



Article

Aplikasi Desktop Manajemen Data Buku untuk Solusi Praktis untuk Pengelolaan Digital

Agriby Diandra Chaniago^{1*}, Annastasya Nabila Elsa Wulandari¹, Tri Styo Famuji²

¹Program Studi Informatika, Universitas Harapan Bangsa, Purwokerto, Indonesia

²Program Studi Teknologi Informasi, Universitas Harapan Bangsa, Purwokerto, Indonesia

* Correspondence: agribychaniago@gmail.com

Abstract: *In the digital era, book data management is an important challenge for libraries, bookstores, and private collections. This article discusses a Java-based book data management application designed with an Object-Oriented Programming (OOP) approach for modularity, efficiency, and flexibility. Key features include CRUD operations, data retrieval, recording of borrowing and book return transactions, and fine calculation, with MySQL integration for accurate, stable, and structured relational data storage. Security is strengthened through SHA-256 encryption on the login feature, ensuring limited access to authenticated users only. Testing showed stable performance, with improvements to the interface design and database connections to improve the user experience. This application not only supports efficient data management, but also offers flexibility in adjusting user needs. The development plan includes data export, statistical reports, mobile device compatibility, and interface improvements to provide comprehensive, responsive, and adaptive solutions to meet the needs of future book data management.*

Keywords: Book Data, Management, Java Swing, MySQL and JDBC, CRUD Applications, SHA-256

Received: 23 December 2024

Revised: 8 January 2025

Accepted: 9 January 2025

Published: 9 January 2025



Copyright: © 2025 by the authors.

License Universitas Harapan Bangsa, Purwokerto, Indonesia. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Pendahuluan

Pengelolaan data buku yang efisien menjadi kebutuhan penting bagi institusi seperti perpustakaan dan toko buku, mendorong pengembangan aplikasi berbasis teknologi untuk pencatatan dan pencarian data digital (Pang, 2022). Bahasa pemrograman Java dengan pendekatan *Object-Oriented Programming* (OOP), menawarkan struktur modular, skalabilitas, serta mendukung operasi CRUD, transaksi peminjaman,

pengembalian, dan perhitungan denda (Sharma et al., 2022). OOP memungkinkan modularitas dan pengembangan independen komponen seperti manajemen data buku dan transaksi, serta penambahan fitur baru tanpa merombak sistem (Onu et al., 2024).

Java terintegrasi dengan MySQL untuk penyimpanan data terstruktur, menjaga integritas dan konsistensi dengan penerapan *primary key* dan *foreign key* (Jiang & Naumann, 2020); (Györödi et al., 2021). Keamanan aplikasi ditingkatkan dengan enkripsi SHA-256 untuk login, memastikan akses terbatas pada pengguna terotorisasi (Franck et al., 2023). Dengan fleksibilitas untuk integrasi API eksternal dan analisis data, aplikasi berbasis Java menjadi solusi pengelolaan data buku yang dapat berkembang untuk memenuhi kebutuhan masa depan.

Metodologi Pembuatan Aplikasi

Pengembangan aplikasi dilakukan melalui tahapan sistematis untuk memastikan kebutuhan pengguna terpenuhi, desain fungsional, dan aplikasi berfungsi optimal.

1. Analisis Kebutuhan

Tahap ini mencakup identifikasi kebutuhan pengguna, seperti fungsi CRUD, pencarian data, validasi input, serta manajemen data Buku, Peminjam, Peminjaman, dan Pengembalian. Struktur data dirancang dengan entitas utama beserta atribut dan hubungan antar entitas untuk menjaga integritas data (Wei, 2024).

2. Desain

Diagram alur, seperti *flowchart* dan *use case diagram*, digunakan untuk menggambarkan alur kerja sistem. Struktur basis data dimodelkan menggunakan diagram ER, sementara antarmuka pengguna dirancang menggunakan Java Swing dengan fokus pada elemen seperti panel, dialog, formulir input, dan tabel data.

3. Implementasi

Aplikasi dikembangkan menggunakan Java dan MySQL. Java Swing membangun antarmuka pengguna, dengan folder dialog menangani operasi data dan autentikasi (login dengan SHA-256), folder panel menampilkan data, dan folder util menyimpan utilitas seperti DatabaseUtil. Struktur kode terorganisasi dan file utama mengatur alur aplikasi.

4. Pengujian

Pengujian mencakup verifikasi fungsi CRUD, pencarian data, validasi input, serta integrasi komponen, termasuk koneksi database. Bug yang ditemukan diperbaiki untuk memastikan kualitas aplikasi sebelum rilis.

Teknologi yang Digunakan

Pengembangan aplikasi ini memanfaatkan berbagai teknologi untuk memastikan kinerja, keamanan, dan fleksibilitas:

1. IDE

IntelliJ IDEA mendukung pengembangan berbasis Java dengan fitur *refactoring*, *debugging*, dan integrasi sistem kontrol versi seperti Git. Selain itu, IntelliJ mempermudah pengelolaan pustaka melalui Maven dan Gradle (Kaczorowski, 2023).

2. Bahasa Pemrograman

Java Development Kit (JDK) versi 22 dipilih untuk fitur-fitur modern seperti optimalisasi mesin virtual, algoritma enkripsi terbaru, dan efisiensi pengembangan, yang memastikan kompatibilitas dengan teknologi terkini.

3. Database

MySQL digunakan sebagai RDBMS untuk pengelolaan data terstruktur. Fitur seperti transaksi atomik, optimasi query, dan fleksibilitas skema mendukung stabilitas sistem. Komunikasi dilakukan melalui JDBC untuk pengelolaan koneksi dan eksekusi query (Żuchnik & Kopniak, 2021).

4. Pustaka Tambahan

Pustaka tambahan yang digunakan dalam pengembangan aplikasi ini meliputi MySQL Connector/J, yaitu driver JDBC untuk koneksi ke MySQL dengan dukungan SSL, enkripsi, dan efisiensi koneksi. Selain itu, JCalendar digunakan sebagai pustaka GUI berbasis Java Swing untuk menyediakan widget kalender interaktif yang mempermudah pemilihan tanggal secara visual serta mengurangi risiko kesalahan input data waktu (Toedter.com, n.d.). Algoritma SHA-256 diterapkan untuk mengenkripsi kata sandi, yang meningkatkan keamanan login dengan mengubah kata sandi menjadi hash yang tidak dapat dibaca, memungkinkan verifikasi data pengguna tanpa menyimpan kata sandi asli (Anwar et al., 2021).

Arsitektur Sistem Manajemen Data Buku

Arsitektur sistem ini mencakup hubungan antara komponen utama untuk mendukung pengelolaan inventaris buku secara efisien dan aman. Setiap komponen memiliki fungsi yang saling melengkapi untuk memastikan kinerja sistem yang optimal.

1. User Interface (UI)

UI mencakup panel untuk mengelola data dan dialog untuk menambah, mengedit, atau menghapus informasi buku, peminjam, peminjaman, dan pengembalian. *Form login* menggunakan algoritma SHA-256 untuk meng-hash kata sandi sebelum disimpan di database, menjaga keamanan data pengguna (Anwar et al., 2021).

2. Logic Layer

Logic Layer menghubungkan UI dengan *database*, mengelola logika aplikasi, validasi data, dan autentikasi pengguna. Proses autentikasi memverifikasi kata sandi yang di-hash dengan hash di database, memastikan akses hanya untuk pengguna terotorisasi (Litvinov, 2020).

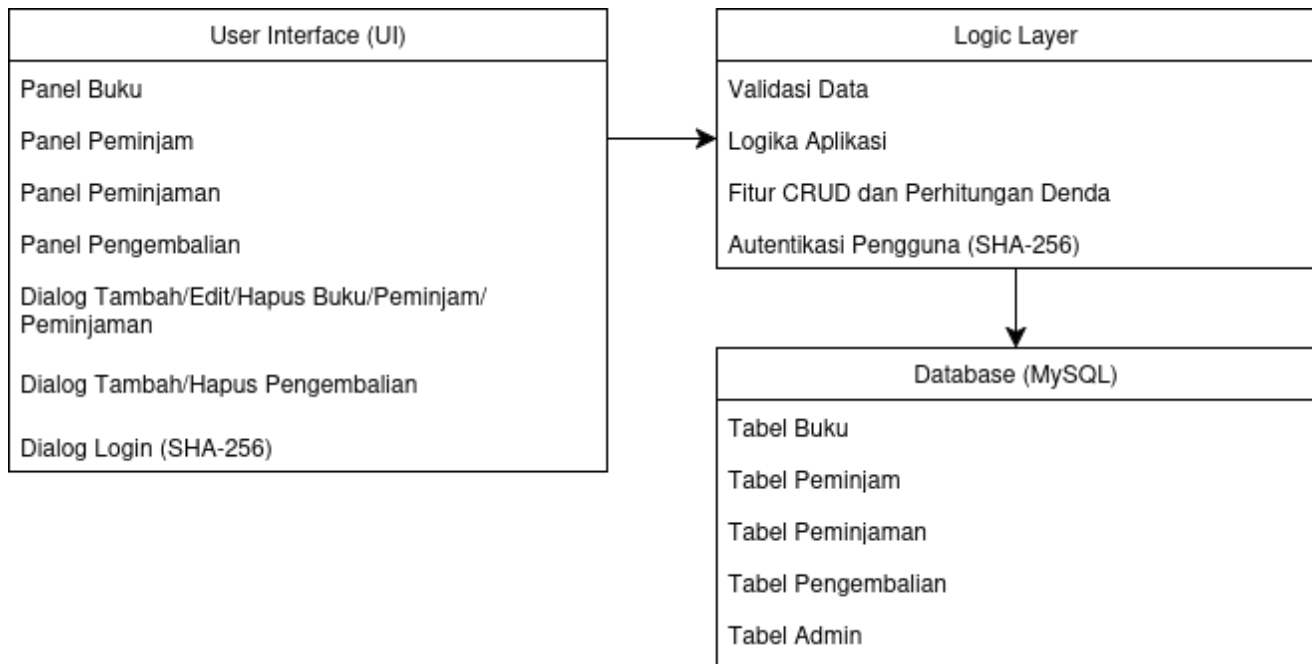
3. Database (MySQL)

Database menyimpan data dalam tabel terorganisir seperti Buku, Peminjam, dan Peminjaman. MySQL Connector/J digunakan untuk komunikasi antara aplikasi Java dan database. Keamanan data dijamin dengan pengaturan hak akses dan penguncian data (Blagojche et al., 2023).

Hubungan Antar Komponen

Tiga komponen utama, yaitu UI, *Logic Layer*, dan *Database*, berkolaborasi secara sinergis untuk menciptakan alur kerja yang efisien dan terorganisir. UI bertindak sebagai antarmuka pengguna dan *Logic Layer* memproses serta memvalidasi data, sementara *Database* menyimpan informasi secara terstruktur. Dalam proses autentikasi, misalnya, kata sandi pengguna di-hash menggunakan algoritma SHA-256 di *Logic Layer*, kemudian dibandingkan dengan hash yang tersimpan di *Database* untuk memastikan keamanan dan keabsahan kredensial. Integrasi ketiga komponen ini memastikan pengelolaan data yang optimal, aman, dan

andal, sebagaimana ditunjukkan pada Gambar 1. Struktur ini dirancang untuk efisiensi dan perlindungan informasi pengguna.

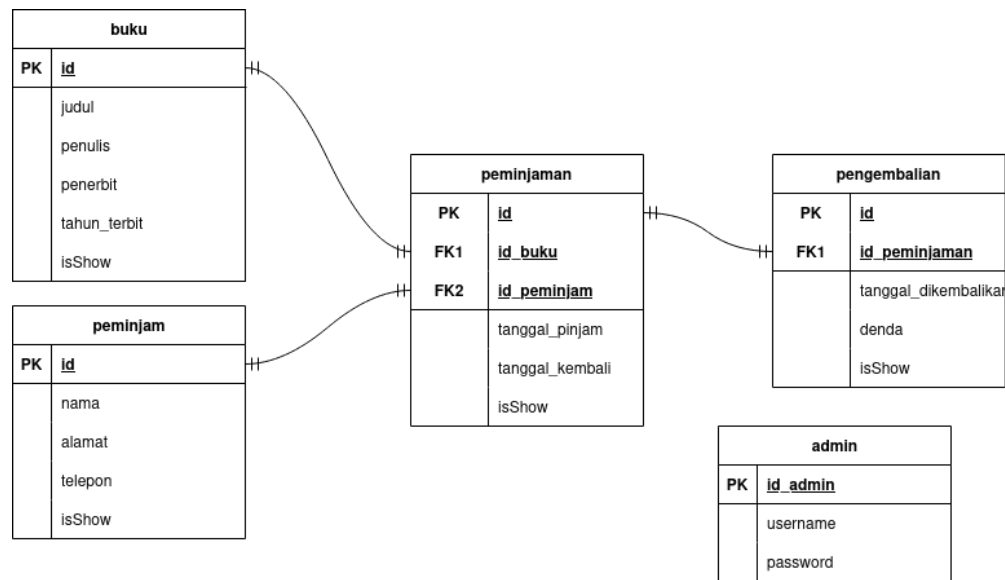


Gambar 1. Diagram Komponen

Implementasi dan Pengujian Sistem

Integrasi dengan Database

Struktur database dirancang untuk efisiensi pengelolaan data. Tabel Buku menyimpan informasi buku dengan ID sebagai kunci utama dan atribut lainnya seperti judul, penulis, penerbit, dan tahun terbit, serta kolom isShow untuk status tampilan. Tabel Peminjam mencatat data peminjam dengan ID sebagai kunci utama, termasuk nama, alamat, dan telepon. Tabel Peminjaman mencatat transaksi peminjaman dengan kolom ID sebagai kunci utama dan relasi ke Buku dan Peminjam, termasuk tanggal peminjaman dan jadwal pengembalian. Tabel Pengembalian mencatat pengembalian buku, dengan ID Peminjaman sebagai foreign key dan atribut tanggal pengembalian serta denda keterlambatan. Tabel Admin menyimpan data administrator dengan ID_Admin, username, dan password terenkripsi. Hubungan antar tabel memastikan integritas data dan pengelolaan yang terorganisir, seperti yang ditunjukkan dalam Gambar 2.

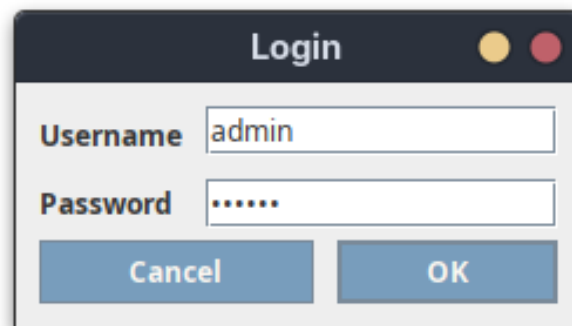


Gambar 2. Desain Database

Pengujian Sistem

Pengujian sistem bertujuan memastikan aplikasi manajemen data buku berjalan optimal dengan menguji fitur CRUD, pencarian, validasi input, keamanan SHA-256 untuk login, dan integrasi MySQL. Skenario pengujian mencakup penambahan, pencarian, pengeditan, penghapusan data, serta verifikasi keamanan login.

1. Modul Proses Login



Gambar 3. Halaman login

Gambar 3 Halaman login menunjukkan tampilan login dengan enkripsi SHA-256 untuk memastikan keamanan autentikasi. Sistem memverifikasi kredensial pengguna dengan mencocokkan data di tabel admin. Pengujian yang tertera di Tabel 1 mencakup kombinasi kredensial yang benar dan salah untuk menguji ketahanan sistem terhadap akses tidak sah. Hasil pengujian menunjukkan hanya pengguna dengan kredensial valid yang dapat mengakses sistem, dengan tingkat akurasi

tinggi. Status "Valid" mengonfirmasi modul login berfungsi sesuai spesifikasi, menjamin keamanan data dan efisiensi autentikasi.

Table 1. Pengujian Login

Pengujian	Keterangan	Status
Proses Login	Sistem memverifikasi kredensial pengguna berdasarkan data pada tabel admin dengan menggunakan enkripsi SHA-256 untuk memastikan keamanan akses.	Valid

2. Modul Panel Data Buku

Gambar 4 menunjukkan modul panel data buku yang mempermudah pengelolaan informasi secara efisien. Pengujian melibatkan fungsi CRUD (Create, Read, Update, Delete) dengan fokus pada akurasi dan validasi input, seperti format numerik pada kolom Tahun Terbit. Hasil pengujian menunjukkan bahwa sistem dapat menangani operasi dengan respons tinggi, menjaga integritas data. Tabel 2 menyajikan hasil pengujian CRUD dan fitur pencarian, yang memverifikasi kelengkapan input, pembaruan data real-time, dan penghapusan data fisik. Fitur pencarian memungkinkan akses cepat berdasarkan kata kunci. Secara keseluruhan, modul ini berfungsi dengan baik dan esensial untuk manajemen data buku.



ID	Judul	Penulis	Penerbit	Tahun Terbit
1	Pemrograman Java Dasar	John Doe	TechPress	2020
2	Belajar Python	Jane Smith	CodeBooks	2021
3	Pemrograman C++	Albert Camus	Programming World	2019
4	Dasar-Dasar SQL	Isaac Newton	DataPress	2022
5	Pemrograman Web	Marie Curie	WebDev	2020
6	Machine Learning	Alan Turing	AI Books	2021
7	Deep Learning	Andrew Ng	AI Future	2023
8	Belajar Kotlin	Ada Lovelace	Mobile Books	2021
9	Pemrograman Swift	Nikola Tesla	iOS Dev	2022
10	Flutter untuk Pemula	Thomas Edison	Mobile UI	2021
11	React.js Modern	Charles Babbage	Frontend Dev	2023
12	Node.js Lanjutan	Tim Berners-Lee	Backend Books	2022
13	Belajar HTML	John Smith	Web Books	2022
14	CSS untuk Pemula	Jane Doe	Web Books	2021
15	JavaScript Modern	Albert Einstein	CodeWorld	2020
16	React.js Dasar	Isaac Newton	FrontEnd	2023
17	Node.js untuk Pemula	Charles Darwin	Backend	2022
18	Pemrograman PHP	Alan Turing	CodeMaster	2019
19	Laravel Framework	Ada Lovelace	PHP Experts	2021
20	Data Science Python	Marie Curie	AI Books	2023

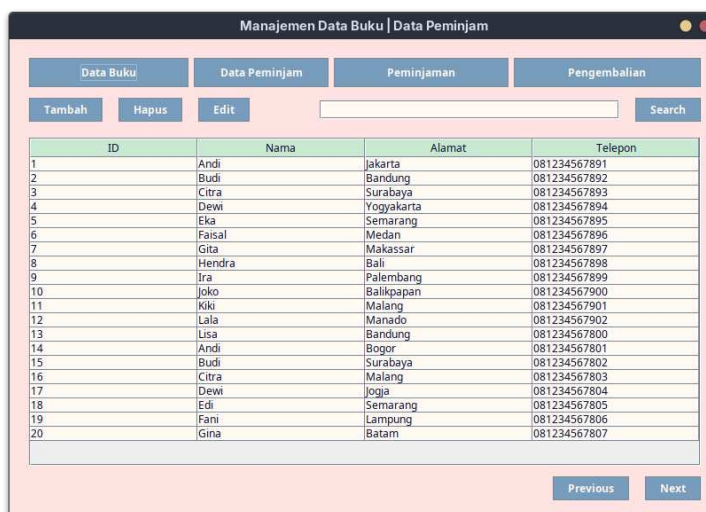
Gambar 4. Panel Data Buku

Table 2. Pengujian CRUD dan Pencarian pada Panel Data Buku

Pengujian	Keterangan	Status
Penambahan Data	Semua kolom, termasuk Judul, Penulis, Penerbit, dan Tahun Terbit, harus diisi dengan benar. Tahun Terbit harus memiliki format numerik dengan empat digit angka.	Valid
Pembaruan Data	Kolom Judul, Penulis, Penerbit, dan Tahun Terbit harus diisi dengan lengkap, dan Tahun Terbit harus berformat numerik dengan empat digit angka.	Valid
Penghapusan Data	Data yang dihapus akan dihilangkan secara permanen dari sistem.	Valid
Pencarian Data	Pencarian data dilakukan dengan menggunakan kolom pencarian untuk menemukan buku berdasarkan kata kunci.	Valid

3. Modul Panel Data Peminjam

Gambar 5 menunjukkan bahwa modul panel data peminjam berfungsi utama dalam aplikasi manajemen data buku untuk mengelola informasi peminjam. Modul ini mendukung fungsi CRUD dan pencarian dengan validasi pada kolom penting seperti Nama Peminjam, Alamat, dan Nomor Telepon. Penghapusan data diuji untuk memastikan bahwa data yang dihapus tidak tersimpan lagi, sementara pencarian memungkinkan akses cepat berdasarkan kata kunci. Hasil pengujian, yang dirangkum dalam Tabel 3, menunjukkan sistem berfungsi dengan akurasi dan efisiensi tinggi dalam memvalidasi, menyimpan, dan memperbarui data. Modul ini mendukung pengelolaan data peminjam dan berkontribusi pada efisiensi operasional aplikasi.



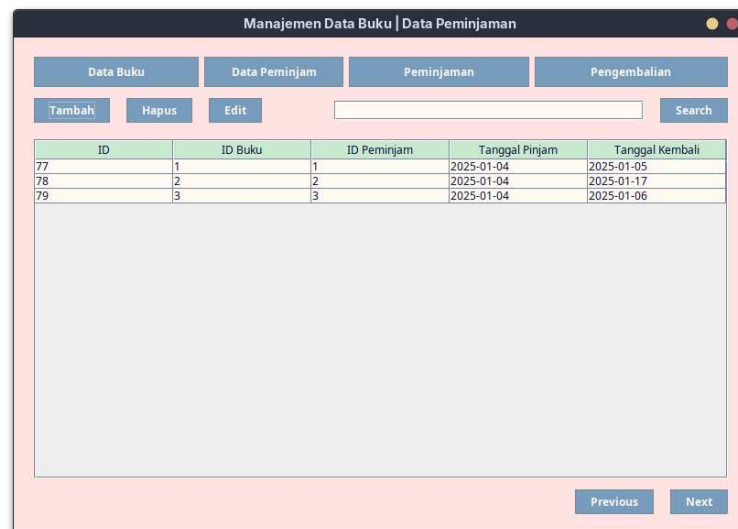
Gambar 5. Panel Data Peminjam

Table 3. Pengujian CRUD dan Pencarian pada Panel Data Buku

Pengujian	Keterangan	Status
Penambahan Data	Semua kolom, yaitu Nama, Alamat, dan Nomor Telepon, harus diisi dengan lengkap dan sesuai format. Nomor Telepon harus berupa angka dengan panjang antara 11 hingga 15 digit.	Valid
Pembaruan Data	Kolom Nama, Alamat, dan Nomor Telepon harus diisi dengan benar, dengan nomor telepon berupa angka dan panjang antara 11 hingga 15 digit.	Valid
Penghapusan Data	Data peminjam yang dihapus akan dihapus secara permanen dari sistem dan tidak dapat dipulihkan.	Valid
Pencarian Data	Pencarian data peminjam dilakukan melalui kolom pencarian berdasarkan Nama Peminjam atau Nomor Telepon yang dimasukkan.	Valid

4. Modul Panel Data Peminjaman

Gambar 6 menunjukkan modul panel data peminjaman yang mendukung pencatatan, pembaruan, dan penghapusan data peminjaman buku. Data seperti ID Peminjam, ID Buku, dan Tanggal Kembali divalidasi dengan ketat. Hasil pengujian fungsi CRUD dan pencarian yang dirangkum dalam Tabel 4 menunjukkan sistem berfungsi optimal. Pencarian data dilakukan berdasarkan ID dan tanggal, dengan menjaga integritas data. Modul ini meningkatkan efisiensi pengelolaan peminjaman dan kualitas data perpustakaan.



Gambar 6. Panel Data Peminjam

Table 4. Pengujian CRUD dan Pencarian pada Panel Data Peminjaman

Pengujian	Keterangan	Status
Penambahan Data	Semua kolom, termasuk ID Buku, ID Peminjam, Tanggal Pinjam, dan Tanggal Kembali, harus valid dengan format yang sesuai.	Valid
Pembaruan Data	Semua kolom harus diisi dengan benar, termasuk ID Buku, ID Peminjam, Tanggal Pinjam, dan Tanggal Kembali, dengan format yang benar.	Valid
Penghapusan Data	Data yang dihapus hanya disembunyikan, tidak dihapus fisik, dan terkait buku serta peminjam akan kembali muncul.	Valid
Pencarian Data	Pencarian menggunakan kata kunci ID Buku, ID Peminjam, Tanggal Pinjam, atau Tanggal Kembali.	Valid

5. Modul Panel Data Pengembalian

Gambar 7 menunjukkan bahwa modul panel data pengembalian dirancang untuk mengelola proses pengembalian buku dengan akurasi. Pengguna dapat menambahkan data pengembalian dengan memasukkan ID Peminjaman, yang otomatis memuat informasi terkait. Sistem menghitung denda keterlambatan sebesar 1.500 per hari berdasarkan perbedaan antara Tanggal Kembali aktual dan yang ditentukan. Modul ini menggunakan metode soft delete, menyembunyikan data yang dihapus tanpa menghapusnya secara permanen. Fitur pencarian mendukung parameter seperti ID Pengembalian dan Tanggal Dikembalikan, dan pengujian menunjukkan akurasi tinggi dalam menampilkan data, bahkan dalam jumlah besar. Modul ini meningkatkan efisiensi operasional dan mempermudah akses informasi pengembalian buku

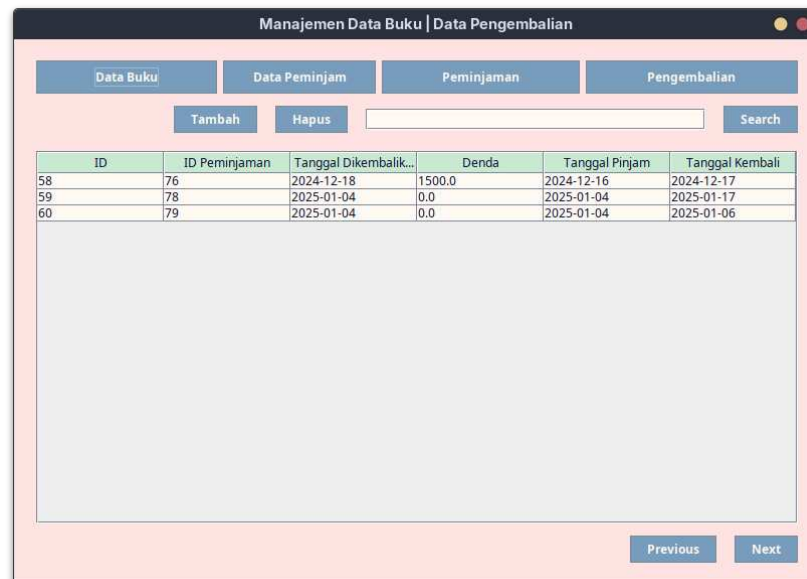
**Gambar 7.** Panel Data Pengembalian

Table 5. Pengujian CRUD dan Pencarian pada Panel Data Pengembalian

Pengujian	Keterangan	Status
Penambahan Data	Menginput ID Peminjaman untuk menampilkan data terkait dan menghitung denda berdasarkan keterlambatan.	Valid
Penghapusan Data	Data disembunyikan, tidak dihapus secara fisik, untuk menjaga integritas.	Valid
Pencarian Data	Pencarian berdasarkan ID, Tanggal, Denda, dan informasi terkait pengembalian.	Valid

Pengujian dalam Tabel 5 menunjukkan bahwa semua fungsi utama, termasuk CRUD dan pencarian, berjalan dengan baik. Penghitungan denda sesuai dengan skema, dan penyembunyian data menjaga integritas informasi. Sistem terintegrasi dengan baik, menjaga konsistensi data dan operasional. Keandalan dalam mengelola data pengembalian dan denda meningkatkan efisiensi, kualitas layanan, serta kepuasan pengguna, mempermudah manajemen perpustakaan dalam mengelola peminjaman dan pengembalian buku.

Hasil pengujian menunjukkan bahwa seluruh proses pengembalian, mulai dari penambahan data, penghitungan denda, hingga penyembunyian data, berjalan dengan baik. Pencarian data pengembalian juga mudah dilakukan. Sistem dapat menangani proses secara efisien, memberikan respons cepat, dan menjaga integritas serta keamanan data. Aplikasi ini terbukti andal, siap digunakan sebagai solusi digital yang praktis untuk pengelolaan data buku, serta memberikan kemudahan dan efisiensi bagi pengguna. Sistem ini cocok untuk implementasi di sektor yang membutuhkan pengelolaan data yang terstruktur dan aman.

Hasil dan Diskusi

Hasil pengujian menunjukkan bahwa seluruh proses pengembalian buku, seperti penambahan data baru, penghitungan denda berbasis waktu keterlambatan, dan penyembunyian data pengembalian lama, telah berjalan lancar sesuai dengan skenario yang dirancang. Sistem menunjukkan kinerja efisien dengan waktu respons cepat, menjaga integritas data melalui validasi yang kuat, dan memastikan keamanan data menggunakan mekanisme otentikasi pengguna. Aplikasi terbukti andal serta layak diimplementasikan sebagai solusi digital yang memberikan kemudahan kepada pengguna, khususnya dalam pengelolaan data buku yang terstruktur dan aman.

Pengembangan aplikasi difokuskan pada peningkatan fungsionalitas dan kualitas pengalaman pengguna (User Experience/UX). Fitur ekspor data menjadi prioritas utama dengan rencana penambahan tombol ekspor yang menggunakan pustaka Java, seperti Apache POI untuk menghasilkan file berformat CSV (Apache POITM, n.d.) dan iText untuk membuat laporan dalam format PDF (iText PDF, n.d.). Fitur ini dirancang untuk mempermudah pengguna dalam menyimpan atau berbagi data dengan lebih fleksibel. Penyempurnaan antarmuka pengguna (UI) juga menjadi fokus, dengan penekanan pada desain modern yang mengedepankan konsistensi elemen visual, tata letak intuitif, dan navigasi yang mudah digunakan. Langkah ini diharapkan dapat mengurangi kebingungan pengguna sekaligus meningkatkan daya tarik visual aplikasi (Wei, 2024)

Fitur statistik penggunaan juga direncanakan untuk memberikan informasi analitik yang mencakup jumlah buku yang dipinjam dalam periode tertentu, daftar buku yang paling sering dipinjam, dan total akumulasi denda (Durrant, 2022). Fitur ini bertujuan membantu pengguna memantau aktivitas operasional aplikasi secara menyeluruh. Aplikasi juga dirancang untuk mendukung kompatibilitas lintas platform, terutama pada perangkat seluler berbasis Android atau iOS, guna meningkatkan aksesibilitas pengguna. Sebagai upaya mempermudah pemeliharaan dan pengembangan di masa depan, dokumentasi teknis yang komprehensif akan disiapkan, mencakup penjelasan kode, struktur database, dan panduan pengembangan lebih lanjut (Eke & Salihu, 2021). Dengan pengembangan ini, aplikasi diharapkan menjadi solusi digital yang lebih fleksibel, fungsional, dan dapat diandalkan untuk kebutuhan pengelolaan data buku di berbagai sektor.

Kesimpulan

Aplikasi manajemen data buku yang telah dikembangkan berhasil memenuhi kebutuhan pengelolaan data secara efektif melalui penerapan fitur CRUD (*Create, Read, Update, Delete*), validasi input yang ketat, dan kemampuan pencarian data yang cepat dan efisien. Sistem pencatatan peminjaman dan pengembalian buku beroperasi secara stabil, dilengkapi dengan mekanisme perhitungan denda yang akurat sesuai dengan aturan yang ditetapkan. Integrasi dengan database MySQL memberikan dukungan yang signifikan terhadap efisiensi pengolahan data, memastikan kinerja sistem yang optimal. Hasil evaluasi menunjukkan bahwa aplikasi ini memiliki potensi besar untuk dikembangkan lebih lanjut, khususnya melalui penambahan fitur ekspor data ke berbagai format dan penyediaan laporan statistik yang komprehensif. Langkah pengembangan berikutnya juga dirancang untuk meningkatkan pengalaman pengguna, termasuk perbaikan antarmuka pengguna (UI) agar lebih intuitif dan menarik, serta pengembangan versi aplikasi yang kompatibel dengan perangkat seluler. Dengan inovasi tersebut, aplikasi ini diharapkan menjadi solusi yang semakin andal dan relevan untuk kebutuhan pengelolaan data buku di masa mendatang.

Daftar Pustaka

- Anwar, M. R., Apriani, D., & Adianita, I. R. (2021). Hash Algorithm In Verification Of Certificate Data Integrity And Security. *Aptisi Transactions on Technopreneurship (ATT)*, 3(2), 65–72. <https://doi.org/10.34306/att.v3i2.212>
- Apache POITM. (n.d.). *The Java API for Microsoft Documents*. Apache. Retrieved January 1, 2025, from <https://poi.apache.org/>
- Blagojche, N., Dimitrovska, G., & Joshevska, E. (2023). The importance of databases in web programming. *International Journal of Advanced Natural Sciences and Engineering Researches*, 7(4), 319–322. <https://doi.org/10.59287/ijanser.724>
- Durrant, S. (2022). Using an Evaluation Grid to Holistically Assess Library Databases. *Collection Management*, 47(1), 20–36. <https://doi.org/10.1080/01462679.2021.1958723>
- Eke, N. O., & Salihu, I. A. (2021). Design and Implementation of a Mobile Library Management System for Improving Service Delivery. *Path of Science*, 7(4), 3001. <https://doi.org/10.22178/pos.69-7>
- Franck, L. D., Ginja, G. A., Carmo, J. P., Afonso, J. A., & Luppe, M. (2023). Custom ASIC Design for SHA-256 Using Open-Source Tools. *Computers*, 13(1), 9. <https://doi.org/10.3390/computers13010009>
- Győrödi, C. A., Dumșe-Burescu, D. V., Győrödi, R. Ș., Zmaranda, D. R., Bandici, L., & Popescu, D. E. (2021). Performance Impact of Optimization Methods on MySQL Document-Based and Relational Databases. *Applied Sciences*, 11(15), 6794. <https://doi.org/10.3390/app11156794>
- iText PDF. (n.d.). *iText Suite Java*. IText PDF. Retrieved January 1, 2025, from <https://itextpdf.com/itext-suite-java>
- Jiang, L., & Naumann, F. (2020). Holistic primary key and foreign key detection. *Journal of Intelligent Information Systems*, 54(3), 439–461. <https://doi.org/10.1007/s10844-019-00562-z>
- Kaczorowski, C. (2023). A comparative analysis of contemporary integrated java environments. *Journal of Computer Sciences Institute*, 26, 42–47. <https://doi.org/10.35784/jcsi.3077>

- Litvinov, A. A. (2020). On Business Logic Layer Design and Architecture. *System Technologies*, 1(126), 86–95. <https://doi.org/10.34185/1562-9945-1-126-2020-09>
- Onu, F. U., Oliver, I. C., Hyginus, M. C., Yinka, A. I., O, N. J., & Sunday, Y. (2024). Demystifying The Object-Oriented Features Of Popular Object-Oriented Programming Languages. *IOSR Journal of Computer Engineering*, 26(5), 26–32. <https://doi.org/10.9790/0661-2605012632>
- Pang, L. (2022). Library Management System Based on Data Mining and Clustering Algorithm. *Wireless Communications and Mobile Computing*, 2022, 1–6. <https://doi.org/10.1155/2022/1398681>
- Sharma, S., Mishra, S., Gupta, S., & Kumar, S. (2022). Library Management System. *International Journal for Research in Applied Science and Engineering Technology*, 10(5), 889–893. <https://doi.org/10.22214/ijraset.2022.42375>
- Toedter.com. (n.d.). *Java Swing calendar date picker*. Retrieved January 1, 2025, from <https://toedter.com/jcalendar/>
- Wei, Z. (2024). The design of library database management system based on MySQL. *Applied and Computational Engineering*, 38(1), 41–50. <https://doi.org/10.54254/2755-2721/38/20230527>
- Żuchnik, M., & Kopniak, P. (2021). Comparative analysis of connection performance with databases via JDBC interface and ORM programming frameworks. *Journal of Computer Sciences Institute*, 21, 309–315. <https://doi.org/10.35784/jcsi.2729>