

DEVELOPMENT OF AN OFFLINE ANDROID-BASED LIBRARY BORROWING SYSTEM USING SQLITE DATABASE

***Lisa Almayanti¹, M. Khizbullah², Ardian³**

Information Technology Education, STKIP Taman Siswa Bima, Indonesia^{1,2,3}

lisaalmayanti@gmail.com, Mbullax71@gmail.com, Ardpadende@gmail.com

**Corresponding Author*

ABSTRACT (10pt, Bold)

Many institutional libraries still rely on manual borrowing records, causing difficulties in transaction tracking, stock monitoring, and accurate fine calculation. This condition reduces service efficiency and increases the risk of data errors in library administration. To address this problem, this study develops an Android-based Library Book Borrowing System as a digital library management solution. The system was developed using a structured Software Development Life Cycle (SDLC) approach consisting of analysis, design, implementation, and testing phases, employing the Java programming language and a locally embedded SQLite relational database. Unlike cloud-dependent approaches, the proposed system operates entirely without an internet connection, making it suitable for institutions with limited network infrastructure. The system encompasses five core modules: book management, member registration, borrowing and return transactions, automated fine computation, and reporting. A distinctive three-tier validation mechanism—comprising stock availability verification, membership status authentication, and outstanding fine arrear inspection—enforces transactional integrity and prevents unauthorized borrowing. Black-box testing confirmed that all 10 primary functional features achieved a 100% success rate. Media expert validation further affirmed the system's high feasibility for deployment as a fully operational digital library management solution.

Keywords : Library Management System, Android Development, SQLite Database, Offline System, Automated Fine Calculation.

1. Introduction

The library is one of the most important academic infrastructure facilities within educational institutions, serving as a centralized repository of textual and reference resources for students, scholars, and the broader academic community. Ensuring the operational efficiency and accuracy of book lending management is indispensable to maintaining high-caliber library services. In the case of the studied institution, the observed library manages hundreds of book collections and student borrowing transactions on a daily basis. However, manual recording methods have caused difficulties in monitoring availability, tracking borrowing history, and calculating overdue penalties accurately. These conditions directly reduce the quality of library services and increase the administrative burden on staff.

Notwithstanding advancements in information technology, a substantial proportion of libraries continues to depend on conventional manual recording practices utilizing ledger books or physical borrowing cards. Manual recording increases the risk of duplicate records, delays transaction verification, and requires additional administrative effort when retrieving borrowing histories or calculating overdue penalties. Such approaches are beset by critical limitations, including susceptibility to data loss and physical deterioration, prolonged retrieval processes, inability to monitor real-time book availability, and a pronounced tendency toward inaccuracies in overdue fine computation (Prayitno, 2019). These operational impacts directly reduce the quality of library services and increase the administrative workload of library staff.

The pervasive proliferation of mobile technology has rendered Android-based smartphones an integral component of contemporary daily life. Exploiting the Android platform as the foundation for a library information system constitutes an architecturally sound solution, given the platform's inherent accessibility, portability, and capacity to operate independently of internet infrastructure (Safitri, 2020).

Previous studies have developed Android-based library systems; however, many rely on online connectivity and provide limited transaction validation mechanisms. Therefore, this study

focuses on developing an offline Android-based Library Book Borrowing System Application with multi-level borrowing validation and automated fine calculation, constructed upon the Java programming language and SQLite database engine. This study introduces an offline Android-based library management architecture integrating three-tier transaction validation and automated fine calculation mechanisms, with the aim of providing an operationally efficient, computationally precise, and ergonomically intuitive solution for library administrative personnel.

2. Literature Review

Library Information Systems

A library information system is defined as an integrated system specifically engineered to administer the full spectrum of library operational activities, encompassing the management of bibliographic collections, patron membership, book lending, and return processing (Supriyanto, 2018). The primary purpose of such systems is to enhance administrative efficiency and elevate data accuracy within library environments. Previous library information systems have improved administrative efficiency; however, several studies still rely on centralized online architectures and provide limited support for offline operational environments. Furthermore, many existing systems were developed using web-based platforms that require stable internet connectivity, making them less suitable for institutions with limited network infrastructure. This study addresses these gaps by developing an offline Android-based system using a locally embedded SQLite database, offering a more accessible and portable solution for library management.

Android Application Development with Java

Android is a Linux kernel-based operating system purpose-built for mobile computing devices, widely adopted across diverse institutional and educational contexts due to its accessibility and affordability (Hermawan, 2020). Android offers practical advantages for institutional applications by supporting affordable mobile deployment, local database integration, and offline functionality in environments with limited connectivity. Java serves as the predominant programming language for Android application development due to its object-oriented paradigm, cross-platform portability, and extensive ecosystem of development libraries. Unlike web-based solutions that depend on continuous server access, Android applications built with Java can operate independently on local devices, making them particularly suitable for library environments where internet infrastructure may be unreliable. Android Studio constitutes the officially designated Integrated Development Environment (IDE) for the Android development toolchain.

SQLite as an Embedded Local Database Engine

SQLite is a self-contained, server-independent, lightweight relational database management system (RDBMS) natively embedded within the Android runtime environment. Operating as an in-process component, it eliminates the need for a dedicated database server instance (Kusuma, 2019). SQLite was selected for this study because its embedded architecture enables local data processing without requiring an external database server, making it suitable for offline library management applications. The SQLiteOpenHelper class, a component of the Android SDK, provides a standardized abstraction layer for database schema lifecycle management, enabling efficient creation, updating, and querying of local data without network dependency.

Digital Library Lending Systems

The digitalization of library borrowing workflows confers substantial operational advantages over their manual counterparts, including accelerated data retrieval, deterministic fine computation, real-time bibliographic availability monitoring, and materially enhanced data integrity and security (Nugroho, 2021).

3. Research Methods

This study applies the Waterfall model consisting of five sequential phases: requirements elicitation and analysis, system design and architecture, implementation, functional testing, and expert validation and evaluation. The Waterfall model was selected due to its structured and linear development process, which is well-suited for projects with clearly defined requirements and a fixed scope, as is the case in this library borrowing system development.

Requirements Elicitation and Analysis

The initial phase entails a systematic identification of operational deficiencies encountered by library administrators within existing manual recording workflows. Requirement data were collected through direct observation and interviews with library administrators to identify problems in book management, borrowing transactions, and reporting processes. The study was conducted at the library of STKIP Taman Siswa Bima during the 2026 academic year, involving two library staff members as respondents. System requirements are subsequently categorized into functional specifications—encompassing bibliographic management, patron administration, transactional processing, and report generation—and non-functional specifications, including cross-device portability, usability, and offline operational capability.

System Design and Architecture

The system design phase encompasses the construction of process workflow diagrams, the formalization of borrowing validation algorithms, the delineation of the application's navigational menu hierarchy, and the specification of the relational database schema. The database schema is structured around four primary relational entities: Books, Members, Borrowing Transactions, and Staff Records. The primary workflow diagram of the application is presented in Figure 1.

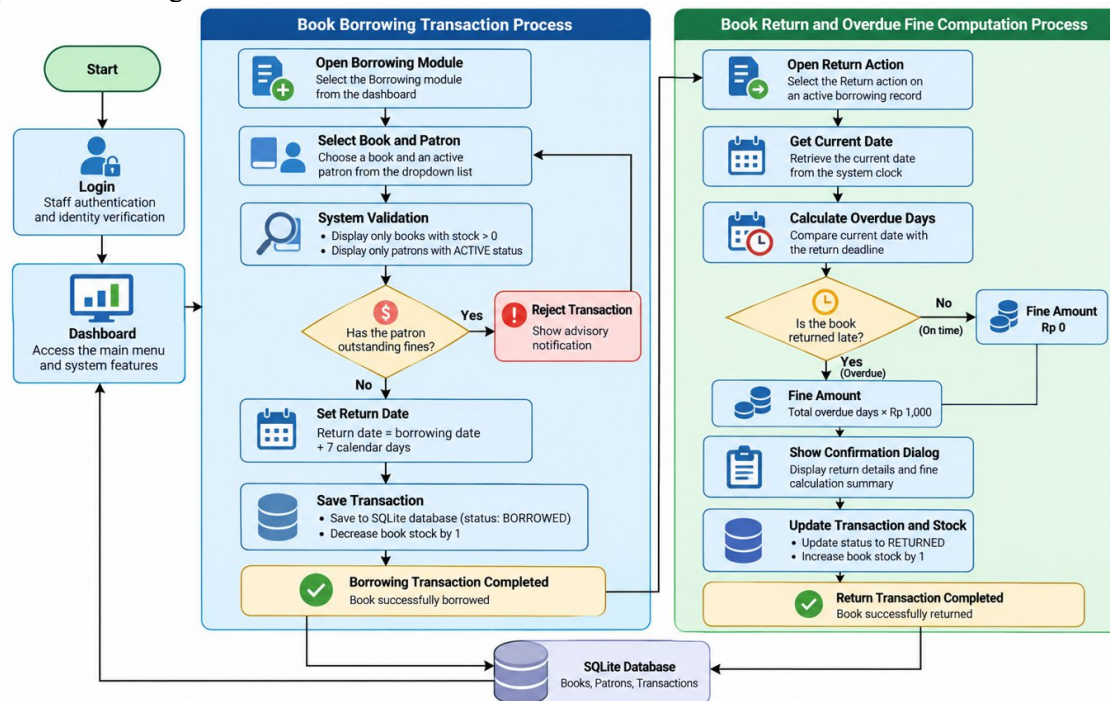


Fig. 1. Process Workflow Diagram of the Library Book Borrowing System Application

Book Borrowing Transaction Algorithm

The procedural logic governing borrowing transaction validation is executed through the following sequential steps:

- Administrative staff initiates a borrowing transaction by selecting the Borrowing module from the Dashboard interface;
- The system renders a modal dialog incorporating dropdown selectors for book and patron selection;
- The system filters the book inventory to display only entries with available stock quantities exceeding zero;
- The system filters the patron registry to present only records with ACTIVE membership status;
- In the event that the selected patron carries unresolved outstanding fines, the system rejects the transaction and triggers an advisory notification;
- The system automatically computes and assigns the return deadline as the borrowing date incremented by seven calendar days;

- Transaction data is committed to the SQLite database repository with a status designation of BORROWED.
- The corresponding book's stock inventory is automatically decremented by one unit.

Overdue Fine Computation Algorithm

The algorithmic workflow governing overdue fine computation and book return processing proceeds as follows:

- Administrative staff initiates the return process by activating the Return action control on the corresponding active borrowing record;
- The system retrieves the current date from the device's system clock;
- The system performs a temporal comparison between the current date and the recorded return deadline;
- In cases of late return: fine amount = total number of overdue days $\times$ Rp 1,000;
- In cases of on-time return: fine amount = Rp 0;
- The system displays a return confirmation dialog inclusive of a detailed fine calculation summary;
- The borrowing transaction record status is updated to RETURNED;
- The corresponding book's stock inventory is automatically incremented by one unit.

Application Navigational Architecture

The application's module hierarchy is structured as follows: Login (staff identity authentication), Dashboard (real-time operational statistics and primary navigation hub), Book Data Management (bibliographic entry creation, modification, deletion, and keyword search), Patron Data Management (member registration, editing, deletion, access control, and status verification), Borrowing Transactions (transaction recording and return processing with integrated fine computation), Reports (tabbed interface encompassing Active Borrowings, Overdue Items, and Fine Ledger), Transaction History (comprehensive audit log of all processing events), and System Logout.

Implementation

System implementation was executed within Android Studio utilizing the Java programming language. The source code is organized into four modular packages: the model layer (Buku.java, Peminjam.java, Peminjaman.java), the data access layer (DatabaseHelper.java), the presentation layer comprising seven primary Activity classes, and the adapter layer consisting of five RecyclerView Adapter implementations. The DatabaseHelper.java class encapsulates all SQLite database operations, including schema initialization, record insertion, modification, deletion, and structured query execution.

Testing

Functional evaluation was conducted employing the black-box testing methodology to systematically verify the behavioral correctness of each implemented feature. Test scenarios encompassed nominal operational flows, boundary-condition edge cases, and validation constraint checks designed to confirm that the system accurately rejects malformed or unauthorized transaction requests. Conclusive feasibility assessment was administered by a designated media expert to appraise the overall deployment readiness of the application.

4. Results and Discussions

Results

Technical Specifications

The application is constructed upon an Activity-based architectural paradigm employing the Adapter design pattern for RecyclerView-driven list rendering. The implementation results demonstrate that the developed system successfully replaces manual borrowing procedures by providing automated transaction validation, inventory monitoring, and digital reporting features. These capabilities directly address the operational deficiencies identified during the requirements analysis phase, including difficulties in tracking borrowing history, monitoring book availability, and calculating overdue penalties. Table 1 enumerates the comprehensive technical specifications of the developed system.

Table 1. Technical Specifications of the Developed Application

Component	Specification
Programming Language	Java (JDK 1.8)
Integrated Development Environment	Android Studio Hedgehog 2023.1
Database Engine	SQLite (natively embedded in Android)
Minimum SDK Version	API Level 21 — Android 5.0 (Lollipop)
Target SDK Version	API Level 34 — Android 14
Architectural Pattern	Activity-Based with Adapter Pattern
Data Persistence Mechanism	Local SQLite (fully offline; no network dependency)
Compiled APK Size	±6–7 MB

Application User Interface

The following subsection delineates the user interface design of each implemented module. All screens maintain a cohesive visual identity anchored by a consistent red chromatic theme aligned with the application's institutional branding guidelines.

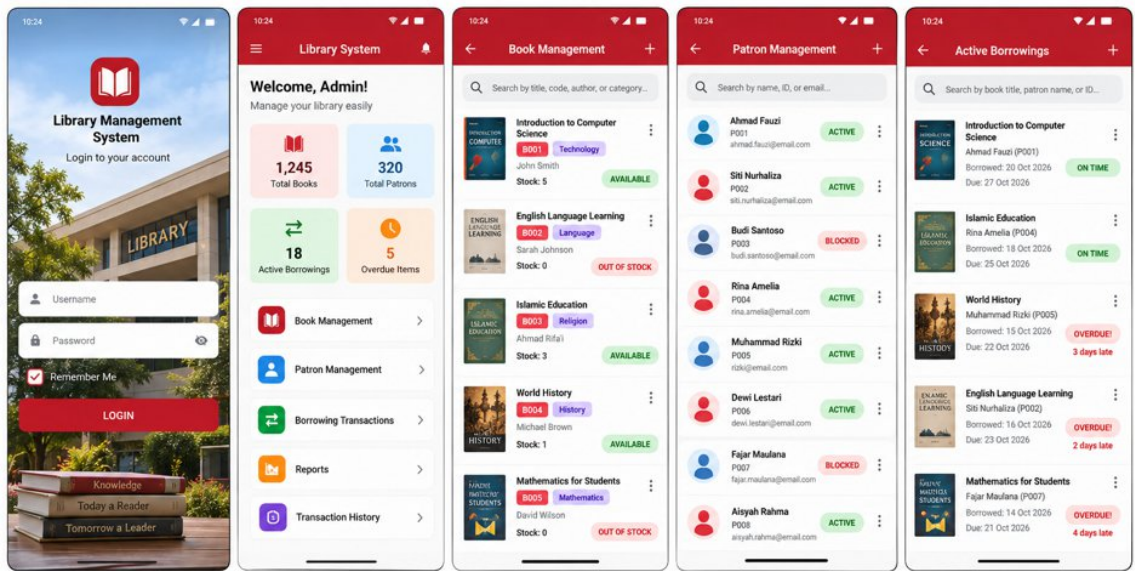


Fig. 2. User Interface of the Library Management System Application

The figure presents the implemented user interfaces of the Android-based library management system, comprising five main screens: authentication, dashboard, bibliographic management, patron management, and active borrowing transactions. These interfaces were developed to support the daily administrative activities of library staff through an integrated mobile application with a consistent visual design and a red-themed color scheme. Each screen provides specific functionalities that contribute to the efficiency, accuracy, and accessibility of library management operations.

The authentication screen serves as the entry point to the application, providing username and password fields for staff identification and a “Remember Me” option to maintain the login session. The dashboard functions as the primary navigation hub and displays four operational indicators: total books, total registered patrons, active borrowing transactions, and overdue items. These indicators provide staff with an overview of the library's current operational status, while the navigation menu provides direct access to the book management, patron management, borrowing transactions, reports, and transaction history modules.

The bibliographic management screen enables staff to access and monitor the book collection through a searchable catalog. The search function supports filtering by book title, item code,

author, and category. Each book entry displays its title, author, classification, item code, available stock quantity, and availability status. Color-coded labels distinguish book categories and inventory conditions, including available and out-of-stock items. This functionality facilitates book identification, supports inventory monitoring, and helps staff determine whether a requested book can be borrowed.

The patron management screen provides a centralized interface for accessing registered library member records. Each entry presents the member's name, identification code, contact information, and membership status. The application distinguishes active and blocked members using green and red status indicators, respectively. This feature assists staff in verifying membership eligibility and managing access to borrowing services. The displayed information also supports the identification of individual patrons during administrative and borrowing transactions.

The active borrowing transactions screen presents records of books that have been borrowed but not yet returned. Each transaction includes the book title, borrower identity, borrowing date, and scheduled return date. The application uses color-coded indicators to differentiate transactions that remain within the permitted borrowing period from those that have exceeded their return deadlines. Overdue transactions are highlighted with a red notification accompanied by the number of overdue days, enabling staff to identify delayed returns and prioritize follow-up actions. This interface supports systematic transaction monitoring and provides essential information for the implementation of the application's return-processing and overdue-fine computation features.

Overall, the implemented interfaces demonstrate the operational integration of authentication, dashboard monitoring, bibliographic management, patron administration, and borrowing transaction tracking. The combination of searchable records, real-time operational indicators, membership status verification, inventory availability labels, and overdue notifications reduces reliance on manual recordkeeping and facilitates more structured library administration. The interfaces also provide the visual foundation for the application's reporting, transaction history, and automated borrowing and return-processing functions.

Feature Implementation and Testing Outcomes

Table 2 presents a consolidated summary of the implementation status and functional test outcomes for all application modules. All features demonstrated correct operational behavior across Android devices running version 5.0 (API Level 21) and above.

Table 2. Application Feature Implementation and Testing Results

No.	Feature Module	Functional Description	Status
1	Staff Authentication	Credential validation against the SQLite database with persistent session support	✓ Success
2	Bibliographic Management	CRUD operations on book records, categorical classification, and automated stock synchronization	✓ Success
3	Patron Administration	CRUD operations on patron records, access control toggling, status verification, and arrear tracking	✓ Success
4	Borrowing Transaction Processing	Three-tier validation enforcement with automated return deadline assignment (+7 days)	✓ Success
5	Book Return and Fine Settlement	Automated fine computation (Rp 1,000 per overdue day) with confirmation dialog	✓ Success
6	Report Generation	Tri-tabbed reporting: Active Borrowings, Overdue Items, and Unpaid Fine Ledger	✓ Success
7	Transaction History	Comprehensive transaction log with fine	✓ Success

	Auditing	payment status indicators	
8	Dashboard Statistics	Four real-time operational metrics with five primary navigation controls	✓ Success
9	Real-Time Catalog Search	Keyword-based filtering across title, item code, authorship, and categorical fields	✓ Success
10	Visual Theme Consistency	Uniform red header and floating action button (FAB) styling across all screens	✓ Success

Discussion

Three-Tier Validation Architecture

A principal differentiating capability of the developed system is its three-tier transactional validation framework governing borrowing operations. The primary validation layer performs a real-time examination of bibliographic stock availability to preclude lending of physically unavailable items. The secondary validation tier enforces patron membership authentication, permitting borrowing privileges exclusively to records designated with an ACTIVE status. The tertiary validation stratum conducts an inspection of outstanding fine arrears, disallowing the initiation of new borrowing transactions by patrons with unresolved penalty obligations. This layered enforcement mechanism demonstrated complete efficacy in intercepting all malformed or unauthorized transactions throughout the evaluation phase.

Automated Fine Computation Mechanism

The penalty computation subsystem leverages Java's SimpleDateFormat API in conjunction with temporal difference calculations expressed in milliseconds—equivalent to 86,400,000 milliseconds per calendar day—achieving a deterministic calculation precision rate of 100% across all evaluated test scenarios. A standardized penalty rate of Rp 1,000 per overdue day is persistently recorded within the SQLite database, with staff-initiated payment confirmation accessible through the Reports module interface.

Technical Challenges and Corresponding Mitigations

Four principal engineering challenges were identified and resolved throughout the development lifecycle:

- SQLite referential integrity enforcement triggered application runtime exceptions during cascading deletion operations; this was mitigated by implementing preemptive dependency verification prior to the execution of DELETE statements;
- Fine computation inaccuracies attributable to device timezone inconsistencies were rectified by adopting millisecond-epoch-based temporal arithmetic in lieu of string-formatted date comparisons;
- Memory leakage anomalies within RecyclerView and Dialog component lifecycles were resolved by enforcing mandatory dismissal of Dialog instances prior to instantiation of new overlay objects;
- Android 5.0 (API Level 21) compatibility regressions were remediated through the application of targeted style attribute overrides within the res/values-v21/ resource directory. All identified challenges were successfully resolved, yielding a stable and production-ready application artifact.

5. Conclusion

This study investigated the development of an offline Android-based Library Book Borrowing System at STKIP Taman Siswa Bima as a solution to address inefficiencies in manual library administration, particularly in transaction recording, book availability monitoring, and automated fine calculation. The system was implemented using Java programming language and a locally embedded SQLite database, following the Waterfall development model. All ten

evaluated functional modules successfully passed the defined black-box testing scenarios. The three-tier validation framework effectively enforces transactional integrity by preventing unauthorized borrowing operations; the automated fine computation mechanism achieves deterministic accuracy; and the application maintains full operational functionality in offline environments without any network dependency. Expert validation by media specialists confirms the system's high degree of feasibility for deployment as a comprehensive digital library management platform. These findings demonstrate that an offline Android-based approach constitutes a practical and reproducible solution for digitizing library services in institutions with limited network infrastructure, and may serve as a basis for future development incorporating cloud synchronization, barcode scanning, and push notification features. Prospective enhancements recommended for subsequent development iterations include: (1) Integration of remote supervisory capabilities via IoT-enabled Wi-Fi communication modules; (2) Implementation of report export functionality to Excel (XLSX) and Portable Document Format (PDF); (3) Incorporation of push notification services to deliver automated return deadline reminders to patrons; (4) Cloud-based data synchronization through Firebase Firestore integration; (5) Deployment of barcode scanning capabilities to accelerate transactional throughput; (6) Execution of formal User Acceptance Testing (UAT) protocols with operational library administrative personnel.

References

- Adikusuma, R. W., Al-Fikri, A. M., Bhakti, M. E. N., & Wardhana, S. R. (2023). Rancang Bangun Sistem Informasi Perpustakaan Pada SMP Negeri 2 Gedangan Menggunakan Metode Waterfall. *Kernel*, 4(2). <https://doi.org/10.31284/j.kernel.2023.v4i2.5578>
- Agustin, R. D., Krisdiyanto, W., & Rifan, M. (2024). Aplikasi Perpustakaan Berbasis Android untuk Mempermudah Peminjaman dan Pengembalian Buku di SMPN 2 Pubian. *Journal of Computer Science and Informatics (JOCSI)*, 2(1). <https://doi.org/10.69747/jocsi.v2i1.64>
- Ainy, A. H., Diantoro, K., Rohman, A., Juwari, J., & Rinaldo, R. (2026). Perancangan Sistem Perpustakaan Digital Berbasis Android dengan QR Code pada MTsN 2 Kota Bekasi. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, 7(3), 380-387. <https://doi.org/10.63447/jimik.v7i3.1980>
- Choiri, M. A., Rachman, A., Purwadi, A., & Salim, A. K. (2021). Rancang Bangun Sistem Informasi Perpustakaan Sekolah Berbasis Web di SMK Islam Al-Futuhiyyah Menggunakan Model Waterfall. *Prosiding SNESTIK*. <https://doi.org/10.31284/p.snestik.2021.1798>
- Mahardika, F., Zulfan, A., & Suseno, A. T. (2023). Implementasi Metode Waterfall pada Sistem Informasi Kepegawaian Berbasis Web. *Blend Sains: Jurnal Teknik*, 2(2). <https://doi.org/10.56211/blendsains.v2i2.300>
- Meiyanti, R. (2022). Rancangan Aplikasi Perpustakaan Berbasis Android di Perpustakaan Universitas Malikussaleh. *Jurnal Teknologi Terapan & Sains*, 3(2). <https://doi.org/10.29103/tts.v3i2.8275>
- Mohammed, D. Y., & Cooper, P. A. (2022). Developing cross-platform library using Intel Multi-OS Engine. *European Journal of Electrical Engineering and Computer Science*, 6(2), 15–17. <https://doi.org/10.24018/ejece.2022.6.2.422>
- Musyaffa, M. A., & Sancoko, S. D. (2026). Pengembangan Aplikasi Perpustakaan Digital Berbasis Mobile Untuk Efisiensi Layanan Informasi Di Era Digital. *Jutisi : Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, 14(3), 2317–2327. <https://doi.org/10.35889/jutisi.v14i3.3356>
- Nurrahman, D. A., & Agustina, D. R. (2024). Peminjaman Buku Perpustakaan Modern Lampung Berbasis Android. *Digital Transformation Technology*, 4(2). <https://doi.org/10.47709/digitech.v4i2.4595>
- Pasaribu, R. F., & Azhari, F. (2024). Implementasi Database SQLite pada Data Mahasiswa Universitas Labuhanbatu Berbasis Android. *Jurnal Ilmiah Informatika Global*, 1(1). <https://doi.org/10.58369/git.v1i1.83>
- Rahmad, A., & Hidayat, F. (2020). Sistem Informasi Perpustakaan di Universitas Batam Berbasis Android. *Zona Komputer*, 10(1). <https://doi.org/10.37776/zk.v10i1.489>

- Rangkuti, F. W. P., & Alda, M. (2025). Pengembangan Aplikasi Android Sistem Informasi Perpustakaan Sekolah. *Teknik: Jurnal Ilmu Teknik dan Informatika*, 5(2). <https://doi.org/10.51903/teknik.v5i2.925>
- Sakinah, M., & Alda, M. (2025). Sistem Informasi Perpustakaan SMP Negeri 1 Pegajahan Dengan Fitur Notifikasi Untuk Monitoring Peminjaman Buku Berbasis Mobile Dengan Metode Collaborative Filtering. *Teknologi: Jurnal Ilmiah Sistem Informasi*, 15(2). <https://doi.org/10.26594/teknologi.v15i2.5733>
- Wiguna, E. E. S., Lie, H., Niro, L., Linggarjati, J., Atmadja, W., Hutabarat, D. P., & Hedwig, R. (2024). Raspberry Pi-Based Offline Digital Library for Indonesian Villages Without Stable Power and Internet Access. *Information Technology and Libraries*, 43(4). <https://doi.org/10.5860/ital.v43i4.16869>
- Yuda, A., & Muludi, K. (2021). Aplikasi Perpustakaan Digital Berbasis Android pada Perpustakaan Jurusan Ilmu Komputer Universitas Lampung. *Pepadun*, 2(1). <https://doi.org/10.23960/pepadun.v2i1.28>