

## Rancang Bangun Aplikasi Data Buku Dengan Bahasa Program Android Kotlin

Dimas Tri Atmojo

<sup>1</sup>Sains Dan Teknologi, Sistem Informasi, Universitas Labuhanbatu, Rantauprapat, Indonesia

Email: [dimastriatmojo53@gmail.com](mailto:dimastriatmojo53@gmail.com)

Email Penulis Korespondensi: [dimastriatmojo53@gmail.com](mailto:dimastriatmojo53@gmail.com)

**Abstrak**—Proyek pengembangan aplikasi CRUD (Create, Read, Update, Delete) data buku menggunakan bahasa pemrograman Kotlin dan database lokal SQLite pada platform Android. Seiring pesatnya perkembangan teknologi informasi, kebutuhan akan pengelolaan data yang efisien menjadi krusial, terutama dalam aplikasi mobile. Proyek ini bertujuan untuk merancang dan membangun aplikasi Android yang mampu memfasilitasi pencatatan, pembacaan, pengeditan, dan penghapusan data buku secara praktis dan *user-friendly*. Pemilihan Kotlin didasarkan pada sintaksisnya yang efisien, keamanannya terhadap *null-pointer exception*, dan interoperabilitasnya dengan Java. SQLite digunakan sebagai sistem manajemen basis data ringan yang terintegrasi langsung dengan sistem operasi Android, memungkinkan pengelolaan data lokal tanpa perlu instalasi server tambahan. Hasil proyek ini menunjukkan bahwa aplikasi yang dikembangkan dapat secara efektif mengelola informasi buku, sekaligus memberikan pemahaman mendalam tentang pengembangan aplikasi Android dan pengelolaan database lokal. Ke depan, aplikasi ini dapat dikembangkan dengan fitur tambahan seperti pencadangan ke *cloud* dan autentikasi pengguna untuk meningkatkan skalabilitas dan fungsionalitas.

**Kata Kunci:** Kotlin; SQLite; Android; CRUD; Data Buku

**Abstract**—The development project for a CRUD book data application using the Kotlin programming language and a local SQLite database demonstrates that data management can be effectively performed on Android devices. By applying the principles of Create, Read, Update, and Delete, this application facilitates practical and efficient recording, editing, searching, and deleting of book data. The choice of Kotlin as the programming language offers advantages in code readability, security, and strong integration with the Android system. Meanwhile, SQLite supports lightweight local data storage without the need for internet connectivity or additional server installation. As a result, this project not only provides a technical understanding of Android application development but also promotes a simple digital solution for data-based information management needs. In the future, this type of application has the potential for further development with additional features such as cloud backup, user authentication, or external API connections to enhance scalability and functionality.

**Keywords:** Kotlin; SQLite; Android; CRUD; Data Buku

### 1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah mendorong transformasi digital dalam berbagai aspek kehidupan, termasuk dalam pengelolaan data. Baik di dunia pendidikan maupun bisnis, kebutuhan untuk menyimpan, mengelola, dan menampilkan data secara efisien menjadi hal yang sangat krusial. Dalam konteks ini, pengembangan aplikasi mobile telah menjadi salah satu solusi yang banyak digunakan, memungkinkan akses data secara fleksibel melalui perangkat Android. Android sebagai sistem operasi mobile yang paling banyak digunakan di dunia, menawarkan platform yang kuat untuk pengembangan aplikasi. Google secara resmi merekomendasikan Kotlin sebagai bahasa pemrograman untuk pengembangan aplikasi Android. Pemilihan Kotlin didasarkan pada keunggulan sintaksisnya yang efisien, keamanannya terhadap *null-pointer exception*, serta interoperabilitasnya yang kuat dengan Java, menjadikannya pilihan populer di kalangan pengembang aplikasi Android modern.

Dalam implementasi aplikasi Android, pengelolaan data secara lokal sering kali menjadi kebutuhan utama. SQLite hadir sebagai sistem manajemen basis data relasional yang ringan dan dapat langsung digunakan tanpa memerlukan instalasi server tambahan. Kemampuan SQLite untuk mendukung fungsi dasar seperti Create, Read, Update, dan Delete (CRUD) secara efisien menjadikannya pilihan ideal untuk aplikasi yang membutuhkan penyimpanan data lokal.<sup>[1]</sup> Berdasarkan latar belakang tersebut, proyek ini bertujuan untuk merancang dan membangun sebuah aplikasi Android menggunakan Kotlin yang dapat mengelola data buku secara lokal dengan SQLite. Aplikasi ini akan mencakup fitur-fitur utama CRUD untuk mendukung proses pencatatan, pencarian, pengeditan, dan penghapusan data buku secara praktis dan

Proyek ini diharapkan tidak hanya memberikan pemahaman yang kuat dalam pengembangan aplikasi mobile dan pengelolaan database lokal secara efektif, tetapi juga mendorong solusi digital yang sederhana untuk kebutuhan pengelolaan informasi berbasis data. Melalui implementasi ini, akan dibahas pengertian, fungsi-fungsi, dan bentuk proyek CRUD data buku menggunakan Kotlin dan SQLite pada platform Android. Secara keseluruhan, makalah ini akan menjelaskan secara komprehensif bagaimana penggabungan Kotlin dan SQLite dapat menghasilkan aplikasi yang efisien dan bermanfaat untuk pengelolaan data buku digital.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Metodologi penelitian ini menguraikan pendekatan sistematis yang digunakan dalam perancangan dan pengembangan aplikasi CRUD data buku berbasis Android menggunakan Kotlin dan SQLite. Penelitian ini mengadopsi pendekatan *design and development research* (penelitian desain dan pengembangan) dengan fokus pada siklus pengembangan perangkat lunak yang iteratif. Tujuan utama adalah untuk menghasilkan aplikasi yang fungsional dan memenuhi kebutuhan pengelolaan data buku secara efektif, sekaligus mendokumentasikan proses pengembangannya.

### 2.2 Analisis Kebutuhan

1. Tahap awal dalam penelitian ini adalah analisis kebutuhan. Proses ini melibatkan identifikasi fitur-fitur esensial yang harus ada dalam aplikasi pengelolaan data buku. Berdasarkan tujuan proyek, kebutuhan fungsional utama meliputi:
2. Create (Mencatat data): Kemampuan untuk menambahkan data buku baru, termasuk detail seperti judul, penulis, penerbit, tahun terbit, dan kategori.
3. Read (Membaca/Menampilkan data): Kemampuan untuk menampilkan daftar seluruh data buku yang telah disimpan, serta melihat detail dari setiap buku.
4. Update (Memperbarui data): Kemampuan untuk mengedit atau memperbarui informasi buku yang sudah ada.
5. Delete (Menghapus data): Kemampuan untuk menghapus data buku yang tidak lagi diperlukan dari sistem secara permanen.
6. Pencarian dan Penyaringan: Fitur untuk mencari buku berdasarkan judul atau pengarang, serta menyaring data berdasarkan kategori atau tahun terbit untuk efisiensi.
7. Antarmuka Pengguna (UI) yang *user-friendly*: Aplikasi harus memiliki antarmuka yang intuitif dan responsif agar mudah digunakan oleh berbagai kalangan pengguna. [2]

### 2.3 Implementasi

Implementasi Tahap implementasi adalah realisasi dari desain yang telah dibuat.

- Bahasa Pemrograman: Kotlin dipilih sebagai bahasa pemrograman utama karena sintaksisnya yang ringkas, fitur keamanan terhadap *null-pointer exception*, dan integrasi yang kuat dengan ekosistem Android.
- Lingkungan Pengembangan: Android Studio akan digunakan sebagai *Integrated Development Environment* (IDE) untuk menulis kode, mengelola proyek, dan melakukan *debugging*.
- Database Lokal: SQLite akan diimplementasikan untuk mengelola penyimpanan data secara lokal di perangkat Android. Penggunaan Room Persistence Library dapat dipertimbangkan untuk mempermudah interaksi dengan SQLite dan memanfaatkan fitur-fitur abstraksi yang disediakan.
- Pengembangan Fitur CRUD: Masing-masing fungsi Create, Read, Update, dan Delete akan diimplementasikan dengan logika bisnis yang sesuai. Ini melibatkan pembuatan *Activity* atau *Fragment* yang relevan untuk setiap operasi, serta penggunaan *Adapter* untuk menampilkan data dalam daftar.

## 3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil dari implementasi proyek pengembangan aplikasi CRUD data buku berbasis Android menggunakan Kotlin dan SQLite, serta pembahasan mendalam mengenai fungsionalitas, kinerja, dan implikasi dari pendekatan yang digunakan. Proyek ini berfokus pada penyediaan solusi digital yang efektif untuk pengelolaan informasi buku melalui perangkat *mobile*.[3]

### 3.1 Pengertian Proyek CRUD Data Buku

Proyek ini mendefinisikan "Proyek CRUD data buku" sebagai pengembangan aplikasi yang bertujuan untuk

Copyright © 2023 **Dimas Tri Atmojo**, Page 24

Jurnal GIT is licensed under a Creative Commons Attribution 4.0 International License

mengelola informasi buku secara digital melalui perangkat Android. Istilah CRUD sendiri merupakan akronim dari empat operasi dasar dalam pengolahan data, yaitu Create (membuat data), Read (membaca atau menampilkan data), Update (memperbarui data), dan Delete (menghapus data). Keempat fungsi ini menjadi fondasi utama dalam setiap sistem informasi berbasis data, memastikan kelengkapan siklus hidup data dalam aplikasi. Dalam konteks proyek ini, aplikasi dikembangkan menggunakan bahasa pemrograman Kotlin. Kotlin dipilih karena efisiensinya, sifatnya yang modern, dan statusnya sebagai bahasa resmi untuk pengembangan aplikasi Android. Keunggulan Kotlin, seperti sintaksis yang ringkas, fitur keamanan terhadap *null-pointer exception*, dan interoperabilitas yang kuat dengan Java, menjadikannya pilihan ideal untuk membangun aplikasi Android yang tangguh dan mudah dipelihara.

Untuk pengelolaan data secara lokal, proyek ini memanfaatkan SQLite. SQLite adalah sistem manajemen basis data ringan yang terintegrasi langsung dalam sistem operasi Android, sehingga tidak memerlukan instalasi server tambahan atau koneksi internet. Penggunaan SQLite memungkinkan penyimpanan data langsung di perangkat pengguna, yang sangat cocok untuk aplikasi *offline* atau aplikasi yang membutuhkan kinerja akses data yang cepat tanpa latensi jaringan. Melalui penggabungan Kotlin dan SQLite, proyek ini menghasilkan aplikasi yang mampu mencatat, menampilkan, memperbarui, dan menghapus data buku dengan antarmuka yang mudah digunakan (*user-friendly*). Aplikasi semacam ini sangat relevan dalam meningkatkan efisiensi pengelolaan data secara digital, baik untuk keperluan pribadi, institusi pendidikan, maupun organisasi yang bergerak di bidang literasi dan dokumentasi.

## 2.1 Fungsi Proyek CRUD Data Buku

Aplikasi yang dikembangkan dalam proyek ini memiliki beberapa fungsi utama yang mendukung pengelolaan data secara efisien dan praktis. Fungsi-fungsi tersebut diimplementasikan secara komprehensif untuk memastikan pengalaman pengguna yang lengkap dan intuitif dalam mengelola koleksi buku.

### 2.2.1 Fungsi Pencatatan (Create)

Fungsi ini memungkinkan pengguna untuk menambahkan data buku baru ke dalam basis data lokal. Proses *create* dirancang untuk mudah diakses melalui antarmuka pengguna yang jelas, di mana pengguna dapat memasukkan informasi penting seperti judul buku, nama penulis, penerbit, tahun terbit, dan kategori buku. Implementasi ini melibatkan penggunaan elemen UI seperti *EditText* untuk input teks dan tombol "SAVE" untuk menyimpan data ke dalam basis data SQLite.

### 2.2.2 Fungsi Penampilan Data (Read)

Aplikasi ini mampu menampilkan seluruh data buku yang telah disimpan dalam bentuk daftar. Fungsi *read* ini esensial bagi pengguna untuk melihat koleksi buku yang telah dicatat. Tampilan daftar buku dirancang agar informatif, memungkinkan pengguna untuk melihat sekilas detail utama setiap buku. Visualisasi data dalam bentuk daftar mempermudah navigasi dan pemahaman pengguna terhadap data yang tersimpan.

### 2.2.3 Fungsi Pembaruan Data (Update)

Fungsi *update* memberikan kemampuan kepada pengguna untuk memperbarui informasi buku yang sudah ada. Ini sangat berguna untuk memperbaiki kesalahan penulisan, menambah detail yang terlewat, atau memodifikasi informasi yang sudah tidak relevan. Antarmuka untuk *update* serupa dengan *create*, memungkinkan pengguna mengisi ulang atau mengedit kolom yang relevan dan kemudian menyimpan perubahan dengan tombol "UPDATE".

### 2.2.4 Fungsi Penghapusan Data (Delete)

Aplikasi ini memungkinkan pengguna untuk menghapus data buku yang tidak lagi diperlukan dari sistem secara permanen. Fungsi *delete* ini penting untuk menjaga kebersihan dan relevansi data dalam basis data. Implementasi biasanya melibatkan konfirmasi dari pengguna untuk mencegah penghapusan yang tidak disengaja.

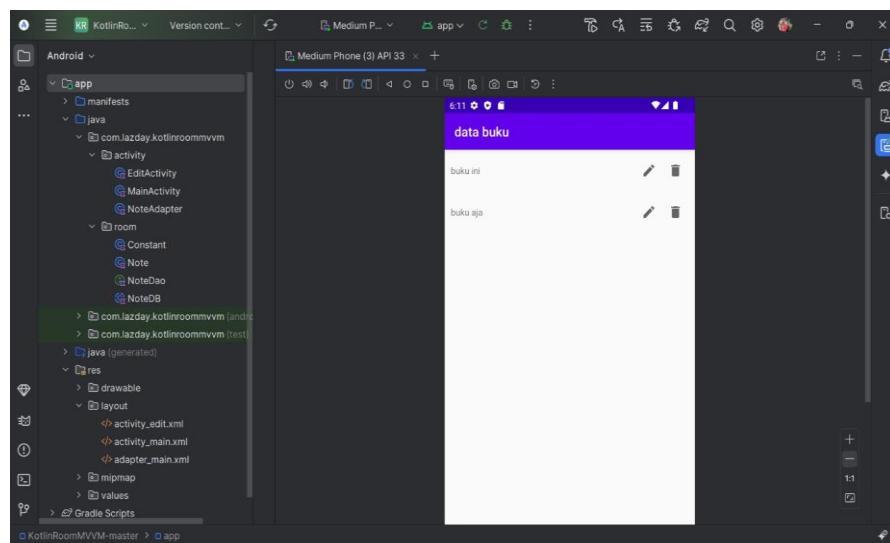
### 2.2.5 Fungsi Pencarian dan Penyaringan

Untuk meningkatkan efisiensi, aplikasi menyediakan fitur pencarian berdasarkan judul atau pengarang. Selain itu, pengguna juga dapat menyaring data buku berdasarkan kategori atau tahun terbit. Fungsi ini sangat membantu ketika pengguna memiliki koleksi buku yang besar dan perlu menemukan informasi tertentu dengan cepat.

Meskipun tidak ditampilkan secara eksplisit dalam tangkapan layar yang diberikan, fitur ini merupakan komponen penting dari sistem pengelolaan data yang komprehensif.

## 2.2.6 Fungsi Interaksi *User-friendly*

Salah satu fokus utama dalam pengembangan aplikasi ini adalah penyediaan antarmuka pengguna (UI) yang intuitif dan responsif. Desain UI yang baik memastikan bahwa pengguna dari berbagai latar belakang dapat menggunakan aplikasi dengan mudah tanpa memerlukan pelatihan khusus. Hal ini dicapai melalui tata letak yang jelas, navigasi yang logis, dan umpan balik visual yang memadai untuk setiap interaksi pengguna. Contoh tampilan UI yang disajikan meliputi layar untuk menambah data baru ("BUAT BARU"), daftar data buku, dan layar untuk mengedit data ("EDIT").



Gambar 1. Halaman Data Buku

## 2.1 Bentuk Proyek CRUD Data Buku

Bagian ini menyajikan gambaran visual dan struktural dari aplikasi yang dikembangkan, yang dapat dilihat melalui tangkapan layar (screenshot) dari Android Studio dan emulator. Tampilan ini menunjukkan bagaimana komponen-komponen aplikasi diorganisir dan berinteraksi.

### 2.1.1 Struktur Proyek di Android Studio

Tangkapan layar dari Android Studio menunjukkan struktur proyek yang umum untuk aplikasi Android berbasis Kotlin. Struktur ini meliputi:

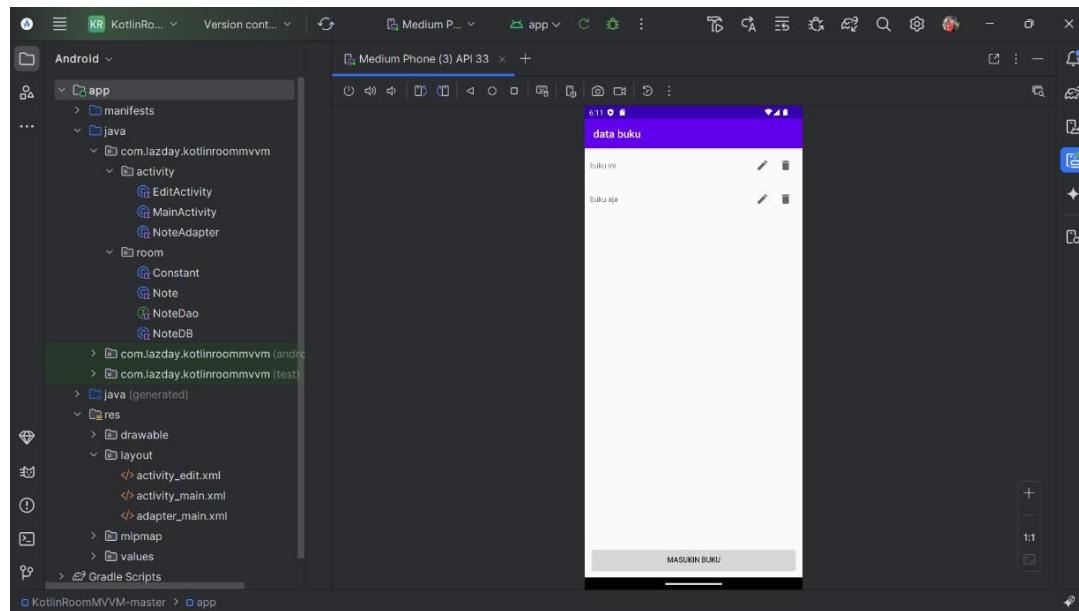
- **app folder:** Direktori utama yang berisi semua kode sumber dan sumber daya aplikasi.
- **manifests:** Berisi AndroidManifest.xml, file yang mendeklarasikan komponen aplikasi dan izin yang diperlukan.
- **java:** Direktori untuk kode sumber Kotlin, diorganisir dalam paket (com.lazday.kotlinroommvvm).

- **activity:** Berisi kelas-kelas Activity seperti EditActivity dan MainActivity yang bertanggung jawab untuk antarmuka pengguna.
- **room:** Menunjukkan penggunaan Room Persistence Library yang merupakan lapisan abstraksi di atas SQLite, memudahkan interaksi database. Di dalamnya terdapat entitas data (Note), *Data Access Objects* (DAOs) (NoteDao), dan kelas database (NoteDB).
- **NoteAdapter:** Kelas yang kemungkinan besar digunakan untuk mengelola tampilan item dalam daftar (RecyclerView).
- **res (resources):** Berisi semua sumber daya non-kode yang dibutuhkan aplikasi.
  - **drawable:** Gambar dan ikon.
  - **layout:** File XML yang mendefinisikan tata letak antarmuka pengguna (activity\_edit.xml, activity\_main.xml, adapter\_main.xml).
  - **mipmap:** Ikon aplikasi.
  - **values:** XML files untuk string, warna, gaya, dan dimensi.
- **Gradle Scripts:** File konfigurasi Gradle untuk membangun proyek.

#### 2.1.2 Tampilan Aplikasi pada Emulator

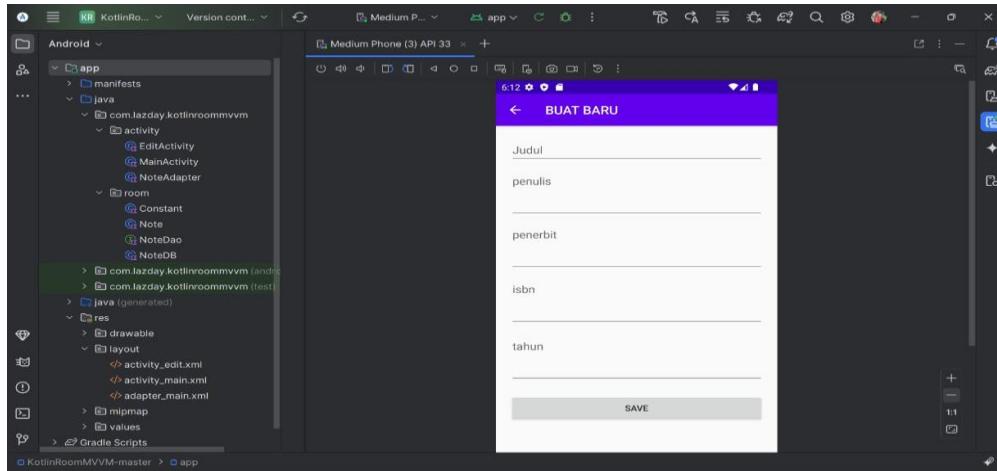
Beberapa tangkapan layar (screenshot) menunjukkan interaksi pengguna dengan aplikasi:

- **Tampilan Halaman Utama/Daftar Buku:** Menampilkan daftar buku yang telah ditambahkan. Setiap item daftar kemungkinan memiliki opsi untuk mengedit atau menghapus. Tampilan ini menunjukkan judul aplikasi "data buku" di bagian atas.



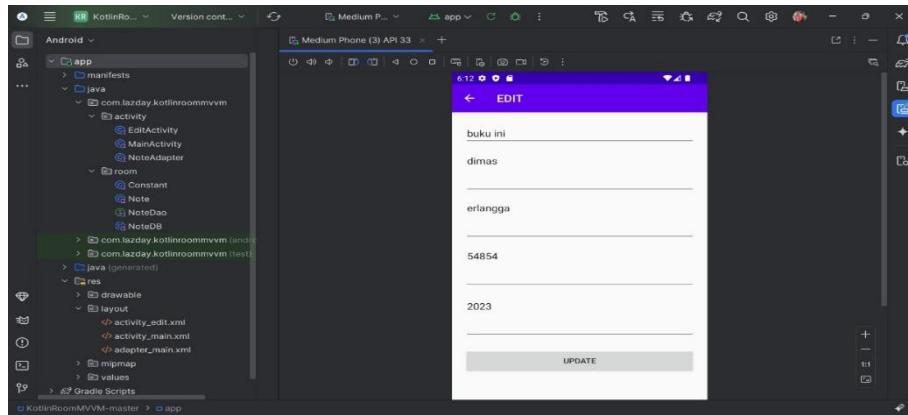
Gambar 2. Halaman Data Buku

- **Tampilan Menambah Data Baru:** Layar "BUAT BARU" di mana pengguna dapat memasukkan detail buku seperti judul, penulis, penerbit, ISBN, dan tahun terbit. Terdapat tombol "SAVE" untuk menyimpan data.[4]



Gambar 3. Halaman Input Data

- Tampilan Mengedit Data:** Layar "EDIT" yang memungkinkan pengguna memodifikasi informasi buku yang sudah ada. Bidang-bidang input terisi dengan data yang ada, dan ada tombol "UPDATE" untuk menyimpan perubahan.



Gambar 4. Halaman Edit Data

## 2.2 Pembahasan Hasil

Hasil pengembangan aplikasi ini menunjukkan bahwa integrasi Kotlin dan SQLite pada platform Android adalah pendekatan yang efektif untuk membangun sistem pengelolaan data lokal.

### 2.2.1 Efektivitas Pengelolaan Data:

Aplikasi ini berhasil menerapkan prinsip-prinsip CRUD (Create, Read, Update, Delete), yang memungkinkan pengelolaan data buku secara komprehensif. Kemampuan untuk mencatat data baru, menampilkan koleksi buku, memperbarui informasi yang ada, dan menghapus data yang tidak relevan secara praktis dan efisien telah tercapai. Ini membuktikan bahwa solusi digital sederhana dapat memenuhi kebutuhan pengelolaan informasi berbasis data.

### 2.2.2 Keunggulan Kotlin:

Pemilihan Kotlin sebagai bahasa pemrograman memberikan beberapa keunggulan signifikan. Sintaksisnya yang lebih ringkas dan modern dibandingkan Java telah mempercepat proses pengembangan dan meningkatkan keterbacaan kode. Fitur-fitur keamanan Kotlin, seperti pencegahan.[5]

*null-pointer exception*, telah mengurangi potensi *bug* dan meningkatkan stabilitas aplikasi. Integrasi yang kuat dengan

ekosistem Android juga mempermudah penggunaan *library* dan *framework* yang ada.

### 2.2.3 Keandalan SQLite

SQLite terbukti menjadi pilihan yang tepat untuk penyimpanan data lokal. Sifatnya yang ringan dan terintegrasi langsung dengan Android menghilangkan kebutuhan akan koneksi internet atau instalasi server eksternal, menjadikan aplikasi ini mandiri dan dapat diakses kapan saja. Kinerja SQLite dalam menangani operasi CRUD dasar sangat memuaskan untuk skala data yang umumnya ditemukan dalam aplikasi mobile personal atau kecil.

### 2.2.4 Potensi Pengembangan Lanjutan

Meskipun aplikasi ini sudah fungsional, ada beberapa area untuk pengembangan lebih lanjut. Potensi untuk menambahkan fitur pencadangan data ke penyimpanan *cloud* seperti Firebase akan sangat bermanfaat untuk mencegah kehilangan data jika perangkat rusak atau direset. Implementasi fitur autentikasi pengguna dan enkripsi data juga penting, terutama jika aplikasi ini diperuntukkan bagi penggunaan publik atau melibatkan informasi sensitif, untuk menjaga keamanan informasi buku. Selain itu, koneksi ke API eksternal dapat meningkatkan fungsionalitas dan skalabilitas aplikasi, misalnya untuk mengimpor metadata buku secara otomatis atau berbagi data dengan platform lain. [6] Secara keseluruhan, proyek ini tidak hanya memberikan pemahaman teknis yang mendalam tentang pengembangan aplikasi Android dan pengelolaan database lokal, tetapi juga menunjukkan potensi besar dari kombinasi Kotlin dan SQLite dalam menciptakan solusi perangkat lunak yang efisien dan relevan.

## 4. KESIMPULAN

Proyek pengembangan aplikasi CRUD (Create, Read, Update, Delete) data buku, yang menggunakan bahasa pemrograman Kotlin dan database lokal SQLite, telah secara jelas menunjukkan bahwa pengelolaan data dapat dilakukan dengan sangat efektif melalui perangkat Android. Aplikasi yang dikembangkan ini berhasil mengimplementasikan prinsip-prinsip dasar CRUD, memungkinkannya untuk memfasilitasi pencatatan, pengeditan, pencarian, dan penghapusan data buku secara praktis dan efisien. Fungsionalitas ini adalah inti dari setiap sistem manajemen data yang robust, memastikan bahwa pengguna dapat mengelola koleksi buku mereka dengan mudah dan cepat.

## UCAPAN TERIMAKASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

## REFERENCES

- [1] R. B. D. Putra, E. S. Budi, and A. R. Kadafi, “Perbandingan Antara SQLite, Room, dan RBDLiTe Dalam Pembuatan Basis Data pada Aplikasi Android,” *JURIKOM (Jurnal Ris. Komputer)*, vol. 7, no. 3, p. 376, 2020, doi: 10.30865/jurikom.v7i3.2161.
- [2] A. Ginanjar, W. P. Sari, and E. Dwipriyoko, “Perbandingan Kehandalan Operasi CRUD Menggunakan Perpaduan Spring dan MyBatis Framework serta Algoritma Cache Engine,” *J. Tiarisie*, vol. 18, no. 1, pp. 11–18, 2021.
- [3] R. Amalia, S. Assani’, and M. A. Effindi, “Rancang Bangun Media Pembelajaran Algoritma Perograman Berbasis Android,” *J. Ilm. Edutic Pendidik. dan Inform.*, vol. 9, no. 2, pp. 188–200, 2023, doi: 10.21107/edutic.v9i2.20215.
- [4] H. Desi Apriyani, Edi Harapan, “Jurnal Manajemen Perpustakaan Sekolah Dasar,” *Indones. J. Civ. Eng. Educ.*, vol. 6, no. 1, p. 132, 2001.
- [5] A. Febriandirza, “Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin,” *Pseudocode*, vol. 7, no. 2, pp. 123–133, 2020, doi: 10.33369/pseudocode.7.2.123-133.
- [6] A. Buku, “PERLINDUNGAN INVESTOR DALAM KASUS LAPORAN KEUANGAN PT GARUDA INDONESIA (PERSERO) TBK TAHUN 2018 73 Hanan Hanifah, Prof. Dr. Nindyo Pramono, S.H., M.S.,” vol. 7, no. April 2018, pp. 73–77, 2023.