Jurnal Vokasi Informatika (JAVIT), Vol. 6, No. 2, pp. 1-26, July 2026
e-ISSN: 2775 - 6807

A Web-Based Library Management System with Notifications and AI-Powered Search at SMAN 1 Tambang

Ilvi Maulidya Nurulisa[1*], Asrul Huda[1], Dedy Irfan[1] and Vera Irma Delianti [1]

[1] Informatics Engineering Education Study Program, Faculty of Engineering, Universitas Negeri Padang, Padang, **Indonesia**

*Corresponding Author Email: ilvimaulidya21@gmail.com

DOI: https://doi.org/10.24036/qg572250

Abstract: Secondary school libraries in Indonesia largely remain dependent on manual or semi-digital record-keeping even as digital library adoption grows at the university level, creating gaps in data accuracy, user access, and accreditation readiness. This study addresses that gap by designing and developing a web-based library management system for SMAN 1 Tambang that integrates five functionalities which, based on the literature reviewed in this study, have not previously been reported within a single school library management system: AI-powered natural-language search using the Gemini API, automated WhatsApp overdue notifications via the Fonnte API, six-role access control, digital rights management (DRM) for digital collections, and QRIS-based fine payment through Midtrans. The system was developed using the Waterfall method, Laravel 12, MySQL, and Bootstrap 5. Black Box Testing across 30 scenarios achieved a 100% success rate. The AI search feature achieved an overall Precision of 69.23%, Recall of 75.00%, and F1-Score of 71.96%, with an average response time of 866 ms, while Load Testing recorded a 0.00% error rate under loads of 10–100 concurrent users. User Acceptance Testing involving 30 respondents yielded a mean score of 4.30 out of 5.00 ("Very Good"), supported by a valid and reliable instrument (Cronbach's Alpha = 0.9428). These findings demonstrate the technical feasibility of integrating AI-powered search, automated notifications, DRM, and digital payment within a unified school library management platform while improving both operational efficiency and service quality.

Keywords: Library Management System; Laravel; Artificial Intelligence; Gemini API; Black Box Testing; User Acceptance Testing; WhatsApp Notifications.

1. Introduction

Advances in information technology during Industry 4.0 have reshaped education, including how school libraries manage collections and services. Digitalization of library services has become essential for supporting teaching and learning while fostering a reading culture in schools.

At the national level, secondary school libraries in Indonesia largely remain on manual or semi-digital record-keeping despite growing digitalization of university libraries, causing delays, data errors, and record duplication [1]. Library accreditation standards further require organized, technology-based management of collections, services, and administration [2].

This challenge is evident at SMAN 1 Tambang, a public senior high school in Kampar Regency, Riau. Between 2023 and 2026, the library recorded 18,252 borrowing transactions, averaging approximately 89 transactions per month, a volume that exceeds the practical capacity of manual record management. During the same period, 1,905 overdue cases were recorded, representing an increase of more than 740% between 2023 and 2024 and indicating the absence of an effective automated reminder system. These statistics were obtained from the operational database of the SMAN 1 Tambang library for the 2023–2026 period. The anonymized operational data were provided by library staff with institutional approval from the school administration for research purposes. The dataset contained no personally identifiable information (PII) relating to individual library users.

Interviews with the school librarian confirmed that manual verification of borrowing records causes long queues, data errors, and delays in issuing Library Clearance Certificates, while the absence of an online catalog and automatic reminders is a primary cause of the overdue cases reported above [3].

Recent advances in artificial intelligence create new opportunities for school library management [4],particularly Natural Language Processing (NLP), which enables systems to interpret user queries expressed in natural language [5]. This study applies NLP through the Gemini API within the OPAC, allowing users to search physical and digital collections using conversational language rather than exact keywords.

Prior studies have shown that web-based library systems can improve borrowing efficiency but generally lack automated notification and AI capabilities [6]. Other studies have enhanced system accessibility, although search functionality remains dependent on exact keyword matching [1]. Multi-role access control has also been implemented in school library systems, but without integrating WhatsApp notifications, AI-driven search, DRM protection, or QRIS-based payment, while several studies have focused primarily on library digitization without incorporating third-party service integration [7],[8],[9]. Collectively, these studies indicate that a comprehensive integration of automated WhatsApp notifications, AI-based semantic search, DRM protection, QRIS payment, and six-role access control within a single school library platform has not yet been reported.

To substantiate the identified research gap, Table 6 presents a systematic feature-by-feature comparison with recent related studies. Accordingly, this study makes two principal contributions. First, it demonstrates through empirical evaluation the feasibility of integrating five previously unconsolidated functionalities, including AI-based semantic search, automated WhatsApp notifications, DRM protection, QRIS payment, and six-role access control, within a single operational school library platform. Second, it establishes a transparent and reproducible evaluation protocol for metadata-based AI search relevance in a low-resource, single-school environment, thereby providing a level of methodological detail beyond that explicitly reported in the related literature.

To address this research gap, this study proposes the development of a web-based library management system for SMAN 1 Tambang with five major innovations: (1) automated WhatsApp notifications using the Fonnte API integrated with Laravel Scheduler; (2) an AI-powered intelligent search feature within the OPAC using the Gemini API capable of processing natural language queries; (3) six user roles with role-based access control [10],[11]; (4) DRM implementation through access tokens and digital watermarking for protecting digital collections [12],[13],[14],[15]; and (5) QRIS-integrated library fine payment through the Midtrans payment gateway [16],[17]. This integration is expected to strengthen management efficiency and support library accreditation readiness at SMAN 1 Tambang.

2. Material and methods

2.1 System Development Method

This study employed the Waterfall software development method, which comprises five sequential phases: Requirements, Design, Implementation, Verification, and Maintenance [18],[19] The Waterfall method was selected because the system's scope was clearly defined from the outset, the requirements remained stable throughout development, and the method aligns well with the documentation demands of academic research. Requirements analysis was conducted through direct observation, interviews with SMAN 1 Tambang library staff, and review of the library's operational documents.

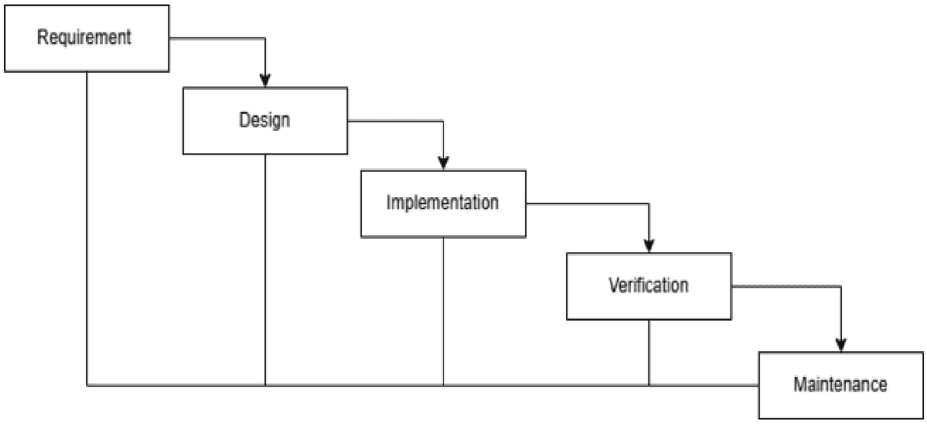

Figure 1. Waterfall Development Method

2.2 Research Location and Respondents

This research was conducted at the SMAN 1 Tambang library, located in Tambang Subdistrict, Kampar Regency, Riau Province. User Acceptance Testing (UAT) involved 30 respondents, comprising 1 library staff member, 21 library members (students, teachers, and staff), and 8 general non-member visitors. Respondents were selected using purposive sampling, whereby participants were deliberately chosen based on their direct involvement as users of the system under evaluation. The selection criteria included: (1) genuine access to and need for SMAN 1 Tambang's library services; (2) representation across all user roles within the system; and (3) willingness to participate in the testing session and complete the questionnaire in full. A sample of 30 respondents is considered adequate, in accordance with guidelines stating that 5 to 20 users are typically sufficient to identify the majority of usability issues in formative testing [20], consistent with standard practice for formative usability evaluation of small, well-defined systems such as a single-school library platform, where the goal is to identify major usability issues rather than to achieve statistical generalization across a large population.

This sample size was further justified by two considerations. First, the study required not only qualitative usability evaluation but also quantitative psychometric validation of the questionnaire through Pearson product–moment item validity and Cronbach's Alpha reliability analyses. Because correlation-based validation generally benefits from a larger sample size than formative usability testing alone, a sample exceeding the minimum recommended range was intentionally adopted. Second, empirical evidence indicates that increasing the number of participants from 5 to 10–20 substantially reduces the likelihood of overlooking usability problems, with the worst-case issue detection rate improving from 55% for five participants to 80% for ten participants [21]. Accordingly, recruiting 30 respondents provided additional methodological support beyond the minimum sample size typically recommended for formative usability studies.

The sample size was selected based on established usability evaluation practices and the statistical requirements of reliability analysis. Software acceptance was evaluated using Cronbach's Alpha, and the resulting reliability coefficient of 0.9428 (Section 3.4) substantially exceeded the minimum acceptable threshold of 0.70, indicating excellent internal consistency. Therefore, the adopted sample of 30 respondents was considered sufficient for software acceptance evaluation [20], [21], [37].

2.3 System Development Tools

The system was built using the Laravel 12 framework with a Model-View-Controller (MVC) architecture to manage backend logic. The MySQL database was administered through XAMPP, while the frontend interface was developed using Bootstrap 5, HTML, CSS, and JavaScript, within a Visual Studio Code development environment. The system integrates three third-party services: the Fonnte API for automated WhatsApp notifications, Google's Gemini API for AI-based intelligent search, and Midtrans as the payment gateway for QRIS-based fine payment.

2.4 Database Design

The database was designed in MySQL with 15 primary tables interconnected through Primary Key (PK) and Foreign Key (FK) relationships. These tables include *users, anggota (members), buku (books), peminjaman (loans), denda (fines), peminjaman_digital (digital loans), bookings, kunjungan (visits), ulasan_buku (book reviews), baca_di_tempat (in-library reading), notifikasi (notifications), poin_anggota (member points), activity_logs, peminjaman_log (loan logs)*, and *buku_penulis (book authors)*. The database design was based on an Entity Relationship Diagram (ERD) developed to ensure data integrity and consistency across all system modules. The core relations link users to anggota (1-to-1, extending base accounts with membership data), anggota to peminjaman (1-to-many, one member to many loan transactions), buku to peminjaman (1-to-many, one book title to many loan records), and peminjaman to denda (1-to-1, each overdue loan generating one fine record), with buku_penulis serving as a many-to-many junction table between buku and authors.

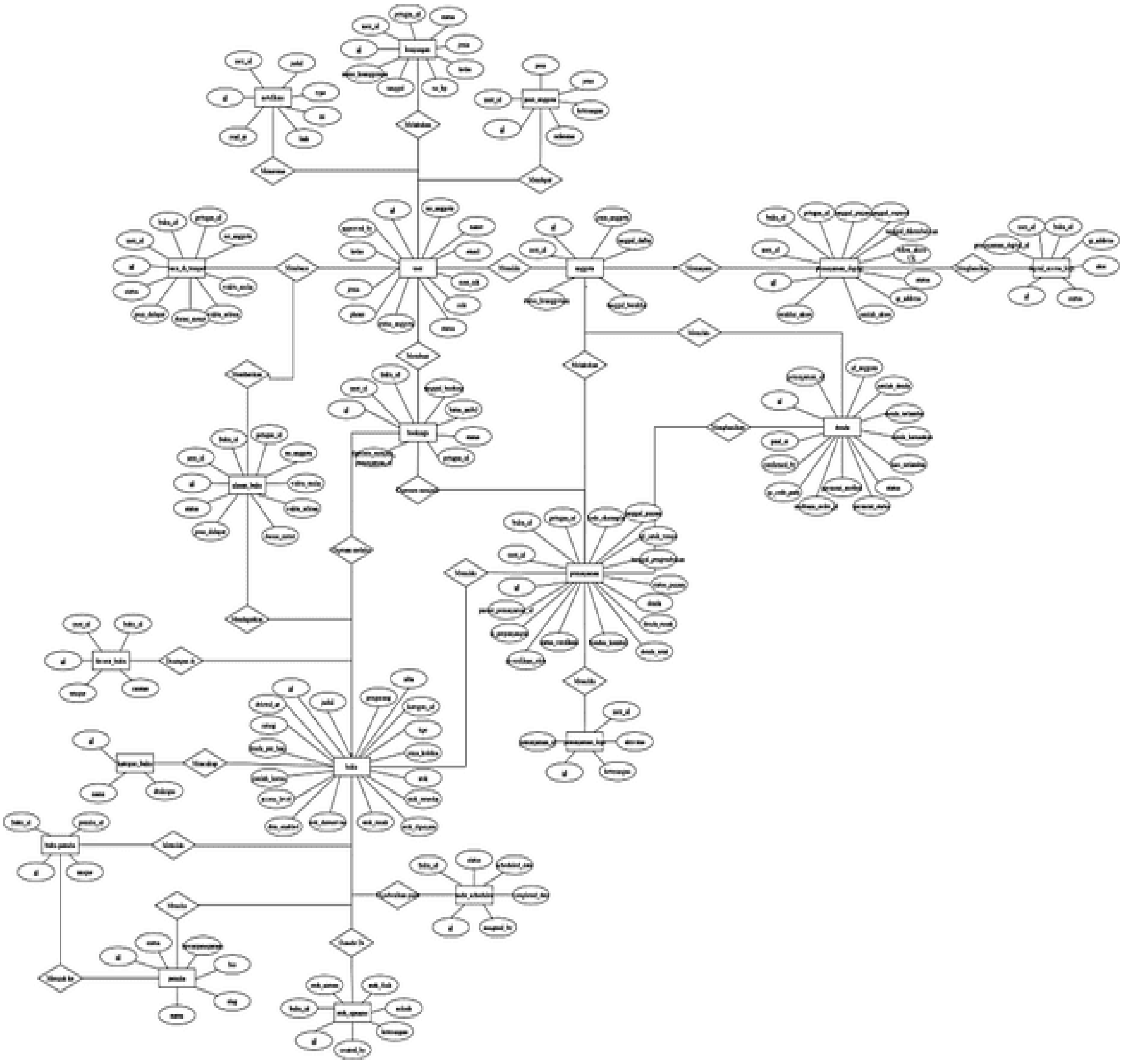

Figure 2. Entity Relationship Diagram

2.5 System Testing Method

System testing was carried out using three methods: (1) Black Box Testing for functional verification, comprising 30 test scenarios, 24 positive and 6 negative; (2) operational validation of the AI feature using five search scenarios covering direct keywords, conceptual synonyms, descriptive informal language, open-ended recommendation queries, and invalid queries; and (3) UAT using a five-item, 1–5 Likert-scale questionnaire. The UAT instrument was tested for validity using Pearson product-moment correlation and for reliability using Cronbach's Alpha. System performance testing was conducted using the Loadly tool across three test types: Load Testing, Stress Testing, and Response Time Testing for the system's primary endpoints. UAT data were analyzed using descriptive quantitative analysis by calculating the mean value of each assessment indicator on the five-point Likert scale, which was then classified into rating categories to determine the overall level of user acceptance.

To support reproducibility, the AI evaluation protocol is described as follows. The evaluation dataset consisted of the complete metadata table of the library catalog available in the system database at the time of testing, comprising 60 titles from both physical and digital collections. This evaluation dataset reflects the operational design of the AI search feature, which matches user queries against the entire library catalog rather than the digital collection alone.

Five representative queries were intentionally selected to cover the principal retrieval scenarios expected in practical school library use, namely direct keyword matching, conceptual synonyms, descriptive natural-language queries, open-ended recommendation requests, and invalid queries. Given the exploratory nature of this study and the relatively small size of the library collection (60 titles), the evaluation was designed to assess system behavior across representative query categories rather than to establish a statistically exhaustive benchmark of semantic retrieval performance. Expanding both the number and diversity of evaluation queries represents an important direction for future research.

For each of the five evaluation queries, every title retrieved by the system was assessed by the researcher as either relevant or irrelevant to the query topic. To reduce potential single-rater bias, Claude (Anthropic), a commercially available general-purpose large language model architecturally distinct from the Gemini model used in the search system, was consulted as a secondary reference. The researcher's judgment remained the final authority in establishing the ground-truth relevance set used to compute Precision, Recall, and F1-Score.

This LLM-assisted relevance assessment adopted in this study is consistent with emerging evidence from information retrieval research indicating that large language models can produce relevance judgments that closely align with human assessments [22]. Nevertheless, the use of a single researcher supported by an LLM represents a lighter-weight methodology than evaluation by an independent multi-rater expert panel. Accordingly, future work should validate the relevance labels through assessments conducted by professional librarians or multiple independent evaluators. Titles not retrieved by the system for a given query were treated as non-retrieved rather than being explicitly labeled as relevant or irrelevant, consistent with standard precision–recall evaluation based solely on retrieved items.

The AI search feature was implemented using the Gemini 2.0 Flash model, accessed through Google's Generative Language API (v1beta endpoint). The model was configured with a decoding temperature of 0.7 and a maximum output limit of 2,048 tokens, while all other parameters, including safety settings and system instructions, remained at their default API values. To improve service availability, the application implemented an automatic fallback mechanism that sequentially attempted Gemini 2.5 Flash, Gemini 2.5 Pro, and Gemini Flash Latest if the primary model request failed. However, all five evaluation scenarios reported in this study were successfully processed using the primary Gemini 2.0 Flash model, and the fallback mechanism was not invoked during testing.

The prompt template presented in Section 3.5 was developed through an iterative prompt-engineering process involving approximately ten candidate prompt templates. The final template was selected because it consistently minimized the retrieval of topically irrelevant titles while preserving semantically relevant recommendations. To facilitate reproducibility, the prompt template and the query–response evaluation dataset are available from the corresponding author upon reasonable request.

2.6 Research Conceptual Framework

The conceptual framework of this study illustrates the logical relationship between the problems identified in the field, the designed solutions, the technologies employed, and the expected outcomes. Four principal problem–solution pairings were identified: (1) manual record-keeping addressed through web-based digitalization using Laravel and MySQL; (2) limited OPAC search capability addressed through an AI-based search feature powered by the Gemini API; (3) the absence of overdue notifications resolved through integration of the Fonnte API with Laravel Scheduler; and (4) cash-based fine management replaced by Midtrans QRIS integration. System development followed the sequential stages of the Waterfall method, from requirements analysis through system maintenance.

3. Results and discussion

3.1 System Implementation Results

System development produced an integrated, web-based digital platform designed to address the fragmentation of library data that had previously characterized library operations. The system was built using the Laravel framework with a Model-View-Controller (MVC) architecture, a MySQL database, and a responsive Bootstrap 5 interface. The MVC architecture was chosen because it clearly separates business logic, data management, and interface presentation, thereby simplifying future maintenance and system development [23],[24],[25]. The system comprises six integrated functional modules, described in the following subsections.

3.1.1 Authentication and Account Management Module

The system implements role-based access control that differentiates each user's access privileges according to their responsibilities [10],[11]. Administrators are authorized to manage all user accounts, including adding, editing, and deleting accounts, configuring access rights, resetting passwords to default values (NISN/NIK), and performing bulk user data imports via Excel files. Member numbers are generated automatically by the system whenever an administrator adds a new user, eliminating the risk of duplicate membership numbers.

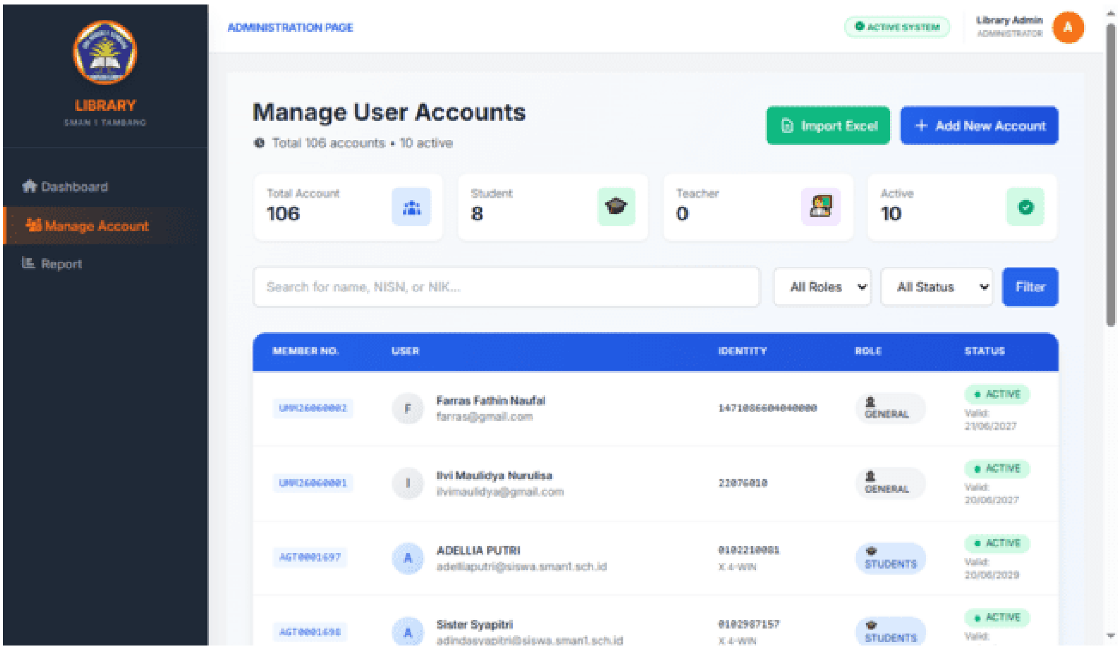

Figure 3. Account Management Page

3.1.2 Collection Management Module

The system integrates the management of both physical and digital collections within a single platform. For physical collections, staff can enter complete bibliographic details, update stock levels in real time, and manage collection categories. The book entry page displays summary statistics covering total books, available copies, total items on loan, and the number of categories. For digital collections, the system provides a PDF upload feature together with metadata management (author, genre, publication year, access status) governed by a lightweight Digital Rights Management (DRM) mechanism, ensuring that available digital content consists solely of collections with authorized distribution rights [14],[15]. Technically, DRM is enforced through two mechanisms: (1) time-limited, single-use access tokens generated per borrowing session, validated server-side on every file request and automatically invalidated upon session expiry or logout, and (2) visible digital watermarking that embeds the borrower's member ID and access timestamp

Jurnal Vokasi Informatika (JAVIT), Vol. 6, No. 2, pp. 1-26, July 2026
e-ISSN: 2775 - 6807

onto each PDF page using the FPDI/TCPDF library, deterring unauthorized redistribution of digital collections.

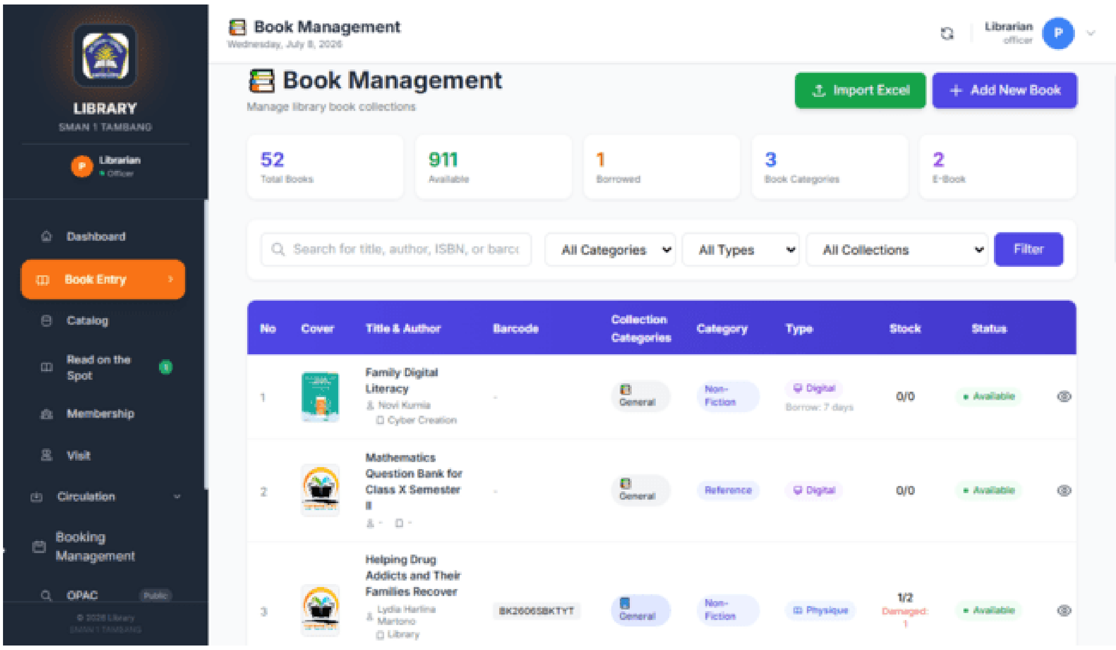

Figure 4. Collection Management Page

3.1.3 Circulation Module (Borrowing and Returns)

The circulation module manages all borrowing and return transactions while ensuring data consistency and accurate inventory management. As illustrated in **Figure 5**, before a borrowing transaction is finalized, the system performs a verification process that displays the selected member, book information, copy code, borrowing date, and due date. This verification step allows librarians to confirm transaction details before the loan is permanently recorded. Once confirmed, the system automatically stores the transaction, updates the borrowing history, and reduces the available stock of the selected book in real time.

Jurnal Vokasi Informatika (JAVIT), Vol. 6, No. 2, pp. 1-26, July 2026
e-ISSN: 2775 - 6807

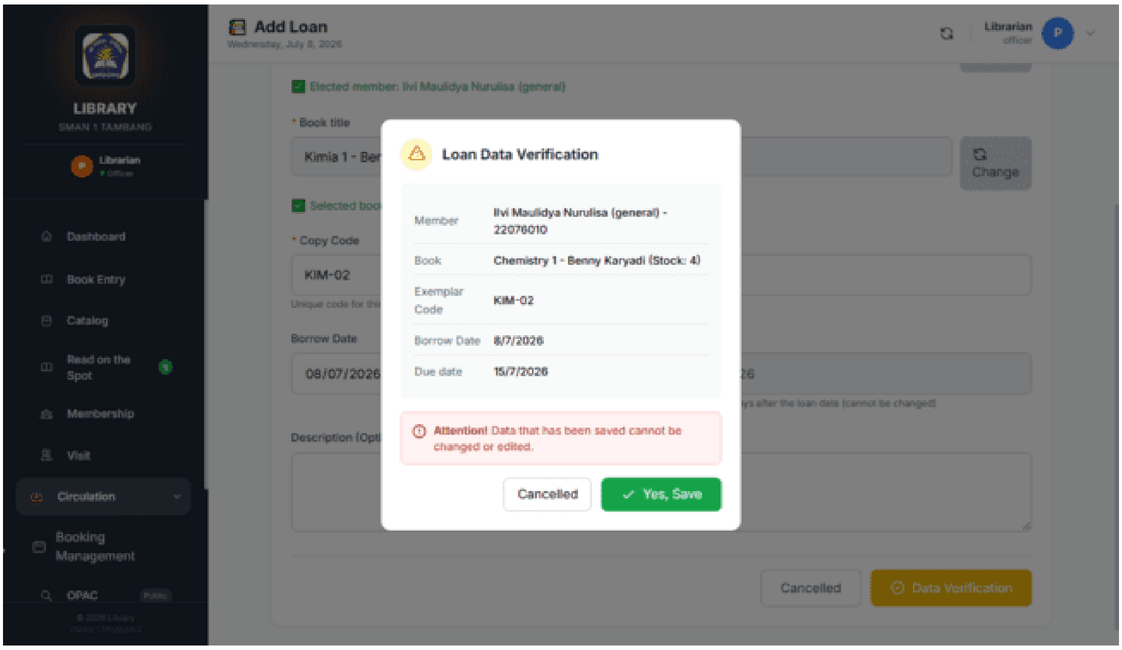

Figure 5. Circulation Borrowing Page

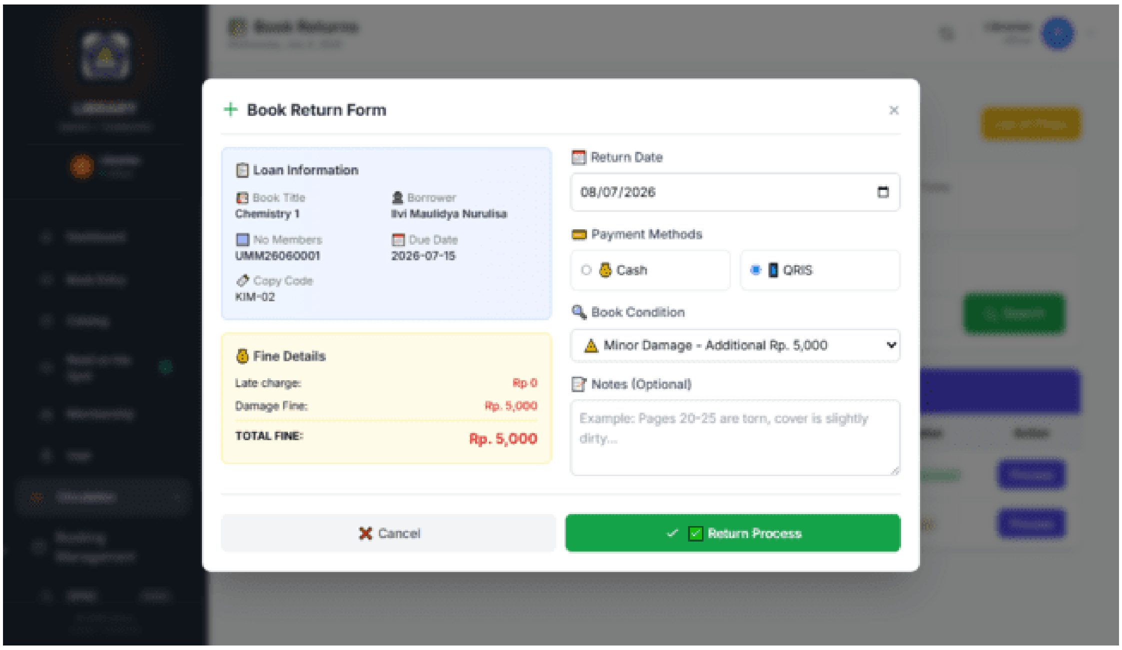

Figure 6. Circulation Return Page

During the return process, shown in **Figure 6**, the librarian records the return date, selects the payment method, and specifies the physical condition of the returned book. The system automatically compares the return date with the due date to determine whether the book is returned on time or overdue. If the returned book is damaged, the librarian can select the appropriate damage category, and the corresponding fine is calculated automatically according to the predefined library policies. The total payable fine is displayed immediately, ensuring transparency and minimizing manual calculation errors.

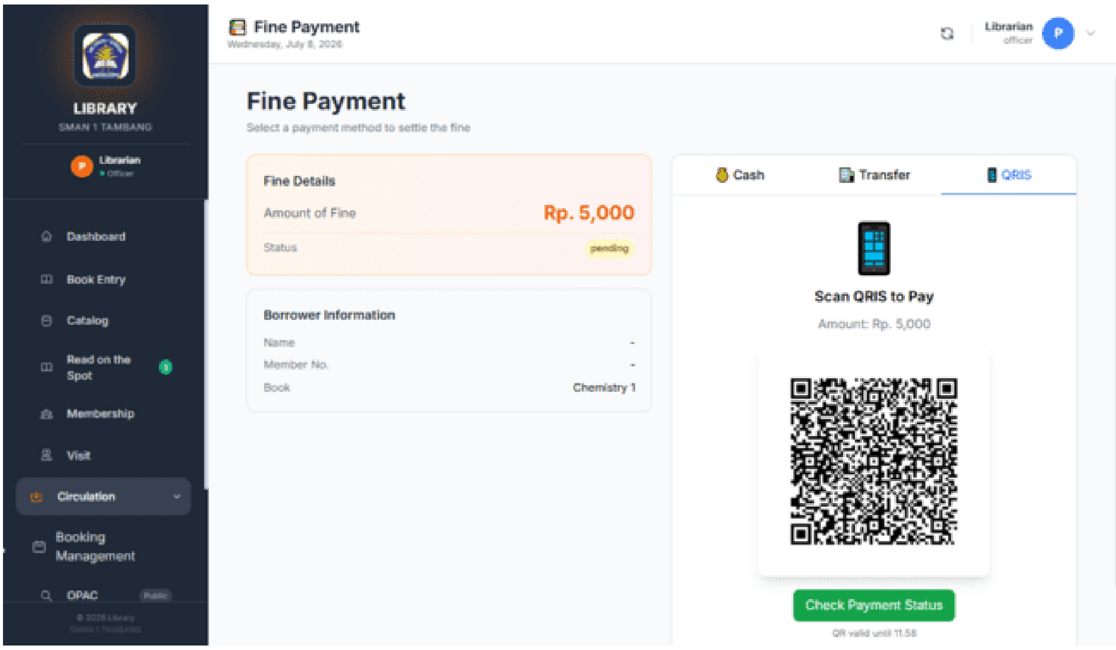

Figure 7. QRIS Page

If a fine is incurred, the system provides multiple payment options, including cash, bank transfer, and QRIS digital payment, as presented in **Figure 7**. For QRIS transactions, the system generates a QR code containing the payment information, allowing borrowers to complete the payment using any QRIS-compatible mobile banking or e-wallet application. After the payment is successfully verified, the fine status is automatically updated, and the return transaction is completed. Simultaneously, the book stock is increased and becomes available for future borrowing, while all transaction records are stored in the circulation history for reporting and audit purposes.

3.1.4 WhatsApp Notification Feature via Fonnte API

The automated WhatsApp notification mechanism was implemented using the Fonnte API integrated with Laravel Scheduler. The system dispatches two scheduled notification types: (1) a reminder sent one day before the due date to all members with active loans approaching their deadline, and (2) an overdue notification sent to members who have exceeded the due date without returning their books. Both notification types operate autonomously without manual staff intervention, substantially reducing the administrative burden and improving the consistency of overdue monitoring.

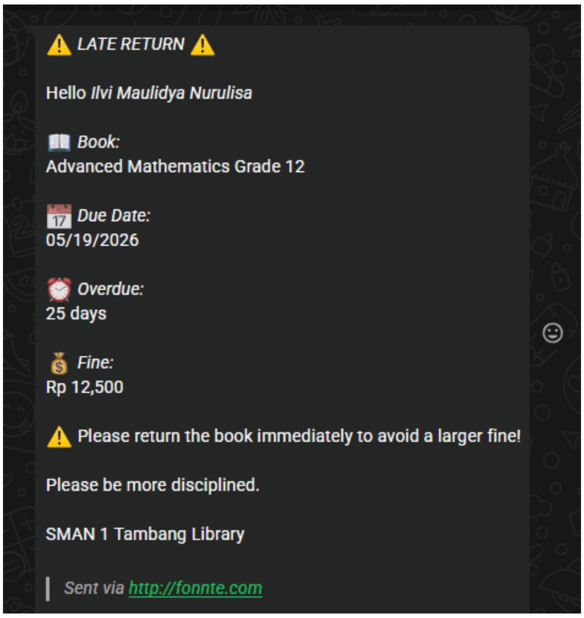

Figure 8. WhatsApp Overdue Notification Display

3.1.5 AI Search Feature with Gemini API

The intelligent search feature within the Digital Collection module was implemented using Google's Gemini API [26],[27],[28]. The system leverages Natural Language Processing (NLP) capabilities to interpret search queries expressed in everyday language and match them against the metadata of digital collections stored in the database. Users can search for e-books using free-form phrases, synonyms, or thematic context without needing to know the exact title or author name. The application of NLP in library search systems has been shown to improve search relevance and reduce access barriers for non-expert users [29],[30].

Technically, the feature aggregates metadata (title, author, category, synopsis) for the full digital collection from MySQL, embeds it as structured context in a formatted prompt, and sends it to the Gemini API together with the user's query. A representative prompt template is: "Based on this book metadata: [metadata], recommend books relevant to the following user query: {user_query}. Return only titles that genuinely match the query's topic or intent." The model's response is then parsed and matched back against the database to render the recommended titles in the OPAC.

To clarify the technical scope of the proposed system, the Gemini API operates exclusively on structured metadata, including the title, author, category, and synopsis of each library item. It does not process full-text digital documents, vector embeddings, or a Retrieval-Augmented Generation (RAG) pipeline. Unlike RAG architectures, which combine a dedicated retriever with a generative model over chunked document embeddings stored in a vector database [31],[32], the proposed approach incorporates the complete metadata table directly into the prompt context at query time.

This design was adopted because the SMAN 1 Tambang library maintains a relatively small collection, allowing the complete metadata to fit within a single prompt without exceeding the model's context window. As a result, the system can perform semantic matching beyond exact keyword searches while avoiding the additional infrastructure required by embedding-based retrieval, such as an embedding pipeline and vector database. This design choice reflects the intended scope of a single-school deployment rather than a technical limitation. As library collections continue to grow, embedding-based retrieval and RAG architectures represent promising directions for future research.

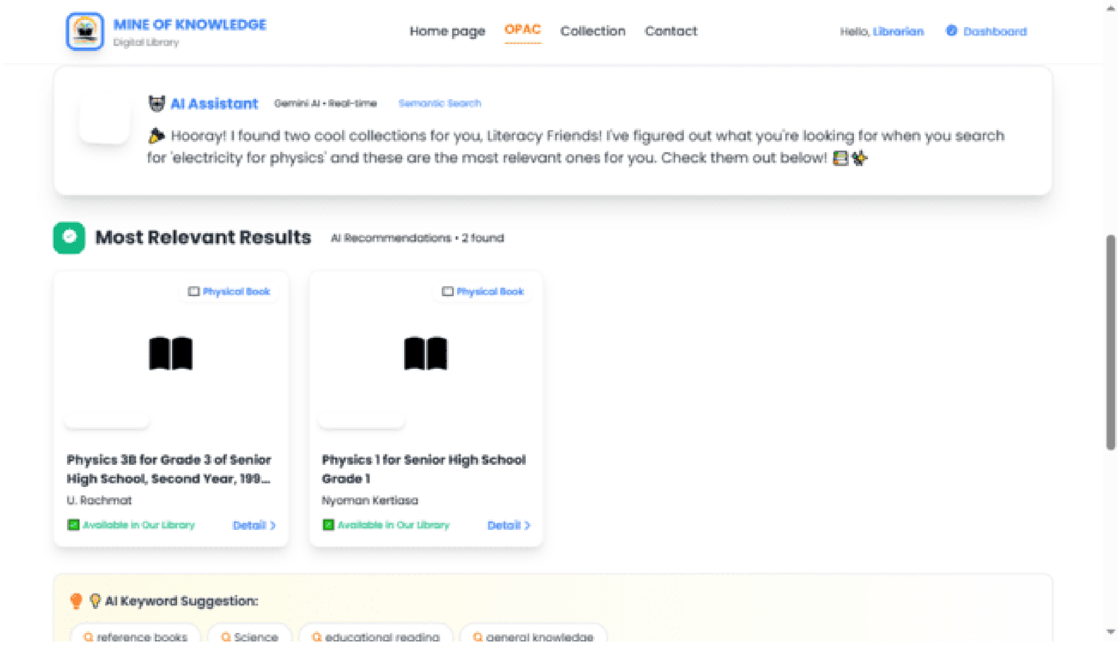

Figure 9. AI Feature Search Results on the OPAC

3.1.6 Visit Tracking, Reporting, and Multi-Role Dashboard Module

The system provides an automated visit-logging feature that generates daily, weekly, and monthly statistics for accreditation reporting purposes. The reporting module enables staff to export various operational reports including member, collection, fine, visit, and loan reports in Excel or PDF format. Each user role has a customized dashboard: the Staff Dashboard displays statistics on total collections, active members, daily visits, and accumulated fines; the Member Dashboard summarizes active loans and reward point information; while the School Leadership Dashboard presents executive reports and library Key Performance Indicators (KPIs) in visual format.

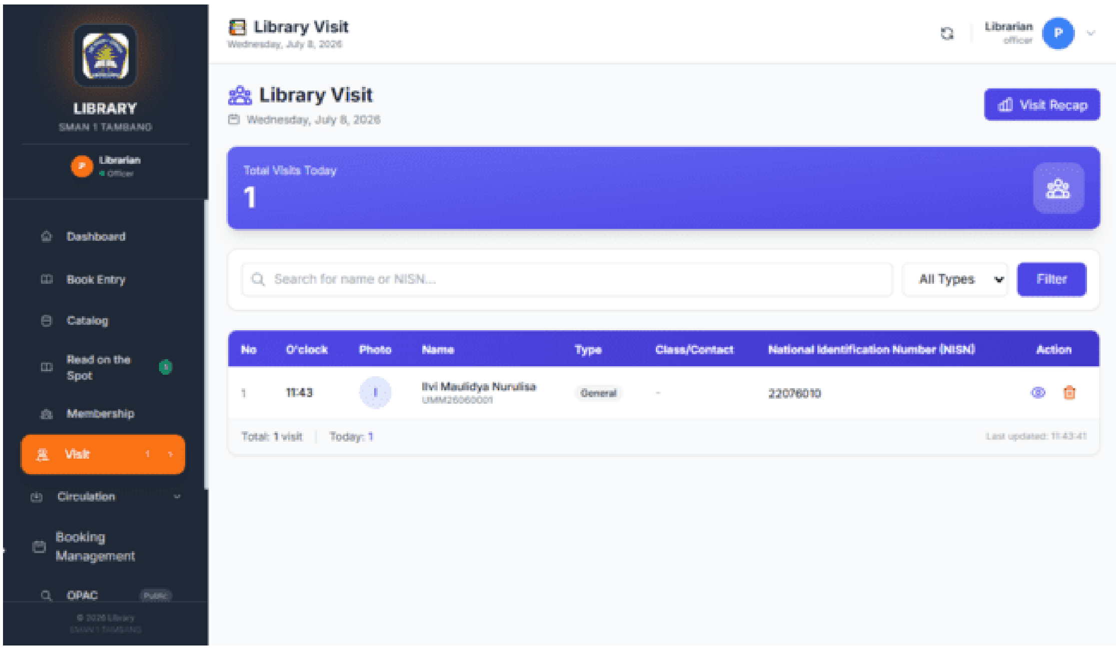

Figure 10. Visit Tracking Page

3.2 Functional Testing with Black Box Testing

Functional testing was conducted using the Black Box Testing method with a dynamic testing approach, in which the system is executed directly and the correspondence between given inputs and produced outputs is observed [33],[34],[35]. Testing covered all user roles across 30 test scenarios, comprising 24 positive and 6 negative scenarios. Table 1 summarizes representative results covering critical scenarios from each major system module.

Table 1. Summary of Black Box Testing Results

ID	Module	Test Scenario	Expected Result	Status
L01	Login	Input registered email/NISN and correct password	System validates and redirects to the role-appropriate Dashboard	Passed
A02	Account Management	Administrator imports user data via Excel file	Data successfully imported into the database in bulk	Passed
P03	Book Borrowing	Staff processes a book loan transaction for a member	Transaction recorded, stock reduced in real time, due date stored	Passed
P04	Book Return	Staff processes a book return past the due date	System automatically calculates fine and updates stock	Passed
AN02	AI Feature	Member searches the collection using everyday language in OPAC	Gemini API processes the query and displays relevant collection recommendations	Passed
N01	WhatsApp Notification	System runs the one-day pre-due-date reminder scheduler	Notification successfully delivered to member's WhatsApp via Fonnte API	Passed

ID	Module	Test Scenario	Expected Result	Status
N-01	Book Borrowing (Negative)	Staff attempts to process a loan with stock = 0	System rejects the transaction and displays a stock-unavailable message	Passed
N-04	WhatsApp Notification (Negative)	Scheduler runs against an inactive WhatsApp number	System logs the failure in the notification log without crashing	Passed

As shown in Table 1, all 30 test scenarios returned a “Passed” result, corresponding to a 100% success rate (30/30 scenarios). These results confirm that every system module functions according to its design specifications and handles edge cases and off-path usage conditions effectively. These findings align with previous research reporting that web-based library systems significantly improve the accuracy and efficiency of collection management compared to manual systems [6].

3.3 Operational Validation of the AI Feature

AI feature testing was conducted to verify that the system could process natural-language queries and return semantically relevant collection recommendations. The AI feature on the OPAC page operates by aggregating metadata for the entire collection (title, author, category, and synopsis) from the MySQL database, structuring this information as context within a formatted prompt, and submitting it to the Gemini API. The model then interprets the semantic context and intent of the query, implicitly recognizing synonyms and word variations without requiring exact keyword matching.

Table 2. AI Feature Test Results

No	Query Category	Input Query	Recommended Collection	Relevance	Status
1	Direct keyword	“books about organic chemistry”	Kimia 1 (2 relevant titles)	Relevant	Passed
2	Conceptual synonym	“electricity material for physics”	Fisika 3B, Fisika 1 for senior high (100% relevant)	Relevant	Passed
3	Descriptive informal language	“children's books”	Anak-anak Duano, Early Childhood Education (100% relevant)	Relevant	Passed
4	Open-ended recommendation query	“recommend math learning materials”	Matematika Kelas X, Matematika 3A, Math Question Bank for Grade X	Relevant	Passed
5	Invalid query (edge case)	“xyzabc unclear”	Structured informational message – no collection displayed	No collection	Passed

Based on Table 2, the AI feature successfully returned semantically relevant collection recommendations in 4 out of 5 test scenarios (80%). The scenario involving an invalid query was handled with a structured, informative message rather than a misleading output, demonstrating the system's capacity to recognize the boundaries of its own capability.

Table 3. Precision, Recall, F1-Score, and Response Time Results for the AI Feature

Query Category	Precision (%)	Recall (%)	F1-Score (%)	Avg. Response (ms)
Direct keyword	66.67	66.67	66.67	917
Conceptual synonym	100.00	66.67	80.00	829
Descriptive informal language	100.00	100.00	100.00	903
Open-ended recommendation query	50.00	75.00	60.00	876
Invalid/edge-case query	-	-	-	803
Overall average	**69.23**	**75.00**	**71.96**	**866**

As shown in Table 3, the AI feature returned 13 collection items across the five test scenarios, of which 9 were judged relevant (True Positive) and 4 irrelevant (False Positive), yielding an average precision of 69.23%, recall of 75.00%, and accuracy of 71.96%, with an average response time of 866 ms well under the one-second threshold generally considered responsive for interactive search. The best performance was achieved in the descriptive-informal-language scenario, with precision, recall, and accuracy each reaching 100%. The lower accuracy observed in the open-ended recommendation query scenario stemmed from the limited number of mathematics-related items in the library database at the time of testing, rather than from a failure of the AI system itself.

Verification of the underlying calculations indicates that the metric originally labeled "Accuracy" in Table 3 is mathematically the harmonic mean of Precision and Recall for each query category. For example, in the conceptual-synonym category, $2 \times 100.00 \times 66.67/(100.00+66.67)=80.00$, which exactly matches the reported value. Accordingly, this metric is more appropriately reported as the F1-Score rather than Accuracy, and Table 3 has been revised accordingly.

In the conventional classification setting, Accuracy requires a well-defined count of true negatives (correctly non-retrieved irrelevant items). However, this requirement is not applicable to an open-set information retrieval task, where the pool of non-retrieved irrelevant items effectively comprises the remainder of the catalog rather than a fixed and enumerable set. Consequently, Accuracy is not reported as a primary evaluation metric, and the F1-Score is adopted instead, consistent with established practice in information retrieval evaluation. An aggregate confusion matrix summarizing the retrieved results across all five evaluation scenarios is presented in Table 4.

Table 4. Aggregate Confusion Matrix for the AI Search Performance Evaluation

Actual	Retrieved	Not Retrieved
Relevant	TP = 9	FN = 3
Irrelevant	FP = 4	TN = Not defined

Note: True negatives are not defined because the evaluation follows an open-set information retrieval setting rather than a closed-set classification task.

The evaluation metrics reported in Table 3 were computed from the aggregate confusion matrix shown in Table 4. Let TP denote a relevant item correctly retrieved, FP an irrelevant item incorrectly retrieved, and FN a relevant item that was not retrieved. Precision, Recall, and F1-Score were calculated using their standard definitions. Using the aggregate counts (TP = 9, FP = 4, FN = 3), Precision is calculated as 69.23%, Recall as 75.00%, and F1-Score as 71.99%. The slight difference from the reported value of 71.96% in Table 3 is attributable to rounding at the individual query-category level before aggregation.

Compared with previous studies on AI-assisted library search, which primarily reported qualitative relevance assessments without formal precision–recall analysis [1],[6], the present study adopts a quantitative evaluation framework based on Precision, Recall, and F1-Score. The resulting performance establishes a baseline for metadata-based semantic retrieval in a school-library environment. Although the system achieved encouraging results, the lower performance observed for direct-keyword and open-ended queries indicates opportunities for further improvement through embedding-based retrieval and Retrieval-Augmented Generation (RAG) architectures as library collections and query complexity continue to grow.

3.4 User Acceptance Testing (UAT)

The validity of the UAT instrument was tested using Pearson product-moment correlation with n = 30 and a significance level of α = 0.05 (r-table = 0.361). All five questionnaire items were found valid, with calculated r-values ranging from 0.8477 to 0.9620, well above the r-table threshold of 0.361. Instrument reliability was assessed using Cronbach's Alpha, yielding a value of 0.9428 substantially above the accepted minimum threshold of α > 0.70 confirming that the UAT instrument is reliable, internally consistent, and appropriate for use [36],[37],[38].

Table 5. Summary of UAT Results Across All User Groups

User Group	No. of Respondents	Mean Score	Category	Remarks
Library Staff	1	4.50	Very Good	Suitable for use
Library Members	21	4.39	Very Good	Suitable for use
General Visitors (Non-Members)	8	4.00	Good	Suitable for use
Overall Average	**30**	**4.30**	**Very Good**	**Suitable for use**

As shown in Table 5, the overall average UAT score across all user groups reached 4.30 out of 5.00, corresponding to the “Very Good” category. Library staff provided the highest rating (mean = 4.50), followed by library members (4.39) and general visitors (4.00). All user groups rated the system at least “Good” (≥ 3.41), confirming its overall suitability for use. The observed differences between groups can be logically attributed to variation in feature-access depth: staff and members interact with the full range of core system features, whereas general visitors engage only with the public-facing pages, which offer more limited functionality.

The lowest score was recorded for system speed among general visitors, at 3.38 (“Fair”), largely attributable to limitations in respondents' internet connectivity during testing rather than to server performance itself. The ease of member registration item received a score of 3.62 (“Good”), reflecting the fact that registration verification involves manual confirmation by staff and is therefore not instantaneous a deliberate library policy rather than a technical limitation of the system. These UAT results indicate that the web-based library system substantially improves both collection management efficiency and service quality.

3.5 System Performance Testing

System performance testing was conducted using the Loadly tool across three test types: Load Testing, Stress Testing, and Response Time Testing per endpoint [39],[40].Testing was performed in a local development environment using Laravel's built-in server (php artisan serve), simulating concurrent HTTP requests against the OPAC page.

Table 6. Load Testing Results for the Library Management System

Concurrent Users	Throughput (req/s)	Avg. Response Time (ms)	Error Rate (%)	P95 Response Time (ms)	Total Requests
10	2.3	4,154	0.00	4,869	78
25	2.3	9,323	0.00	11,439	96
50	2.3	17,382	0.00	22,537	117
75	2.2	24,698	0.00	33,838	142
100	1.7	39,242	0.00	70,922	169

As shown in Table 6, the system exhibited graceful degradation characteristics, with no failed requests (0.00% error rate) across all tested loads, ranging from 10 to 100 concurrent users. Throughput remained relatively stable at approximately 2.3 req/s up to the 50-concurrent-user scenario, suggesting that the server's processing capacity approached its saturation point within this range. At peak load (100 concurrent users), throughput declined to 1.7 req/s, with an average response time of 39,242 ms. This elevated response time reflects the constraints of the local development environment rather than the system's likely performance in an actual production deployment. The consistently zero error rate indicates that the underlying system architecture and application logic remain stable under load. Given that concurrent active usage at SMAN 1 Tambang is unlikely to exceed 30–50 users, the system has demonstrated its capacity to process all requests without failure under realistic usage conditions.

To support reproducibility, the hardware environment used for performance evaluation is described as follows: an Intel Core i3-7020U processor running at 2.30 GHz, 20 GB RAM, and a hybrid storage configuration consisting of a 932 GB HDD and a 119 GB SSD. The Laravel application was executed locally using XAMPP without a dedicated production-grade web server, database server, load balancer, or caching layer. Consequently, the response time observed under a load of 100 concurrent users primarily reflects the limitations of the local development environment rather than the application architecture itself.

In a typical production deployment, Laravel applications are commonly hosted on multi-core server hardware with solid-state storage and application-level performance optimizations, such as OPcache, Redis caching, and database connection pooling, all of which generally contribute to improved response times under comparable workloads. For deployments involving multiple schools or substantially larger user populations, horizontal scaling through multiple application server instances behind a load balancer and the use of database read replicas would provide a suitable scalability strategy. Within the context of a single-school deployment such as SMAN 1 Tambang, however, the proposed architecture is expected to adequately support typical concurrent access once deployed on production-grade infrastructure.

3.6 Discussion of Novelty and Research Contribution

The system developed in this study introduces a combination of features not found jointly in prior research. Prior work focused on basic digitalization without notification or AI capabilities. In contrast, the present system incorporates Fonnte API integration with Laravel Scheduler for automated overdue notifications, validated as effective through Black Box Testing. Another prior OPAC system supported only exact keyword matching [6], whereas the AI feature built on the Gemini API in this study processes semantic, natural-language queries with 69.23% precision and 75% recall [1]. A further prior system offered multi-role access without digital payment [7]; the present system, by comparison, integrates Midtrans QRIS, enabling cashless fine payment.

The principal contribution of this study is demonstrating that the simultaneous integration of five distinct novelties automated WhatsApp notifications, AI-based intelligent search, six user roles, DRM for digital collections, and QRIS payment within a single web-based school library platform is technically feasible and well received by users (UAT mean = 4.30/5.00, "Very Good" category). This finding supports the argument that AI plays an important role in advancing modern library management systems [41],[42],[43].The 100% Black Box Testing success rate further confirms that developing school library systems using the Waterfall method, supported by thorough requirements analysis, remains a relevant and effective approach [44].

Table 7. Comparison Between This Study and Prior Web-Based School Library Systems

Feature	Prior Studies	This Study	Remarks
Web-based library system	[6], [1], [7]	Yes	Established approach
AI-based semantic search	No	Yes (Gemini API)	Novel
Automated WhatsApp notification	No	Yes (Fonnte API)	Novel
QRIS-based fine payment	No	Yes (Midtrans)	Novel
Digital Rights Management (DRM)	No	Yes	Novel

As summarized in Table 7, prior systems have digitized core library operations but have not combined AI-based semantic search, automated notifications, DRM, and cashless payment within a single platform, which is the principal novelty demonstrated in this study.

Beyond feature-level comparison, recent Scopus-indexed research on AI-enabled library search suggests that Retrieval-Augmented Generation (RAG) architectures, which combine a dedicated retriever with vector embeddings derived from full document content, represent a more scalable long-term solution for academic and school library search as collections grow beyond the context capacity of a single prompt [31],[32]. The metadata-in-prompt approach adopted in this study achieved a comparable precision–recall profile (69.23% precision, 75.00% recall, and 71.96% F1-Score) while requiring substantially less infrastructure than an embedding-based RAG pipeline. These findings indicate that, for single-school libraries managing collections of only a few hundred titles, a lightweight metadata-based approach provides an effective balance between implementation feasibility and retrieval performance. As collection size and query complexity increase, however, adopting a RAG-based architecture becomes progressively more justifiable.

Beyond its technical contribution, the proposed system has broader practical implications for secondary-school library management in Indonesia. By automating overdue monitoring and fine management, the platform directly supports the technology-based collection and administrative practices required under national school-library accreditation standards [2], suggesting that similar systems could be adopted by other schools seeking to strengthen their accreditation readiness. The integration of QRIS further aligns the system with Indonesia's ongoing transition toward cashless public-service transactions [16],[17]. In addition, the use of WhatsApp notifications leverages a communication platform that is already widely adopted by students, teachers, and parents, thereby reducing barriers to user adoption compared with solutions that require a dedicated mobile application.

4. Conclusion

This study designed and developed a web-based library management system for SMAN 1 Tambang that integrates AI-based semantic search (Gemini API), automated WhatsApp overdue notifications (Fonnte API), six-role access control, DRM-protected digital collections, and QRIS-based fine payment (Midtrans). Black Box Testing across 30 scenarios achieved a 100% success rate; the AI search feature reached 80% relevance with an average response time of 866 ms; WhatsApp notifications were delivered with 100% success; load testing showed a 0.00% error rate for 10-100 concurrent users; and UAT across 30 respondents yielded a mean score of 4.30/5.00 ("Very Good"), supported by a valid and reliable instrument (Cronbach's Alpha = 0.9428).

These findings demonstrate that integrating AI, automated notifications, DRM, and digital payment into a single school library platform is technically feasible and substantially improves management efficiency and service quality, supporting the library's accreditation readiness. This study was tested at a single school and in a local development environment, which may limit generalizability and production-scale performance conclusions. Future work should replicate this system across multiple schools, add an Android/iOS application and barcode scanner integration, and evaluate more advanced AI models for full e-book content indexing.

Author contribution

Ilvi Maulidya Nurulisa: conceptualization, methodology, software development, data collection, formal analysis, and writing original draft. Asrul Huda: supervision, validation, and writing to review. Dedy Irfan: supervision and validation. Vera Irma Delianti: validation and review. All authors have read and approved the final manuscript.

Funding statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgements

The authors would like to thank the librarian of SMAN 1 Tambang, Dewi Etika, S.E., and the school administration for their assistance and support during the data collection and system testing process.

Competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] F. L. Asari, S. Meimaharini, T. Khotimah, and J. Artikel, "Implementasi Sistem Perpustakaan Berbasis Web untuk Meningkatkan Efisiensi Layanan Peminjaman dan Pengguna," *bit-Tech*, vol. 7, no. 3, pp. 770–778, Apr. 2025, https://doi.org/10.32877/BT.V7I3.2185.

[2] I. Krismayani, "Analisis Kesesuaian Instrumen Akreditasi Perpustakaan Sekolah Terhadap Ketentuan Standar Nasional Perpustakaan Sekolah Menengah Atas," *ANUVA*, vol. 3, no. 2, pp. 199–205, 2019.

[3] L. Ariani, Hamdani, and Alahuddin, "Meningkatkan Efisiensi Pengadaan Buku di Perpustakaan dengan Sistem Informasi Berbasis Komputer," *J. Manaj. Sist. Inf.*, vol. 3, no. 2, pp. 60–71, Oct. 2024, https://doi.org/10.59431/JMASIF.V3I2.466.

[4] A. T. Paiste and S. B. Siago, "Academic Librarians as Catalysts of Innovation : AI Competencies for Transforming Education," *Int. Res. Innov. J. Educ.*, vol. 1, no. 2, pp. 44–72, 2025.

[5] A. A. S. Ali and V. K. Shandilya, "AI-Natural Language Processing (NLP)," *Int. J. Res. Appl. Sci. Eng. Technol.*, no. August, 2021, https://doi.org/https://doi.org/10.22214/ijraset.2021.37293.

[6] S. Mukodimah, F. A. Wilson, and D. Susianto, "Aplikasi Manajemen Perpustakaan Berbasis Website Di SMAN 1 Pringsewu," vol. 6, no. 1, pp. 1–9, 2025.

[7] D. R. P. Perdana, S. Handayani, and M. Kholil, "Pengembangan Aplikasi Perpustakaan Berbasis Website untuk Meningkatkan Layanan Informasi di SMK," *J. Pendidik. Teknol. Inf.*, vol. 10, no. 3, pp. 19–28, 2025.

[8] A. N. Putra and G. Z. Muflih, "Perancangan Sistem Informasi Perpustakaan SMA Negeri 1 Gombong Berbasis Web Menggunakan Hypertext Preprocessor (PHP) dan MySQL," *J. Kridatama Sains Dan*

Teknol., vol. 6, no. 02, pp. 522–535, 2024, https://doi.org/10.53863/kst.v6i02.1245.
[9] R. P. Oktarina *et al.*, "Implementasi Manajemen dan Digitalisasi Perpustakaan di SMA Negeri 13 Surabaya," *Pubmedia J. Penelit. Tindakan Kelas Indones.*, vol. 2, no. 4, p. 10, 2025, https://doi.org/10.47134/ptk.v2i4.1678.
[10] M. Belchior, D. Schwabe, and F. Silva Parreiras, "Role-based Access Control for Model-driven Web Applications," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7387 LNCS, pp. 106–120, 2012, https://doi.org/10.1007/978-3-642-31753-8_8/SAVE-RESEARCH.
[11] V. Alturi and D. Ferraiolo, "Role-Based Access Control," *Encycl. Cryptogr. Secur. Privacy, Third Ed.*, pp. 2124–2127, Jan. 2025, https://doi.org/10.1007/978-3-030-71522-9_829/SAVE-RESEARCH.
[12] Ravichandra H S, "Securing the Digital Landscape : Present Concerns and Hurdles in Digital Rights Management," *Int. J. Sci. Res.*, vol. 13, no. 1, pp. 2022–2025, 2024.
[13] S. Acharya, A. K. Mishra, and O. Acharya, "Digital Rights Management : Addressing Copyright Protection in the Digital Era," *Int. J. Multidiscip. Res.*, vol. 7, no. 4, pp. 1–14, 2025.
[14] N. Agarwal, A. K. Singh, and P. K. Singh, "Survey of robust and imperceptible watermarking," *Multimed. Tools Appl. 2019 787*, vol. 78, no. 7, pp. 8603–8633, Jan. 2019, https://doi.org/10.1007/S11042-018-7128-5.
[15] R. C. O. Scholastica C. Ukwoma and N. E. Ugwu2, "Digital Rights Management in Electronic Publishing Environments: Implication for Academic Libraries in Nigeria | Mousaion." Accessed: Jul. 03, 2026. [Online]. Available: https://journals.co.za/doi/abs/10.25159/2663-659X/20935.
[16] E. Hamzah Muchtar *et al.*, "Quick response code Indonesia standard (QRIS) E-payment adoption: customers perspective," *Cogent Bus. Manag.*, vol. 11, no. 1, pp. 1–18, 2024, https://doi.org/10.1080/23311975.2024.2316044.
[17] N. E. Rafferty and A. N. Fajar, "Integrated QR Payment System (QRIS): Cashless Payment Solution in Developing Country from Merchant Perspective," *Asia Pacific J. Inf. Syst.*, vol. 32, no. 3, pp. 630–655, 2022, https://doi.org/10.14329/APJIS.2022.32.3.630.
[18] R.S. Pressman, *Rekayasa perangkat lunak (pendekatan praktisi)*. Yogyakarta: Penerbit ANDI, 2012.
[19] D. B. J. Rao, P. Kalpana, G. S. N. Kumar, and N. M. Atcha, "A Comparative Analysis of Software Development Models: Waterfall, Agile and DevOps," *Smart Innov. Syst. Technol.*, vol. 417, pp. 589–599, 2025, https://doi.org/10.1007/978-981-97-8355-7_51/SAVE-RESEARCH.
[20] J. Nielsen, *Usability engineering*. Boston, MA: Academic Press, 1993.
[21] L. Faulkner, "Beyond the five-user assumption: Benefits of increased sample sizes in usability testing," *Behav. Res. Methods, Instruments,*

Comput., vol. 35, no. 3, pp. 379–383, 2003, https://doi.org/https://doi.org/10.3758/BF03195514.

[22] P. Thomas, S. Spielman, N. Craswell, and B. Mitra, "Large Language Models can Accurately Predict Searcher Preferences," *SIGIR 2024 - Proc. 47th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 1930–1940, 2024, https://doi.org/10.1145/3626772.3657707.

[23] Z. Subecz, "Web-development with Laravel framework," *Gradus*, vol. 8, no. 1, pp. 211–218, 2021, https://doi.org/10.47833/2021.1.csc.006.

[24] A. Sunardi and Suharjito, "MVC Architecture: A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based," *Procedia Comput. Sci.*, vol. 157, pp. 134–141, Jan. 2019, https://doi.org/10.1016/J.PROCS.2019.08.150.

[25] M. H. Rahman, M. Naderuzzaman, M. A. Kashem, B. M. Salahuddin, and M. Z. Mahmud, "Comparative Study: Performance of MVC Frameworks on RDBMS," *Int. J. Inf. Technol. Comput. Sci.*, vol. 16, no. 1, pp. 26–34, 2024, https://doi.org/10.5815/ijitcs.2024.01.03.

[26] A. Wheatley and S. Hervieux, "Artificial intelligence in academic libraries: An environmental scan," *Inf. Serv. Use*, vol. 39, no. 4, pp. 347–356, 2020, https://doi.org/10.3233/ISU-190065.

[27] N. Nawaz, "Artificial Intelligence Chatbots for Library Reference Services." Jul. 10, 2021. Accessed: Jul. 03, 2026. [Online]. Available: https://papers.ssrn.com/abstract=3883917.

[28] N. Khan, D. Elizondo, L. Deka, and M. A. Molina-Cabello, "Natural Language Processing Tools and Workflows for Improving Research Processes," *Appl. Sci.*, vol. 14, no. 24, pp. 1–26, 2024, https://doi.org/10.3390/app142411731.

[29] H. Nabankema, "Evaluation of Natural Language Processing Techniques for Information Retrieval," *Eur. J. Inf. J. Manag.*, 2024.

[30] L. Joshi, "Multidisciplinary Trends (IJARMT) The Role of Artificial Intelligence in Modern Library Management Systems International Journal of Advanced Research and Multidisciplinary Trends (IJARMT)," no. 2, pp. 108–121, 2025.

[31] R. V. K. Bevara *et al.*, "Prospects of Retrieval-Augmented Generation (RAG) for Academic Library Search and Retrieval," *Inf. Technol. Libr.*, vol. 44, no. 2, 2025, https://doi.org/10.5860/ital.v44i2.17361.

[32] P. Mukhopadhyay, "Designing Conversational Search for Libraries: Retrieval Augmented Generation through Open Source Large Language Models," *DESIDOC J. Libr. Inf. Technol.*, vol. 45, no. 2, pp. 109–115, 2025, https://doi.org/10.14429/djlit.45.2.20206.

[33] A. Sultansyah, A. S. Rahayu, I. Yudiana, P. Fauzi, and E. N. Aripin, "Pengujian Black Box Testing Pada Fitur Permohonan Informasi Publik Melalui Website Pemerintah Jawa Barat," *J. Pengabdi. Masy. dan Ris. Pendidik.*, vol. 3, no. 4, pp. 5912–5919, 2025.

[34] D. I. Pirdaus and R. A. Hidayana, "Analysis Testing Black Box and White Box on Application To-Do List Based Web," *Int. J. Math. Stat. Comput.*, vol. 2, no. 2, pp. 68–75, 2024,

https://doi.org/10.46336/ijmsc.v2i2.95.
[35] D. Mardiati and Y. Saputra, "Implementasi Sistem Informasi Manajemen Klinik Menggunakan Metode Black Box Testing," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 1, 2025, https://doi.org/10.23960/jitet.v13i1.6015.
[36] M. Tavakol and R. Dennick, "Making sense of Cronbach's alpha," *Int. J. Med. Educ.*, vol. 2, p. 53, Jun. 2011, https://doi.org/10.5116/IJME.4DFB.8DFD.
[37] M. A. Bujang, E. D. Omar, and N. A. Baharum, "A Review on Sample Size Determination for Cronbach's Alpha Test: A Simple Guide for Researchers," *Malays. J. Med. Sci.*, vol. 25, no. 6, p. 85, 2018, https://doi.org/10.21315/MJMS2018.25.6.9.
[38] K. S. Taber, "The Use of Cronbach's Alpha When Developing and Reporting Research Instruments in Science Education," *Res. Sci. Educ. 2016 486*, vol. 48, no. 6, pp. 1273–1296, Jun. 2017, https://doi.org/10.1007/S11165-016-9602-2.
[39] J. van Riet, I. Malavolta, and T. A. Ghaleb, "Optimize along the way: An industrial case study on web performance," *J. Syst. Softw.*, vol. 198, p. 111593, 2023, https://doi.org/10.1016/j.jss.2022.111593.
[40] M. Bolanowski, M. Ćmil, and A. Starzec, "New Model for Defining and Implementing Performance Tests," *Futur. Internet*, vol. 16, no. 10, 2024, https://doi.org/10.3390/fi16100366.
[41] Z. Henez, O. Article, and I. Science, "The role of artificial intelligence in library management systems," *J. Vis. Perform. Arts*, vol. 5, pp. 803–806, 2024, https://doi.org/10.29121/shodhkosh.v5.i5.2024.456.
[42] D. J. Borgohain, R. K. Bhardwaj, and M. K. Verma, "Mapping the literature on the application of artificial intelligence in libraries (AAIL): a scientometric analysis," *Libr. Hi Tech*, vol. 42, no. 1, pp. 149–179, Feb. 2024, https://doi.org/10.1108/LHT-07-2022-0331.
[43] P. Vasishta, N. Dhingra, and S. Vasishta, "Application of artificial intelligence in libraries: a bibliometric analysis and visualisation of research activities," *Libr. Hi Tech*, vol. 43, no. 2–3, pp. 693–710, May 2025, https://doi.org/10.1108/LHT-12-2023-0589.
[44] Z. Ghinafikar, M. M. Mu'thy, and M. A. Yaqin, "Perbandingan Metode Agile dan Waterfall Berdasarkan Analisis Waktu Pengembangan Sistem," *J. Manaj. Teknol. Inform.*, vol. 3, no. 1, pp. 26–44, 2025, [Online]. Available: https://doi.org/10.70038/jentik.v3i1.149.