideally serve as an instrument for measuring academic readiness can hinder students' ability to identify their academic weaknesses at an early stage [4], [5]. Consequently, there is an urgent need for a digital platform capable of accurately simulating authentic examination environments.

The development of a web-based tryout application stands as the most relevant solution because websites function as virtual infrastructures capable of distributing information efficiently and interactively for users [6]. This system offers the convenience of practicing both independently and in a structured manner through a dynamic question database and instant assessment [7]. Furthermore, the implementation of this system serves as a collective academic progress monitoring tool for the school, featuring secure archives of grade data within a centralized database storage.

Previous studies have demonstrated the use of PHP/MySQL-based systems for online assessments. For example, [8] developed an online exam simulation platform with immediate scoring; however, the system lacked historical data tracking and analytical tools to support school management. Another study by [9] implemented linear online quizzes with static question lists, but the system did not provide randomized question ordering. To address these limitations, the proposed system for SMK N 2 Kotabumi introduces a centralized administrative monitoring dashboard that enables the tracking of collective student progress and score trends over time, while also incorporating automated pseudo-randomization of questions and instant real-time scoring in accordance with Computer Assisted Test (CAT) standards.

In an effort to build a stable application, this research implements the Waterfall method. As a systematic conventional approach in digital product development [10], this method ensures that every stage from requirements analysis and UML design to coding and testing is conducted sequentially. The use of the Waterfall model is highly appropriate because the system's functional requirements, such as time

management and score processing, can be clearly and statically defined from the outset to minimize the risk of functional failure [11]. Through this structured design, SMK N 2 Kotabumi can possess an independent simulation platform capable of enhancing the quality of student graduation at the national level.

2. Methods

2.1 Data Collection Methods

a. Observation

Observation is defined as an evaluation instrument used to measure individual behavior or monitor an activity directly. This method involves systematic observation and documentation of phenomena emerging within the research object, where the observer records factual data captured through visual and auditory senses [12]. Based on the observation results, it was identified that the current exam simulations still rely on printed media (paper), leading to resource waste and time inefficiency in the distribution of questions.

b. Interview

Interviewing is a data collection technique involving verbal interaction between the interviewer and the interviewee. This communication process can be conducted directly or face-to-face to elicit specific information or obtain relevant data in accordance with the established research objectives [13]. Interviews with guidance counselors and curriculum staff confirmed that the primary challenge encountered is the difficulty of processing scores rapidly to map student capabilities before the actual SBMPTN exam. The informants expressed a critical need for a digital system capable of providing instant and accurate scoring results to enhance the effectiveness of academic evaluations within the school environment.

c. Literature review facilitates a comprehensive analysis of various fundamental theories and concepts that underpin a research study [14].

2.2 System Development Method

Waterfall method (in **Figure 1**) was selected due to its systematic and sequential nature, ensuring that each phase of the tryout application development is thoroughly documented before proceeding to the next stage:

a. Communication

In this initial phase, direct observations and interviews were conducted at SMK N 2 Kotabumi to understand the constraints in conventional tryout implementation. The researcher identified primary user requirements, such as ease of access to questions for students and the necessity of automated assessment for the school, to align the application's vision with field-specific issues.

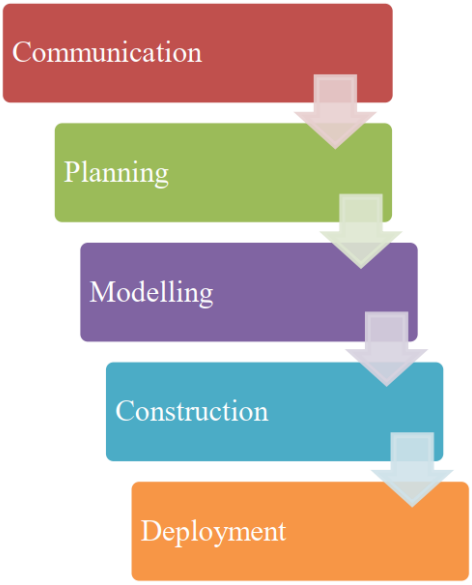

Figure 1. Waterfall method

b. Planning

Based on the data from the communication phase, a work plan was established, encompassing development time estimates and resource requirement analysis. The researcher determined the technological specifications to be utilized, specifically a web-based platform (PHP and MySQL), and set a development schedule ranging from SBMPTN question bank processing to the design of Computer Assisted Test (CAT) features.

c. Modeling

This stage focuses on designing the application architecture and user interface (UI/UX). The researcher created Flowcharts, Use Case Diagrams, and database designs (Entity Relationship Diagrams). The design was developed to closely resemble the actual SBMPTN system interface, providing a realistic simulation experience for students.

d. Construction

This phase involves the realization of the design into program code (coding). The developer built the core functionalities of the application, such as the question-answering system with automated time constraints, real-time score calculation, and question database management. Once coding was completed, Black-box testing was performed to ensure all system modules operated according to the scenarios without logical errors.

e. Deployment

The final stage involves the implementation of the application within the SMK N 2 Kotabumi environment. The application was uploaded to a server to be accessible to students via web browsers. The researcher conducted a system handover to the school, followed by monitoring to ensure that this digital

tryout platform effectively enhances administrative efficiency and student academic readiness.

3.Results and Discussion

3.1 UML System Modeling

3.1.1 Use Case

A Use Case Diagram (in **Figure 2**) serves as a modeling instrument to illustrate the behavior of the information system being developed. Through this diagram, developers can map specific system functionalities and determine the access rights of each actor regarding these features [15].

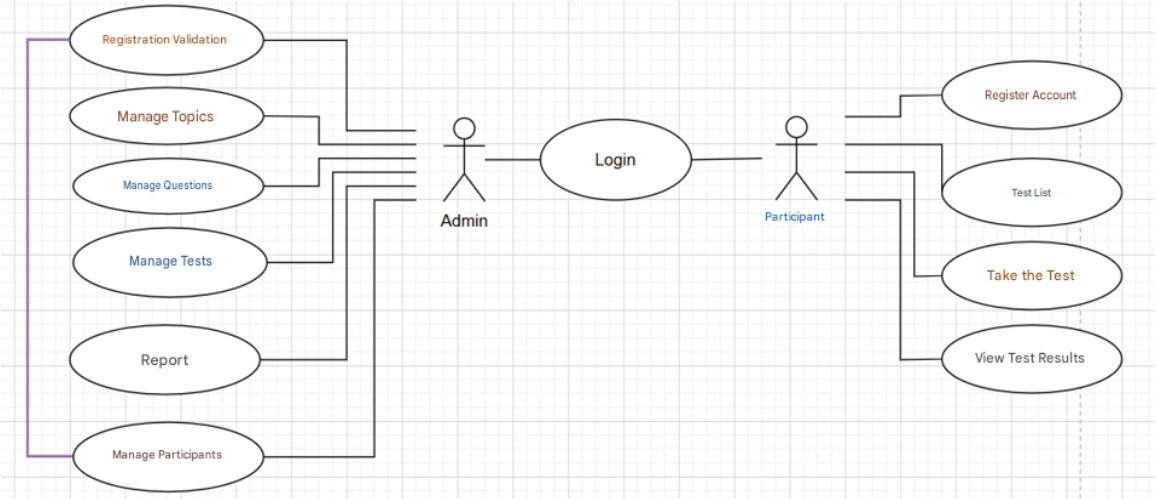

Figure 2. Use Case concept

The Use Case for the 'Design of a Web-Based SBMPTN Try-Out Application' illustrates the interaction between two primary actors, namely the Admin and the Participant, who are connected through a Login process as the system's primary access gateway. The Admin holds full managerial control, which encompasses registration validation, management of topics, questions, and tests, as well as report generation and participant data management. Meanwhile, the Participant focuses on the user-end functional flow, beginning with account registration, selecting tests, engaging in interactive test-taking, and viewing their completed test results.

3.1.2 Activity Diagram

An Activity Diagram (in **Figure 3**) is a visual representation that models the workflow or sequence of activities within a system in a structured manner. Furthermore, this diagram serves as an instrument to define and categorize system interface flows to clarify the interaction between the user and the software [16].

The Activity Diagram for 'Take Test' illustrates the system flow starting with the Login process and account validation by the System through the Database. Upon successful validation and entry into the Dashboard page, the Participant selects the 'Take Test' menu, which triggers the system to receive the request and retrieve

relevant test data from the database. The process continues with the system fetching question items from the database to be displayed on the Participant's screen, until the entire sequence of activities eventually reaches the final state.

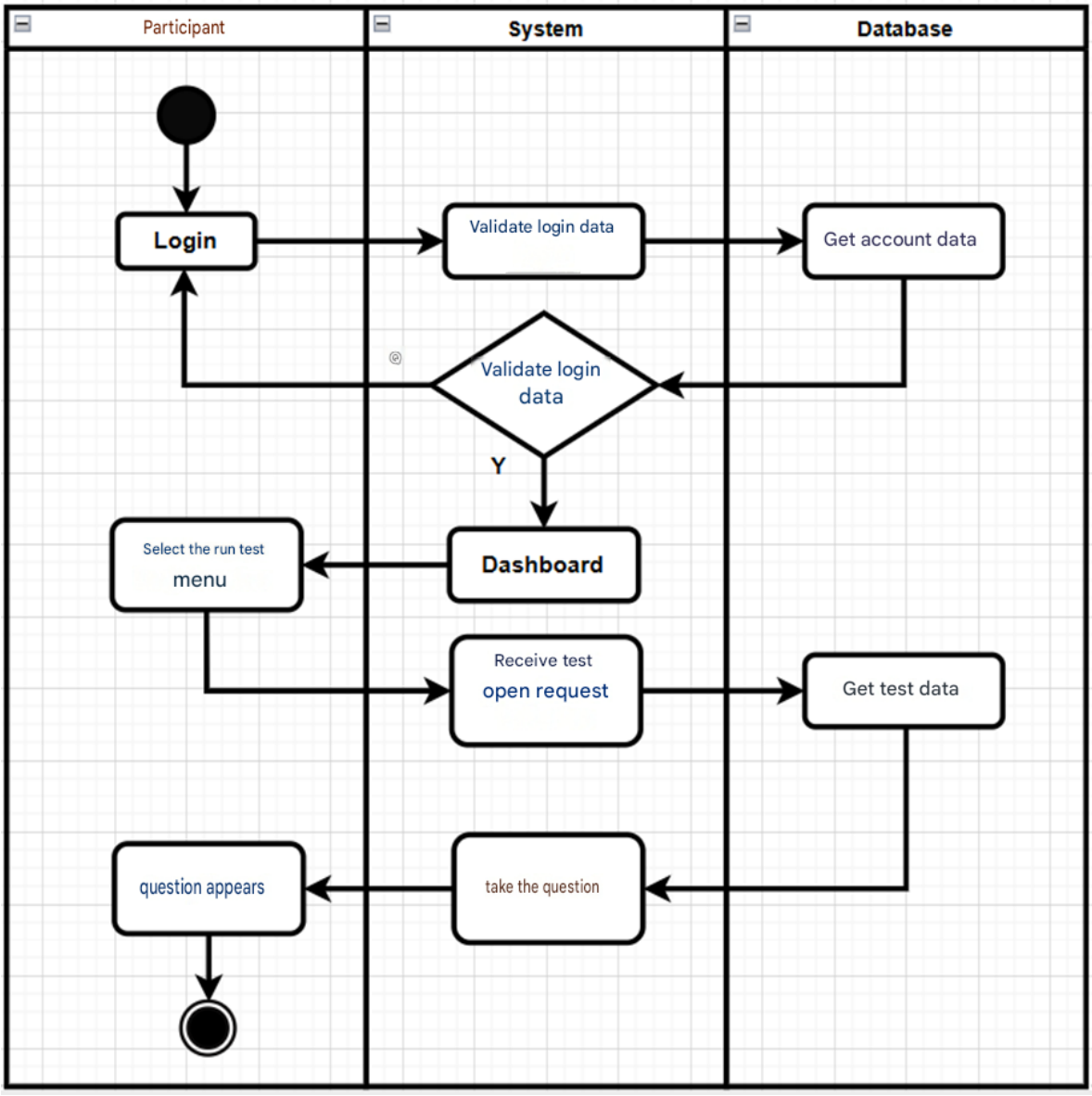

Figure 3. Test-Taking Activity

3.1.3 Sequence Diagram

A Sequence Diagram (in **Figure 4**) plays a role in visualizing interactions between objects within a use case, while also serving as an instrument to model the workflow logic within a system operation, function, or procedure [17].

The 'Manage Reports' Sequence Diagram illustrates the Admin's workflow in processing report data, beginning with entering a username and password to be validated by the System through an account data check in the Database. Once the login is successful, the system displays the dashboard page, allowing the Admin to select the report management menu, which triggers the system to retrieve and display report data from the database. The Admin then performs report data

management (add, edit, or delete), where each change is saved back to the database by the system and concludes with a notification as confirmation that the data has been successfully updated.

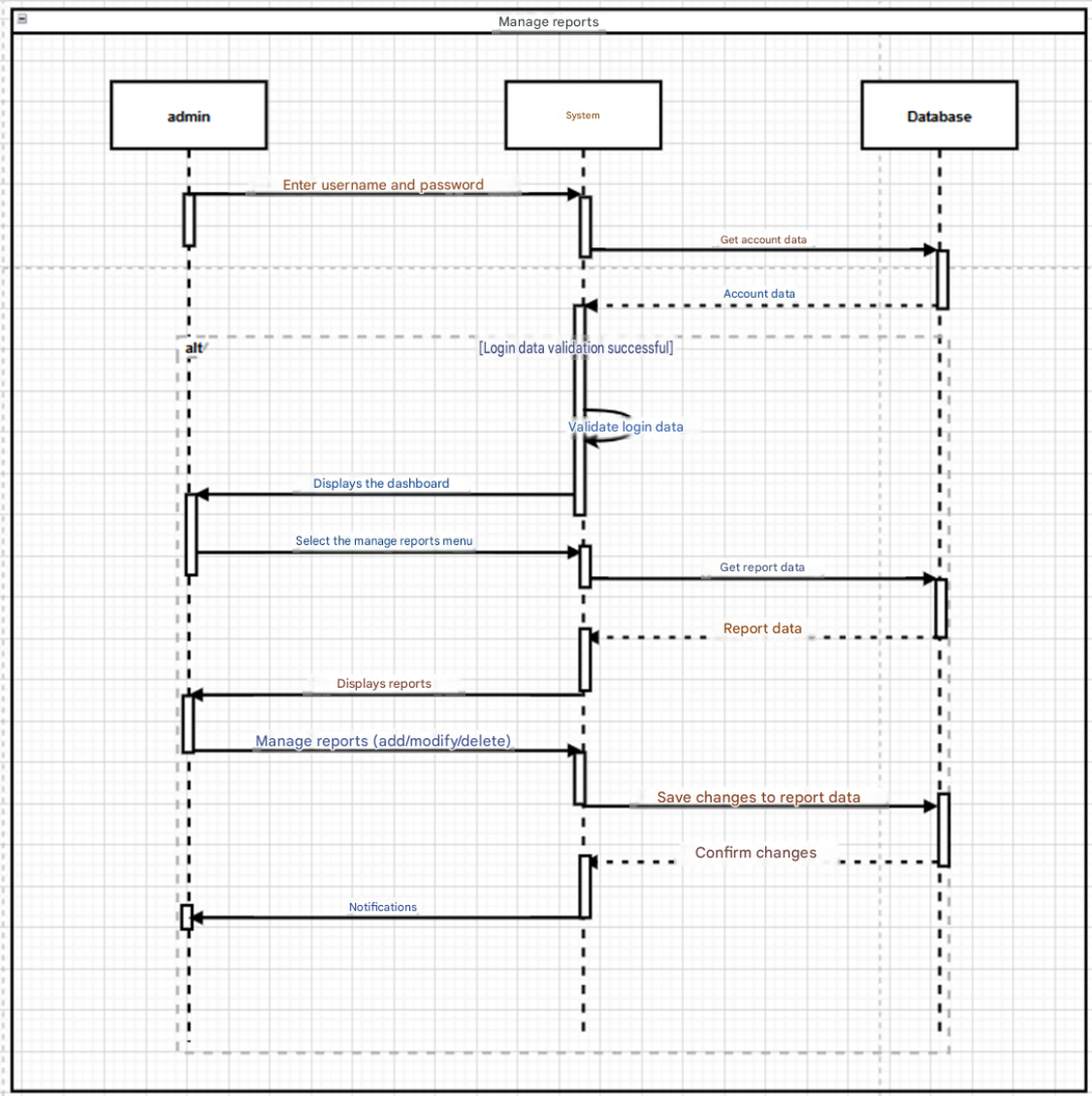

Figure 4. Sequence Diagram Manage Reports

3.1.4 Data Base

A database is defined as a collection of structured information that maintains interconnections or relations between its elements and is managed within a unified storage system [18].

The database design (in **Figure 5**) for this Try Out application utilizes an integrated relational model to manage the entire digital examination ecosystem. This structure maps the interconnections between user management through the user and user_profile tables, the academic grouping of students in the angkatan (batch) and kelas (class) tables, and the core of the examination system involving the topik, naskah (manuscript), and soal (question) tables, which are interconnected with the jawaban (answer) table. Furthermore, transactional tables such as tryout and

naskah_tryout are present to regulate exam schedules and durations, ultimately culminating in the nilai_tryout table as a centralized and systematic repository for students' academic evaluation results.

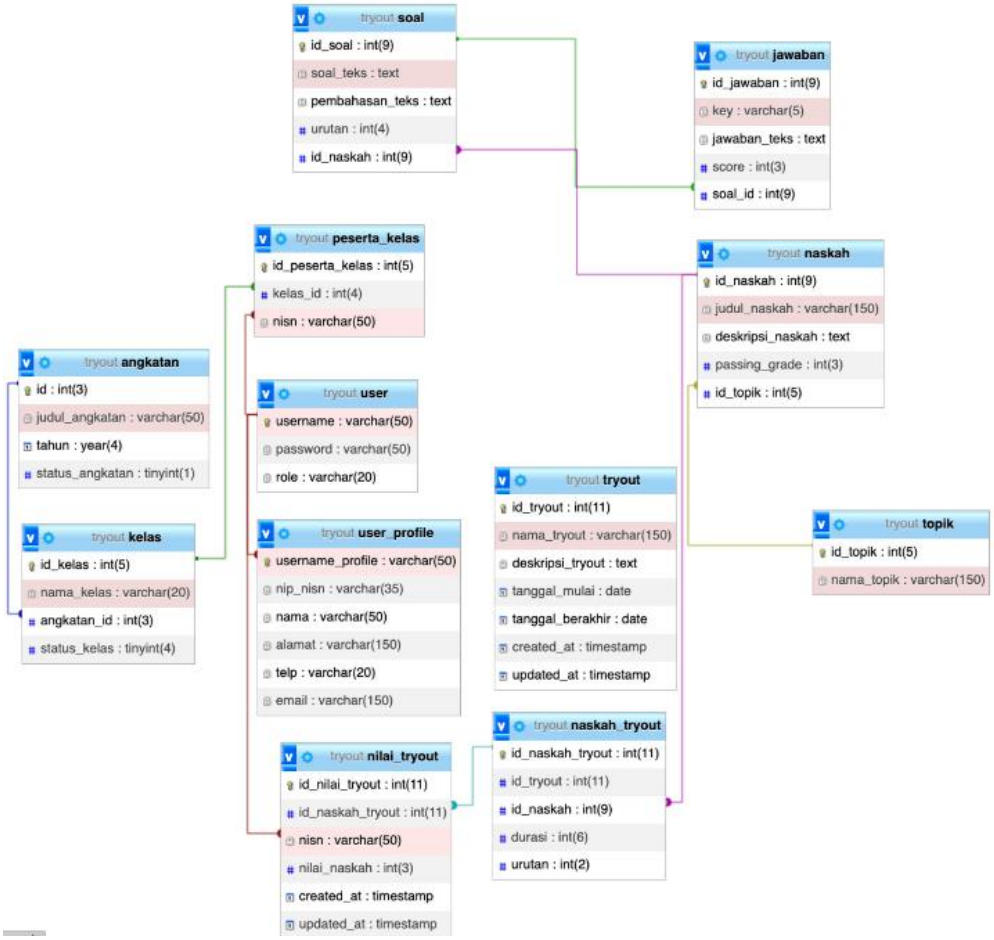

Figure 5. Data Base system management

3.2 Implementation

a. Dashboard interface

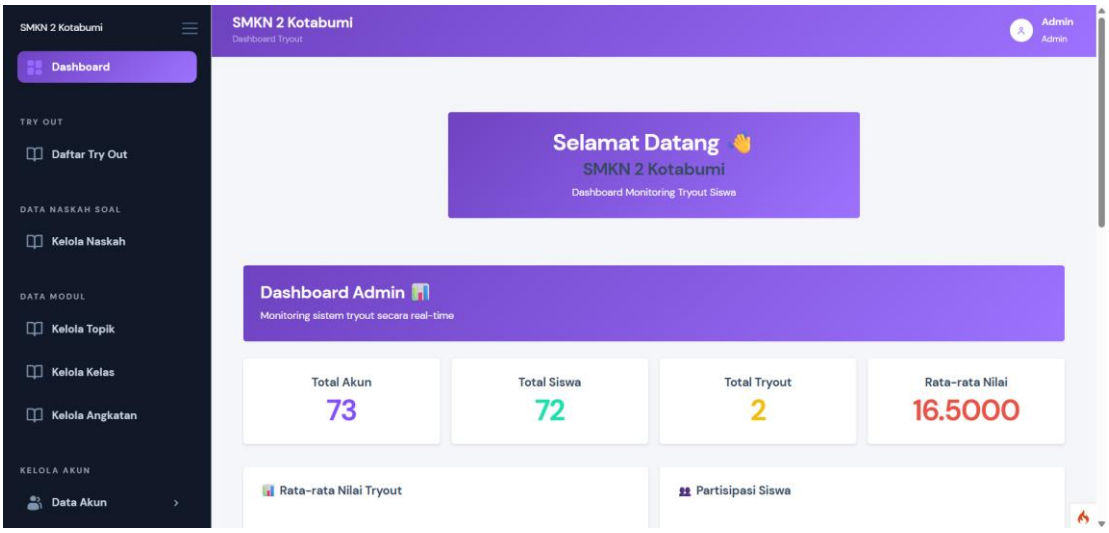

Figure 6. Dashboard page interface

The Admin Monitoring Dashboard (in **Figure 6**) for the SMKN 2 Kotabumi Tryout serves to monitor and oversee the execution of the examination process.

b. Tryout List Page

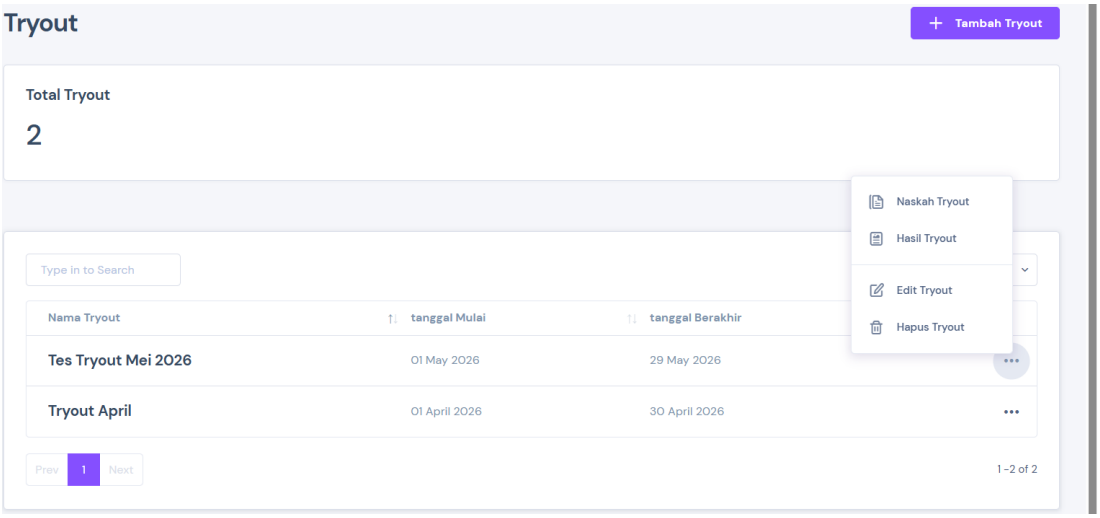

Figure 7. Tryout List Page interface

The Tryout List page (in **Figure 7**) displays a summary of the examination agenda currently being managed, featuring primary information such as the total of two available tryout packages. The tryout list table enumerates specific details, including names and the date range for the start and end of the examination period. The administrator maintains full control through the 'Add Tryout' button as well as interaction menus for each item to view manuscripts, review results, perform edits, or delete tryout sessions according to operational requirements.

c. Tryout Test Page

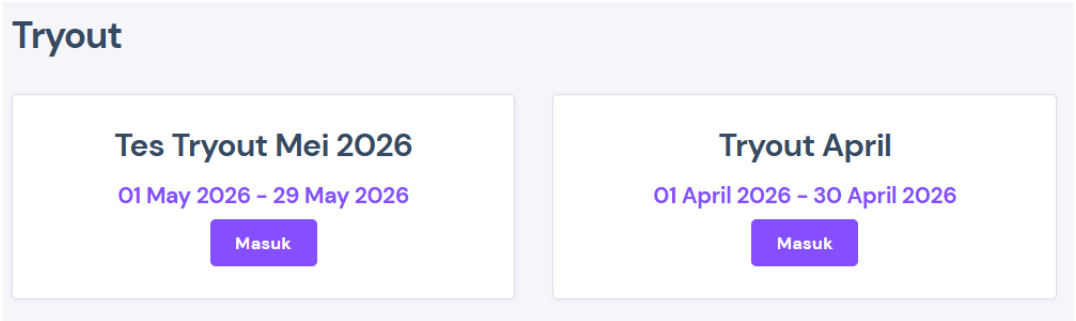

Figure 8. Tryout Test page

The Tryout Test page (in **Figure 8**) displays a series of examination cards available for students to complete, encompassing information on session names and their respective implementation periods. There are two active options: 'Tes Tryout Mei 2026,' running from May 1 to May 29, 2026, and 'Tryout April,' scheduled from April 1 to April 30, 2026. Each card is equipped

with a purple 'Enter' button, which serves as direct access for participants to commence the examination according to the predetermined schedule.

d. Report Page

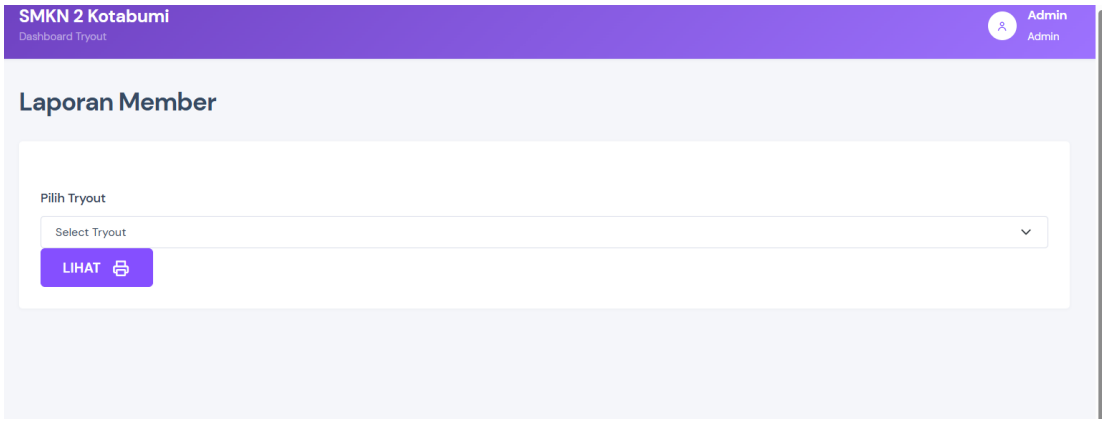

Figure 9. Report Page Interface

The Report Page (in **Figure 9**) serves as an interface for administrators to access specific recapitulations of student examination results. Through the 'Select Tryout' dropdown menu, users can specify a particular exam session to be reviewed, such as 'Tes Tryout Mei 2026', and then click the 'VIEW' button to display participant performance data. This feature facilitates the school in documenting scores and evaluating student learning achievements from each implemented tryout agenda.

3.3 Testing

The implementation of black-box testing techniques aims to verify that the system is capable of operating optimally and consistently across various device infrastructures and in addressing diverse use-case scenarios [19]. The selection of the Black-Box Testing method in this study is based on a testing focus that emphasizes the external functionality of the system. This method was chosen due to its relevance in evaluating whether each feature and module within the inventory system—particularly the QR Code scanning mechanism and asset lending management—operates and generates outputs that align with user requirements. Through this approach, testing is conducted purely based on input-output scenarios without involving an analysis of the internal code structure or the complexity of the program architecture. This ensures that the proposed system is not only technically functional but also adaptive and ready for deployment by end-users.

3.3.1 Black Box Testing

To evaluate the functionality and reliability of the developed system, Black Box Testing was conducted on the main features available to administrators and students. This testing method focuses on examining system functionality by providing various inputs and observing the resulting outputs without considering

the internal program structure or source code. The objective of the testing is to ensure that each feature operates according to its specified requirements and produces the expected outcomes. The test scenarios include dashboard access, navigation menus, tryout management, report generation, and student exam participation. The results of the Black Box Testing are presented in **Tabel 1**.

Tabel 1. Black Box Testing

No	Feature/ Component	Test Scenario	Expected Result	Status
1	Dashboard Admin	Accessing the main admin dashboard page.	The system accurately displays statistics for total accounts, students, tryouts, and average scores.	Succses
2	Navigasi Sidebar	Clicking the "Manage Manuscript" or "Manage Class" menu.	The system directs the administrator to the relevant data management page.	Succses
3	Add Tryout	Clicking the "+ Add Tryout" button.	A form appears to input data for a new tryout session.	Succses
4	Tryout Item Action	Selecting the three-dot menu on the tryout list..	The system displays options: Tryout Manuscript, Results, Edit, and Delete.	Succses
5	Report Filter	Selecting an exam session from the dropdown and clicking "VIEW".	The system processes the request and displays report data according to the selected session.	Succses
6	Student Exam Access	Clicking the "Enter" button on one of the tryout cards.	The student is directed to the question-answering page according to the applicable schedule.	Succses

3.3.2 White Box Testing

White-box testing was conducted to evaluate the internal logic, control flow, and code structure of the web-based SBMPTN try-out system. Unlike black-box testing, which focuses on external outputs, white-box testing examines how program logic is executed through conditions, loops, and internal decision structures.

In this study, white-box testing was applied to key modules, including user authentication, question management, tryout execution, scoring process, result reporting, and dashboard visualization. The testing method used was basis path testing supported by cyclomatic complexity analysis to determine the number of independent execution paths in each module, as shown in **Tabel 2**.

Cyclomatic complexity was calculated using the formula:

$$V(G) = P + 1$$

Where P represents the number of decision points in the program logic. This metric is used to measure the complexity level of each module and determine the minimum number of test cases required for full path coverage.

Tabel 2. Cyclomatic Complexity Result

No	Module	Predicate Nodes	Cyclomatic Complexity V(G)	Independent Paths	Result
1	Login Authentication	3	4	4	Passed
2	Question Management	5	6	6	Passed
3	Tryout Execution	4	5	5	Passed
4	Scoring System	4	5	5	Passed
5	Report Generation	3	4	4	Passed
6	Dashboard Visualization	3	4	4	Passed

Based on the results, all modules show moderate complexity levels (V(G) = 4–6), indicating that the system logic is structured, maintainable, and does not contain excessive branching complexity.

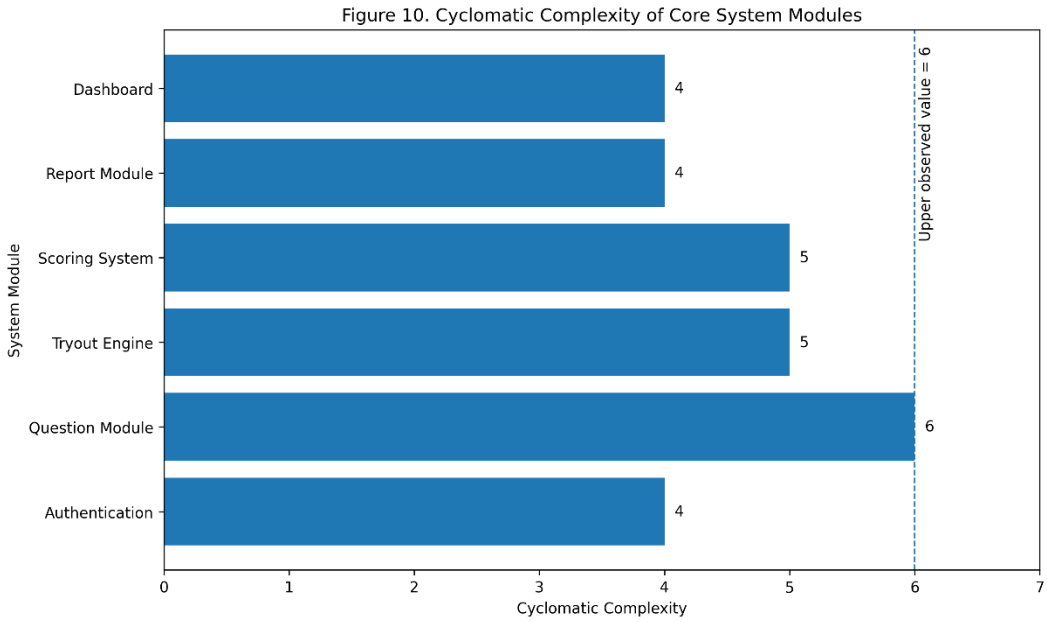

Figure 10. Cyclomatic Complexity of Core System Modules

Figure 10 illustrates the cyclomatic complexity distribution across the main modules of the SBMPTN try-out system. The results show that the Question Module has the highest complexity value of 6, followed by the Tryout Engine and Scoring System with values of 5. Meanwhile, the Authentication, Report Module, and Dashboard each show a lower complexity value of 4.

The higher complexity in the Question Module is caused by multiple decision points related to question selection, validation logic, and dynamic data retrieval. Similarly, the Tryout Engine and Scoring System involve conditional logic for time management, answer evaluation, and score calculation processes.

Overall, all modules remain within the moderate complexity range (V(G) ≤ 6), indicating that the system logic is still structured and maintainable. The presence of controlled complexity suggests that the system does not suffer from excessive branching, which reduces the risk of logical errors during execution and improves long-term maintainability.

3.3.3 Code Coverage Analysis

To assess the quality and completeness of the software testing process, code coverage analysis was conducted on the main modules of the system. Code coverage measures the extent to which the source code is executed during testing and provides an indication of how thoroughly the application has been validated. This evaluation includes three coverage metrics: Statement Coverage, which measures the percentage of executable statements that have been tested; Branch Coverage, which evaluates whether all decision outcomes have been executed; and Path Coverage, which determines the extent to which possible execution paths have been traversed. Higher coverage percentages indicate a greater level of testing completeness and help reduce the likelihood of undiscovered defects. The results of the code coverage analysis for each system module are presented in **Tabel 3**.

Tabel 3. Code Coverage Result

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

Overall, the code coverage results indicate that the system has achieved a consistently high level of test coverage, with all modules obtaining coverage values above 85% across statement, branch, and path testing metrics. This demonstrates that the majority of the application's source code, decision branches, and execution paths have been thoroughly exercised during the testing process. Such results reflect strong software reliability, functional completeness, and a low probability of undetected logical errors in the tested modules. Furthermore, the high coverage values suggest that the implemented testing strategy was effective in validating the core functionalities of the system, including authentication, data management, reporting, and dashboard visualization features. Therefore, the system can be

considered to have a robust testing quality and is adequately prepared for operational deployment.

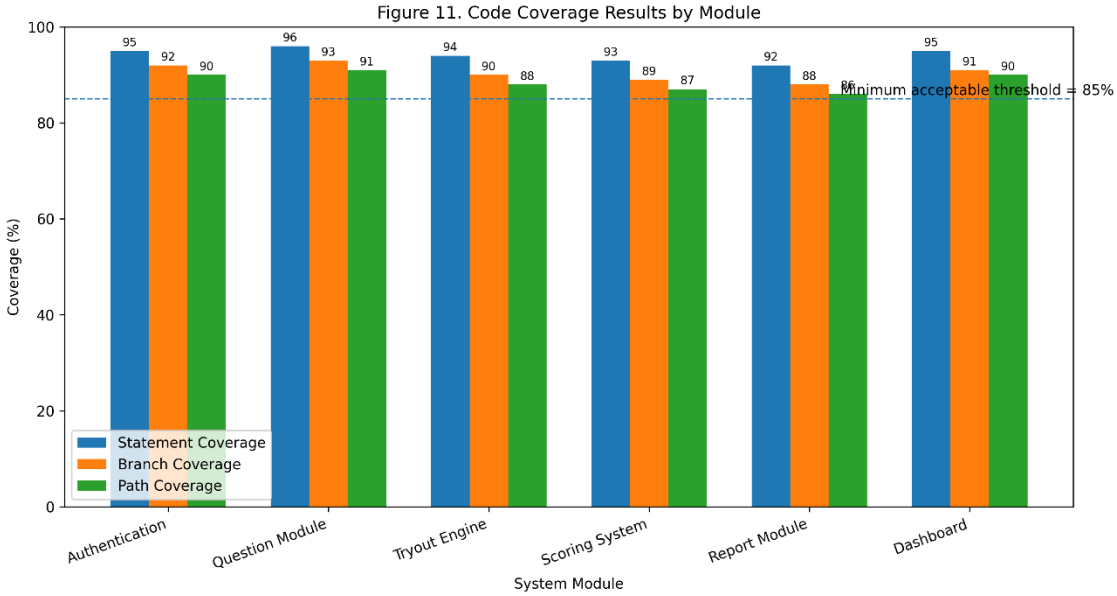

Figure 11. Code Coverage Results by Module

Figure 11 presents the code coverage analysis of the system, including statement coverage, branch coverage, and path coverage across all modules. The results indicate that the Question Module achieves the highest coverage values, with 96% statement coverage, 93% branch coverage, and 91% path coverage. In contrast, the Report Module shows relatively lower values but still maintains coverage above 85%.

The high coverage in the Question Module indicates that most execution paths, including edge cases, have been tested thoroughly. This is important because the module handles dynamic question retrieval and user interaction logic. Meanwhile, the slightly lower coverage in the Report Module is influenced by limited branching scenarios in report generation logic, which depends on predefined dataset conditions.

Overall, all modules exceed the minimum acceptable threshold of 85%, which indicates that the system has undergone comprehensive testing. These results confirm that the application has a high level of reliability, with most statements, branches, and execution paths successfully validated.

3.3.4 Traceability Matrix (Requirement–Testing Alignment)

To ensure alignment between system requirements and testing outcomes, a traceability matrix was constructed. The integration of black-box and white-box testing provides comprehensive validation from both user perspective and internal system logic. Black-box testing confirms that the system meets functional expectations, while white-box testing ensures that internal logic paths are correctly implemented and free from critical structural errors. Cyclomatic complexity results

indicate that all modules remain within acceptable complexity thresholds ($V(G) \leq 6$), suggesting maintainability and low risk of logical instability. Meanwhile, code coverage above 85% demonstrates that most execution paths have been tested, reducing the probability of hidden runtime defects, as shown in **Tabel 4**. Overall, the system exhibits high functional reliability, stable internal architecture, and strong readiness for deployment in educational environments.

Tabel 4. Requirement Traceability

Requirement	Module	Test Type	Result
User authentication	Login System	Black-box / White-box	Valid
Tryout execution	Question Module	Black-box / White-box	Valid
Automatic scoring	Scoring System	Black-box / White-box	Valid
Result reporting	Report Module	Black-box	Valid
Data management	Admin Panel	Black-box	Valid
Requirement	Module	Test Type	Result
User authentication	Login System	Black-box / White-box	Valid

4. Conclusion

Based on the testing and analysis results, it can be concluded that the development of this web-based tryout monitoring system has successfully provided an integrated platform for examination management at SMK N 2 Kotabumi. The implementation of real-time dashboard features, manuscript management, and automated reporting has proven to facilitate administrators in monitoring student score progression efficiently. Black-box testing results indicate that all system functions, ranging from exam package registration to student access, operate in accordance with the established functional requirements. Consequently, this system is deemed suitable for use as a digital tool to optimize students' academic preparation through evaluation mechanisms that are more structured and transparent.

The white-box testing results indicate that all independent execution paths were successfully validated without logical errors. The authentication module correctly handles login validation paths, the question module processes branching logic for selection and retrieval, and the scoring module accurately executes calculation and storage logic. Overall, the system demonstrates stable internal logic and is suitable for academic tryout implementation.

Acknowledgement

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

Authors' Declaration

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

Funding - No funding information from the authors.

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

Competing interests - The authors declare no competing interest.

Additional information - No additional information from the authors.

References

[1] J. Tahsinia, A. A. Zulfa, T. Ibrahim, dan O. Arifudin, "PERAN SISTEM INFORMASI AKADEMIK BERBASIS WEB," J. Tahsinia, vol. 6, no. 1, hal. 115–134, 2025.

[2] A. P. Syalsabella dan A. Prapanca, "Perancangan Website Online Course (KitaPTN) untuk Persiapan UTBK Menggunakan Algoritma Fisher-Yates Shuffle," vol. 06, hal. 581–588, 2024.

[3] M. Agarina, A. S. K, Sutedi, M. R. F. M, H. Purnomo, dan Reyhandon, "Desain Sistem Computer AssistedTest(CAT) Pada Olimpiade Sains Nasional (OSN) Tingkat Provinsi Lampung Berbasis Website dengan Metode SCRUM," vol. 5, no. 3, hal. 260–267, 2025.

[4] Z. Apriza, S. Mujiranda, dan T. Sutabri, "Optimasi Pemilihan Soal Ujian Try Out Menggunakan Algoritma Greedy Berdasarkan Tingkat Kesulitan dan Waktu Pengerjaan," vol. 5, no. 1, hal. 141–152, 2025.

[5] E. Lakilaki, A. F. Salsabila, P. Wijaya, dan M. Fayyad, "Eksplorasi Peran Gamifikasi Dan Motivasi Dalam Dinamika Utbk Di Perguruan Tinggi Negeri Indonesia," vol. 4, hal. 2474–2492, 2024.

[6] F. Ridho dan M. Syahputra, "Perancangan Sistem Informasi atau Aplikasi Monitoring Absensi Karyawan pada PT . Socfindo Menggunakan QR Code Berbasis Web," vol. 1, no. 3, hal. 36–48, 2024.

[7] A. M. Elhanafi, S. Informasi, dan U. H. Medan, "PERANCANGAN DAN IMPLEMENTASI APLIKASI TRYOUT ONLINE BERBASIS DESIGN AND IMPLEMENTATION OF A WEB-BASED ONLINE TRYOUT APPLICATION WITH INTEGRATED QUESTION MANAGEMENT AND," vol. 4, no. 3, hal. 238–245, 2025.

[8] Zulkifli, M. KH, Ansar, dan M. Ikhsan, "Optimasi Proses Evaluasi Belajar melalui Pengembangan Sistem Ujian Komputerisasi di SMP Negeri 2 Majene," J. Digit. Ecosyst. Nat. Sustain., vol. 5, no. 1, hal. 16–21, 2025.

[9] A. Budiman dan I. Ariyati, "IMPLEMENTATION OF A QR CODE-BASED WEB INVENTORY SYSTEM AT PT SELAHONJE JAYA ABADI BEKASI," J. Sci. Res. Dev., vol. 8, no. 1, hal. 255–265, 2026.

[10] M. Sutrisno, A. D. Wiranata, dan D. Irawan, "PENGEMBANGAN APLIKASI

TRY OUT BERBASIS WEB UNTUK LEMBAGA KURSUS DAN PELATIHAN," vol. 14, no. 3, hal. 188–193, 2024.

[11] H. Agriani, I. A. M, dan M. A. Nur, "Pelatihan Mengerjakan Tes Potensi Skolastik Bagi Siswa Kelas XII SMA Untuk Menghadapi Seleksi Nasional Berdasarkan Tes (SNBT)," vol. 1, hal. 126–132, 2024.

[12] A. A. Sarita dan E. Imawati, "PENINGKATAN KETERAMPILAN MEMAHAMI TEKS LAPORAN HASIL OBSERVASI MENGGUNAKAN METODE DISKUSI SISWA KELAS VIII," vol. 1, no. 1, hal. 39–46, 2022.

[13] A. Rahmawati dkk., "Optimalisasi Teknik Wawancara Dalam Penelitian Field Research Melalui Pelatihan Berbasis Participatory Action Research Pada Mahasiswa Lapas Pemuda Kelas IIA Tangerang," J. Abdimas Prakasa Dakara, vol. 4, no. 2, hal. 135–142, 2024.

[14] A. Rustamana dkk., "Menanamkan Kesadaran Akan Jati Diri Sebagai Bangsa Indonesia dan Kesadaran Untuk Mempertahankan Jati Diri Bangsa di Tengah Globalisasi. Pengertian Integrasi Nasional, Integrasi Nasional dalam Masyarakat Pluralitas, Strategi Mewujudkan Integrasi Nasional," vol. 02, no. 04, hal. 186–194, 2025.

[15] T. Wijayanti, F. Nugraha, dan A. P. Utomo, "Rancang Bangun Sistem Manajemen Pengelolaan Pengaduan Masyarakat Di Kabupaten Kudus," vol. 3, no. 1, hal. 56–65, 2022.

[16] B. S. Mare dan A. A. Yana, "PERANCANGAN SISTEM INFORMASI BERBASIS WEB PADA KOPERASI SIMPAN PINJAM SEJAHTERA BERSAMA," vol. 11, no. 2, hal. 70–76, 2022.

[17] Y. Amanda dan M. H. Ujianti, "PERANCANGAN SISTEM INFORMASI PENERIMAAN PESERTA DIDIK BARU PADA DAYCARE DAN PRE SCHOOL ANANDA MANDIRI SLAWI BERBASIS WEB," JATI (Jurnal Mhs. Tek. Inform., vol. 9, no. 1, hal. 177–184, 2025.

[18] B. Haryanto, A. Ardiansyah, dan M. Kurniasih, "PENGENALAN DATABASE NOSQL DAN PERBANDINGANNYA DENGAN DATABASE RELASIONAL," vol. 12, no. 1, 2024.

[19] yohanes dwi Cahyono, linda wahyu Widianti, dan sulvi indah Pramuningsih, "PENGUJIAN BLACK BOX PADA WEBSITE SISTEM INFORMASI PENGOLAHAN DATA TERITORIAL MENGGUNAKAN METODE USE CASE TESTING UNTUK OPERASI MILITER," J. Mhs. Tek. Inform., vol. 9, no. 2, hal. 2250–2256, 2025.