

# Design And Development Of A Web-Based Examination Application Using Django Framework At SMA Negeri 7 Sarolangun

Afif Muhammad<sup>1\*</sup>, Thomson Mary<sup>2\*</sup>, Ami Anggraini Samudra<sup>3</sup>

<sup>1-2</sup> Program Studi Teknologi Informasi, Universitas PGRI Sumatera Barat

<sup>3</sup>Pendidikan Informatika, Universitas PGRI Sumatera Barat

<sup>1\*</sup>[afifm822@gmail.com](mailto:afifm822@gmail.com), <sup>2\*</sup>[thomsonmary1980@gmail.com](mailto:thomsonmary1980@gmail.com), <sup>3</sup>[amianggrainisamudra@gmail.com](mailto:amianggrainisamudra@gmail.com)

## ABSTRACT

The implementation of examinations at SMA Negeri 7 Sarolangun previously encountered operational challenges due to an unintegrated workflow, where exam authoring was conducted via Google Forms and exam delivery relied on an Android-based third-party application paired with a custom keyboard. This workflow caused severe device incompatibility, system crashes, and manual score recapitulation delays lasting up to one week. This research aims to design and develop an integrated web-based examination application utilizing the Django framework under the Rapid Application Development (RAD) model to support Midterm (PTS) and Final (PAS) semester exams. The application features master data management, manual question authoring and bulk imports via Excel, token-protected scheduling, asynchronous AJAX auto-saving on computer-based test (CBT) sheets, live proctoring controls, auto-grading, score visibility authorization (Hide Score), and dynamic class-filtered report exports in Excel and PDF formats. System evaluation was conducted through Alpha testing (White Box and Black Box testing) and Beta testing (ISO/IEC 25010:2011 quality evaluation by system experts and End-User Computing Satisfaction/EUCS by end-users). Alpha testing verified that all 29 independent logic paths in White Box testing executed without branching errors and Black Box testing achieved 100% validity across 13 functional modules. Beta testing with 2 system experts yielded an average score of 98.25% (Very Good), and end-user testing involving 27 respondents achieved an average satisfaction score of 84.75% (Very Good). Thus, the developed system is highly feasible, reliable, and effective for school examination operations.

**Keywords:** *Computer-Based Test (CBT), Django Framework, End-User Computing Satisfaction, ISO/IEC 25010, Rapid Application Development*

## INTRODUCTION

Learning evaluation in secondary education requires an assessment platform that guarantees accessibility, security, and operational efficiency [1]. At SMA Negeri 7 Sarolangun, semester examinations (Midterm Examination/PTS and Final Examination/PAS) involving 839 students and 73 teachers previously faced significant operational and fragmentation issues [2]. Teachers authored exam items using Google Forms and submitted the links to the examination committee, while students took the exams through a third-party Android application paired with a custom keyboard (*Simple Keyboard*) [3]. This process caused severe technical complications, including platform dependency on Android, device incompatibilities across various student smartphone types, background application crashes, slow loading times during unstable internet connections, and a prolonged manual score recapitulation process that took up to one week [4]. Furthermore, access rights were restricted strictly to administrators and students, preventing subject teachers from directly managing question repositories, configuring exams, or supervising student progress [5].

Computer-Based Testing (CBT) systems have been widely investigated to overcome conventional testing constraints. Refandy and Wahyuningtyas [6] developed a web-based CBT application for primary schools to automate scoring and reduce manual grading, yet their work utilized a traditional Waterfall model with limited single-role workflows. Siswanto and Morang [7] focused on backend performance optimization for concurrent student access using caching and queue jobs, but did not provide independent exam configuration features for teachers. Annurwan et al. [8] implemented a web-based online exam platform for madrasah students, yet it lacked automated asynchronous saving mechanisms (auto-save) to prevent data loss during network disruptions. Ambiyar et al. [9] developed an examination application using the 4D development model for hybrid platforms, but it omitted bulk authoring workflows. In addition, Rosa and Abad [10] demonstrated the capability of the Django framework in developing educational assessment tools evaluated under ISO/IEC 25010, although their implementation focused on general classroom quizzes rather than formalized institutional semester exams.

To address these limitations, this study presents an integrated, multi-role web-based examination application developed with the Django framework and MySQL database [11], [12]. The system incorporates master data management, Excel-based bulk question importing, token-protected scheduling, asynchronous AJAX auto-saving on CBT worksheets, real-time proctoring state intervention, exam visibility control (*Hide Score*), and dynamic class-filtered reporting in Excel and PDF formats [13]. This study aims to design and implement the system using the Rapid Application Development (RAD) model, verify its structural logic and functional validity via Alpha testing, and evaluate software product quality and end-user satisfaction through the ISO/IEC 25010:2011 standard and the End-User Computing Satisfaction (EUCS) method.

## METHODOLOGY

This study employed the Research and Development (R&D) methodology with the Rapid Application Development (RAD) software engineering life cycle [12]. The RAD model was chosen due to its iterative structure and continuous user involvement, which enables rapid adaptation to institutional requirements [12]. The research phases are illustrated in Figure 1.



Figure 1. Rapid Application Development (RAD) Research Stages [12]

### Requirements Planning

The requirements planning stage was conducted through field observations and structured interviews with stakeholders at SMA Negeri 7 Sarolangun. This phase identified the core problems of the legacy system and formulated functional requirements across three distinct user roles: Administrator (full data control and global proctoring), Teacher (question authoring, exam configuration, and class-level reporting), and Student (CBT delivery, token access, and score review).

Non-functional requirements were established to guarantee cross-platform compatibility across desktop and mobile browsers, real-time data persistence, and security during live examinations.

### Design Workshop

The design workshop translated requirements into system architecture models utilizing Unified Modeling Language (UML) diagrams (Use Case Diagram, Activity Diagrams, Class Diagram, and Sequence Diagrams), relational database schemas, and responsive user interface wireframes. The database was structured into 11 normalized tables with the *cbt* namespace prefix managed under MySQL 8.4. Responsive user interfaces were designed utilizing Bootstrap 5 to guarantee accessibility on various student smartphone screen dimensions.

### System Implementation and Testing

System implementation was executed using Python 3.12 and the Django 6.0.8 framework on an AlmaLinux 8 enterprise server running Gunicorn WSGI. System evaluation was conducted using two testing paradigms:

#### 1. Alpha Testing

Evaluates internal code logic via White Box testing (Basis Path Testing with Cyclomatic Complexity) on 8 critical algorithmic modules, and evaluates functional input-output behavior via Black Box testing across 13 operational features [13].

#### 2. Beta Testing

Product quality evaluation by 2 system experts based on 5 characteristics of the ISO/IEC 25010:2011 standard (*Functional Suitability, Performance Efficiency, Usability, Reliability, and Security*), alongside user satisfaction evaluation involving 27 respondents (1 Administrator, 4 Teachers, and 22 Students) using the 5 dimensions of the End-User Computing Satisfaction (EUCS) framework (*Content, Accuracy, Format, Ease of Use, and Timeliness*).

The functional validity percentage in Black Box testing was calculated using (1):

$$P = \left( \frac{\text{Number of Successful Test Cases}}{\text{Total Test Cases Evaluated}} \right) \times 100\% \quad (1)$$

The percentage of expert quality and user satisfaction ratings from Likert-scale questionnaires was determined using (2):

$$Y = \left( \frac{\sum(N \times R)}{\text{Ideal Score}} \right) \times 100\% \quad (2)$$

where N represents the Likert scale weight (1 to 4), R represents the response frequency, and the Ideal Score represents the product of total questionnaire items, maximum scale weight (4), and total respondents.

## RESULTS AND DISCUSSION

### System Design and Use Case Modeling

To model the dynamic behavioral interactions, functional abstractions, and role-based access boundaries of the examination platform, a Use Case Diagram was formulated in strict compliance with OMG UML standards, as depicted in Figure 2.

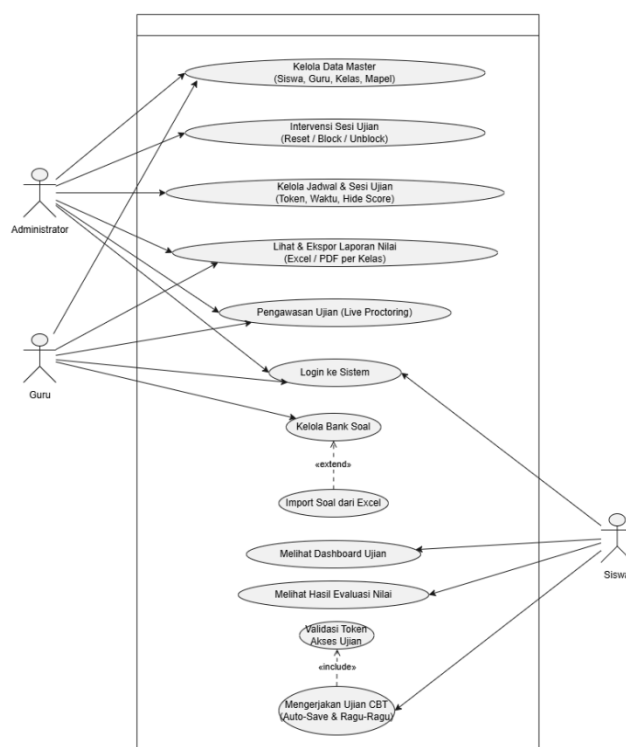


Figure 2. Use Case Diagram

As modeled in Figure 2, the use case architecture strictly avoids depicting sequential execution steps as functional dependency chains. Chronological workflows are governed by activity logic, whereas use case associations represent user goals and direct capabilities:

### 1. Administrator

Directly associates with administrative use cases: Manage Master Data (Users, Classes, and Subjects), Manage Global Exam Schedules, Intervene Student Sessions (Reset / Unblock), and View and Download Comprehensive Reports.

### 2. Teacher

Directly associates with Manage Question Bank, Configure Exam Schedules, Monitor Classroom Proctoring, and Export Filtered Grade Reports. The optional bulk import functionality, Import Questions (Excel), is modeled with an «extend» dependency directed from the extension module (Import Questions (Excel)) toward the base use case (Manage Question Bank), correctly denoting an optional extended capability.

### 3. Student

Directly associates with independent user goals: View Exam Dashboard, Take Computer-Based Examination, and View Exam Results. Token validation is encapsulated within the execution boundary of Take Computer-Based Examination via an «include» relationship (Validate Token), ensuring correct behavioral encapsulation without forming sequential dependency chains.

## System Implementation Results

The web-based examination application was successfully built and deployed across multi-role interfaces. The user authentication interface shown in Figure 3 acts as the single-entry gateway that validates user credentials against the MySQL database and dynamically routes users to their specific role-based dashboard.

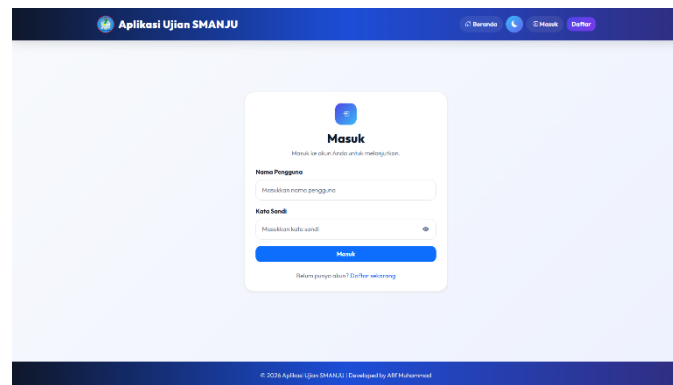


Figure 3. User Authentication and Login Interface

For exam item preparation, the interface depicted in Figure 4 enables teachers to author questions manually with image attachments or perform bulk imports from Excel spreadsheets (.xlsx) using the *openpyxl* library.

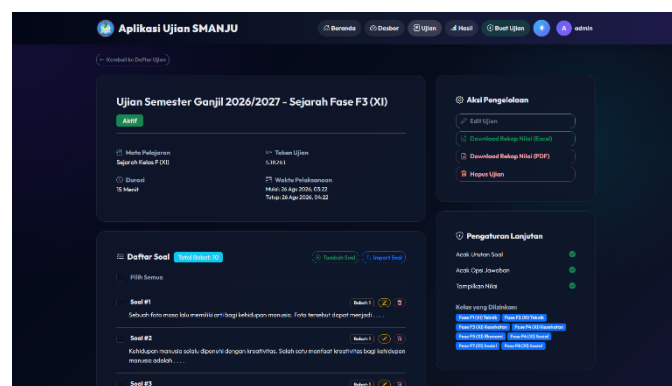


Figure 4. Question Bank Management and Excel Bulk Import Interface

Figure 5 presents the examination scheduling interface where teachers configure exam parameters, including secret access tokens, active date/time windows, allocated duration, target classes, option/question randomization, and the *Hide Score* privacy toggle.

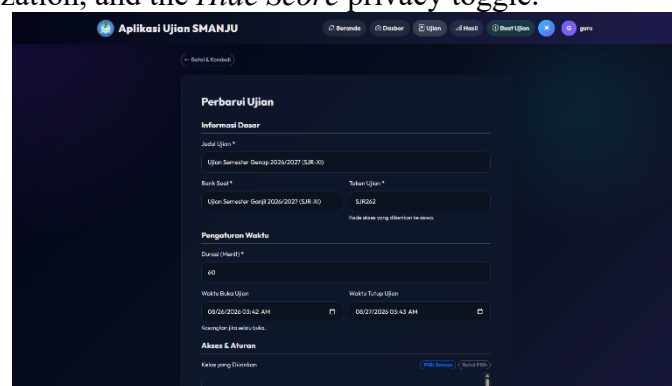


Figure 5. Examination Configuration and Scheduling Interface

The core student Computer-Based Test (CBT) delivery interface is shown in Figure 6. The interface is equipped with a real-time countdown timer, question navigation palette, doubtful question flags, and asynchronous AJAX auto-saving to record answers instantly without full page reloads.

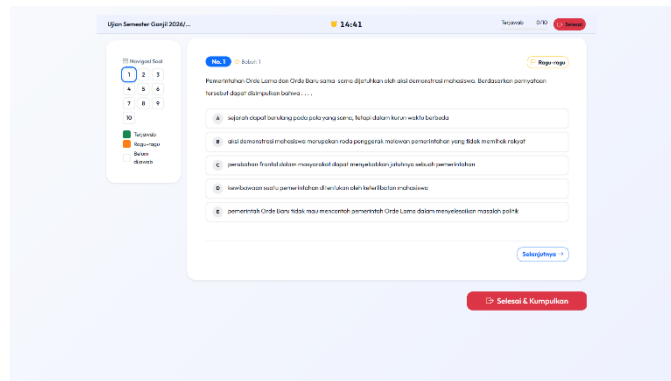


Figure 6. Computer-Based Test (CBT) Student Examination Interface

During live exam sessions, teachers and administrators utilize the live proctoring interface in Figure 7 to monitor student statuses (Ongoing, Submitted, or Blocked) and perform real-time session interventions (Force Submit, Block, or Reset Session).

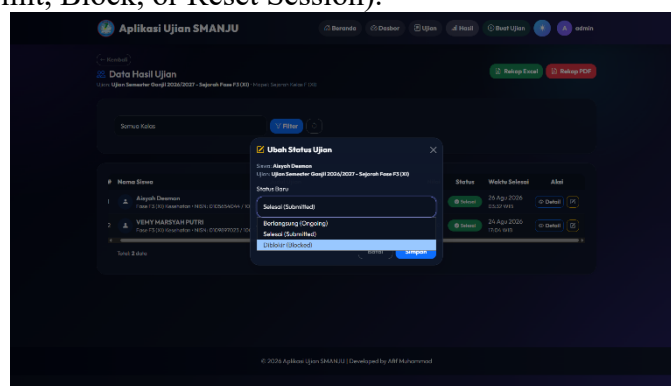


Figure 7. Live Examination Monitoring and Proctoring Interface

Following exam completion, grades are automatically calculated via the auto-grading algorithm. Figure 8 illustrates the reporting interface where teachers can filter results by classroom cohort and export comprehensive score sheets in spreadsheet (.xlsx) or printable PDF format via *xhtml2pdf*.

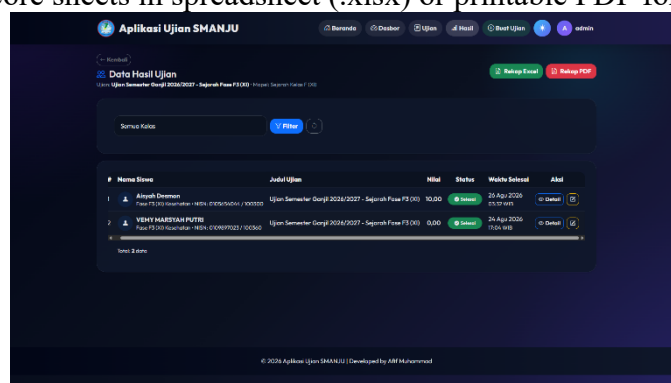


Figure 8. Grade Recapitulation and Dynamic Report Export Interface

## Alpha Testing Result

Alpha testing was conducted internally to verify algorithmic correctness and functional stability:

### 1. White Box Testing (Basis Path Testing)

White Box testing evaluated 8 critical algorithmic modules by constructing control flowgraphs, determining predicate nodes (P), and calculating Cyclomatic Complexity ( $V(G) = E - N + 2 = P + 1$ ) to identify all linearly independent execution paths. As summarized in Table 1, across all 8 modules containing 68 nodes and 81 edges, all 29 independent execution paths were executed

without branching anomalies, unhandled exceptions, or infinite loops, achieving a 100% path coverage rate.

Table 1. Summary of White Box Basis Path Testing Metrics

| No           | Critical Functional Module                    | Nodes<br>(N) | Edges<br>(E) | Predicates<br>(P) | V(G)      | Independent<br>Paths         |
|--------------|-----------------------------------------------|--------------|--------------|-------------------|-----------|------------------------------|
| 1            | User Authentication & Role-Based Redirect     | 11           | 14           | 4                 | 5         | 5 Paths                      |
| 2            | Bulk Import Question Bank via Excel           | 10           | 12           | 3                 | 4         | 4 Paths                      |
| 3            | Exam Access & Secret Token Validation         | 9            | 11           | 3                 | 4         | 4 Paths                      |
| 4            | Asynchronous AJAX Auto-Save Mechanism         | 8            | 9            | 2                 | 3         | 3 Paths                      |
| 5            | Automated Scoring Algorithm (Auto-Grading)    | 9            | 11           | 3                 | 4         | 4 Paths                      |
| 6            | Class-Filtered Grade Recapitulation           | 7            | 8            | 2                 | 3         | 3 Paths                      |
| 7            | Score Visibility Authorization (Hide Score)   | 6            | 6            | 1                 | 2         | 2 Paths                      |
| 8            | Proctoring State Manipulation & Session Reset | 8            | 10           | 3                 | 4         | 4 Paths                      |
| <b>Total</b> | <b>Metric Summary / Full Path Coverage</b>    | <b>68</b>    | <b>81</b>    | <b>21</b>         | <b>29</b> | <b>29 Paths (100% Valid)</b> |

## 2. Black Box Testing

Black Box testing evaluated 13 primary functional modules encompassing user authentication, master data CRUD, manual/bulk question authoring, tokenized scheduling, CBT delivery, AJAX auto-save, doubtful flags, live proctoring, Hide Score toggles, dynamic reporting, countdown force submission, and session logout. All 13 test cases produced outputs strictly matching expected functional specifications without exceptions, achieving a 100% functional validity rate (P = 100%, Valid).

## Beta Testing Results

Beta testing was conducted to evaluate software product quality from technical experts and user satisfaction from operational end-users.

### 1. System Experts Quality Evaluation (ISO/IEC 25010:2011 Standard)

Technical quality evaluation was conducted by 2 system experts across 5 ISO/IEC 25010:2011 quality characteristics through a 15-item questionnaire. The recapitulation of expert ratings is presented in Table 2.

Table 2. System Expert Quality Evaluation Results (ISO/IEC 25010:2011)

| No             | Criteria / Quality Characteristic      | Total Score | Ideal Score | Percentage (%) | Interpretation   |
|----------------|----------------------------------------|-------------|-------------|----------------|------------------|
| 1              | Functional Suitability                 | 39          | 40          | 97.50%         | Very Good        |
| 2              | Performance Efficiency                 | 24          | 24          | 100.00%        | Very Good        |
| 3              | Usability / Interaction Capability     | 24          | 24          | 100.00%        | Very Good        |
| 4              | Reliability                            | 15          | 16          | 93.75%         | Very Good        |
| 5              | Security                               | 16          | 16          | 100.00%        | Very Good        |
| <b>Average</b> | <b>Overall Product Quality Average</b> | <b>118</b>  | <b>120</b>  | <b>98.25%</b>  | <b>Very Good</b> |

As shown in Table 2, the system achieved an overall product quality score of 98.25% (Very Good category). Three characteristics Performance Efficiency, Usability, and Security achieved perfect scores of 100.00%. Functional Suitability obtained 97.50%, confirming that the scoring and bulk import workflows operated accurately. Reliability achieved 93.75%, where experts noted that potential network disconnections during live testing are well mitigated by client-side retry handling and administrative session reset capabilities.

## 2. End-User Satisfaction Evaluation (EUCS Method)

User satisfaction testing was conducted involving 27 respondents (1 Administrator, 4 Teachers, and 22 Students) at SMA Negeri 7 Sarolangun across 5 EUCS dimensions using a 15-item questionnaire. The evaluation results are summarized in Table 3.

Table 3. End-User Computing Satisfaction (EUCS) Evaluation Results

| No             | EUCS Dimension                           | Obtained Score | Ideal Score | Percentage (%) | Interpretation   |
|----------------|------------------------------------------|----------------|-------------|----------------|------------------|
| 1              | Content                                  | 280            | 324         | 86.42%         | Very Good        |
| 2              | Accuracy                                 | 272            | 324         | 83.95%         | Very Good        |
| 3              | Format                                   | 271            | 324         | 83.64%         | Very Good        |
| 4              | Ease of Use                              | 272            | 324         | 83.95%         | Very Good        |
| 5              | Timeliness                               | 278            | 324         | 85.80%         | Very Good        |
| <b>Average</b> | <b>Overall User Satisfaction Average</b> | <b>1373</b>    | <b>1620</b> | <b>84.75%</b>  | <b>Very Good</b> |

As detailed in Table 3, end-user satisfaction reached an overall average of 84.75% (Very Good category). The highest ratings were achieved in the Content dimension (86.42%) and Timeliness dimension (85.80%), demonstrating that the platform provides complete examination information and successfully transforms score recapitulation from a one-week delay into instant, real-time automated reporting. The Accuracy (83.95%), Ease of Use (83.95%), and Format (83.64%) dimensions confirmed that the responsive UI and AJAX auto-save provide a smooth and trustworthy testing experience across diverse student mobile devices.

## Discussion

The experimental findings confirm that migrating to a Django-based web platform successfully resolves the core operational bottlenecks at SMA Negeri 7 Sarolangun. The asynchronous AJAX auto-save mechanism provides robust fault tolerance, ensuring answer data persistence even during abrupt device shutdowns or intermittent signal loss. The bulk question import feature significantly minimizes teacher preparation workload, while the live proctoring dashboard empowers educators to monitor and intervene during examination anomalies independently. Furthermore, replacing Android-exclusive third-party APKs with an open web-browser architecture eliminates device incompatibility issues and background application crashes, enabling seamless cross-platform participation.

## CONCLUSION

The web-based examination application for SMA Negeri 7 Sarolangun was successfully designed and developed using the Django framework and MySQL under the Rapid Application Development (RAD) lifecycle. The application integrates master data management, Excel bulk question importing, tokenized scheduling, responsive CBT worksheets with AJAX auto-saving, real-time proctoring, auto-grading, and dynamic multi-format grade report exports. Alpha testing verified the internal logic and functional validity with a 100% success rate across White Box (29 independent paths) and Black Box evaluations (13 functional modules). Beta testing confirmed high software quality, scoring 98.25% (Very Good) under ISO/IEC 25010:2011 expert evaluation, and high user acceptance, scoring 84.75% (Very Good) across 27 end-user respondents using the EUCS method. The system provides a dependable, scalable, and efficient solution for institutional examination management.

## REFERENCE

- [1] A. Stanoyevitch, "Online assessment in the age of artificial intelligence," *Discover Education*, vol. 3, no. 126, pp. 1–15, 2024.
- [2] E. Cahyaningsih, Shaleh, and A. Fitriyanti, "Utilization of Computer-Based Testing Technology to Optimize Cognitive-Based Assessment Literacy in the Learning of Islamic Elementary School Students," *Journal of Integrated Elementary Education*, vol. 4, no. 2, pp. 151–172, 2024.
- [3] F. Gamaliel and P. Y. D. Arliyanto, "Perancangan Aplikasi Ujian Online Berbasis Website," *Journal Education and Technology (JUTECH)*, vol. 2, no. 2, pp. 112–120, 2021.
- [4] P. Rianda, N. Vebriani, Ambiyar, and D. Rindhani, "Perancangan Sistem Ujian Online dengan Data Base Mysql di SMK N 9 Padang Kelas X Semester Genap Tahun Ajaran 2021/2022," *Journal on Education*, vol. 5, no. 2, pp. 4512–4522, 2023.
- [5] S. Surajsingh Gaur and C. K., "EasyLearnify - A Student Study Portal," *INTI JOURNAL*, vol. 2024, no. 17, pp. 1–9, 2024.
- [6] D. Refandy and E. Wahyuningtyas, "Perancangan Sistem Computer Based Testing Berbasis Web pada Sekolah Dasar," *Melek IT: Information Technology Journal*, vol. 7, no. 1, pp. 67–76, 2021.
- [7] R. Siswanto and D. S. A. Morang, "Rancang Bangun Aplikasi Computer-based Test (CBT) serta Optimasinya Menggunakan Cache dan Queue Job," *Jurnal Komputer Terapan*, vol. 8, no. 2, pp. 160–170, 2022.
- [8] Annurwan, H. L. Purwanto, and W. Setyaningsih, "Rancang Bangun Sistem Informasi Ujian Online Berbasis Web di MTSS Nasruddin Dampit," *Journal of Information Technology, Information System and Communications (JISTIC)*, vol. 1, no. 2, pp. 45–54, 2023.
- [9] Ambiyar, Panyahuti, and A. T. Devega, "Web and Android-based Test Application Development and its Implementation on Final Semester Examination," *International Journal on Informatics Visualization (JOIV)*, vol. 8, no. 2, pp. 768–776, 2024.
- [10] A. P. M. Dela Rosa and R. P. P. Abad, "Leveraging Technology for Teaching and Learning: Developing a Django-based Quiz Application for Education," *Journal of Information Systems Engineering and Management*, vol. 10, no. 13s, pp. 1–11, 2025.
- [11] L. A. Alfitri and Z. Khaerunnisa, "Development of Web-Based Academic Information System Using Django and R&D for Student Achievement Records with Integrated Supervision Module," *Jurnal Guru Komputer*, vol. 6, no. 1, pp. 1–12, 2025.
- [12] M. N. Fachrezi and G. P. Mahardhika, "Perancangan Website CreativePub dengan Penerapan Metode Rapid Application Development (RAD)," *AUTOMATA*, vol. 3, no. 2, pp. 1–8, 2022.
- [13] B. Prasetya and I. Nuryasin, "Pengujian Kualitas Website PT Media Citra Digitalindo Blitar Menggunakan White box Testing Dengan Teknik Basis Path," *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, vol. 10, no. 2, pp. 1595–1601, 2025.