



**SKRIPSI**

**IMPLEMENTASI ARSITEKTUR MVVM DAN STATE  
MANAGEMENT PROVIDER PADA APLIKASI  
AL-QUR'AN BERBASIS OFFLINE MODE**

**MUHAMMAD ALBERT NUR AGATHON**

**NPM 21081010020**

**DOSEN PEMBIMBING**

None

None

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAWA TIMUR  
FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI INFORMATIKA  
SURABAYA  
2025**

# **BAB 1**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Di era digital saat ini, perkembangan teknologi informasi dan komunikasi telah membawa perubahan signifikan dalam cara masyarakat mengakses dan berinteraksi dengan berbagai informasi, termasuk teks-teks suci seperti Al-Quran. Aplikasi Al-Quran digital menjadi salah satu solusi yang banyak dicari oleh umat Muslim untuk memudahkan akses terhadap bacaan Al-Quran kapan saja dan di mana saja. Namun, meskipun aplikasi ini menawarkan berbagai kemudahan, masih terdapat beberapa kendala yang sering dihadapi oleh pengguna, terutama terkait dengan ketergantungan pada koneksi internet dan performa aplikasi.

Salah satu tantangan utama dalam penggunaan aplikasi Al-Quran digital adalah ketergantungan pada koneksi internet. Banyak aplikasi saat ini memerlukan akses internet untuk menampilkan konten atau fitur tertentu, seperti tafsir, terjemahan, atau audio bacaan. Hal ini dapat menjadi masalah bagi pengguna yang tinggal di daerah dengan jaringan internet yang tidak stabil atau terbatas. Dalam situasi seperti ini, pengguna mungkin tidak dapat mengakses Al-Quran secara langsung, yang tentunya mengganggu pengalaman spiritual mereka[1]. Selain itu, performa aplikasi juga menjadi perhatian penting. Aplikasi yang lambat atau tidak responsif dapat menurunkan kepuasan pengguna dan membuat mereka enggan untuk menggunakan aplikasi tersebut secara rutin. Oleh karena itu, pengembangan aplikasi Al-Quran digital harus mempertimbangkan aspek performa agar dapat memberikan pengalaman pengguna yang optimal[4].

Dalam upaya mengatasi masalah-masalah tersebut, pemilihan arsitektur perangkat lunak yang tepat menjadi sangat penting. Model-View-ViewModel (MVVM) adalah salah satu arsitektur yang populer dalam pengembangan aplikasi modern, terutama untuk aplikasi berbasis mobile. MVVM memisahkan logika bisnis dari tampilan antarmuka pengguna, sehingga memudahkan pengembang dalam mengelola

kode dan meningkatkan kualitas aplikasi[2]. Dengan menggunakan MVVM, pengembang dapat menciptakan aplikasi yang lebih terstruktur dan mudah dipelihara. Pengelolaan state yang efisien juga menjadi kunci dalam pengembangan aplikasi yang responsif. Penggunaan state management Provider memungkinkan pengembang untuk mengelola state aplikasi dengan cara yang lebih terorganisir dan reaktif[5]. Dengan pendekatan ini, setiap perubahan pada data dapat langsung tercermin pada tampilan tanpa memerlukan banyak kode tambahan.

Implementasi mode offline pada aplikasi Al-Quran digital merupakan solusi inovatif untuk mengatasi ketergantungan pada koneksi internet. Dengan menyimpan data Al-Quran secara lokal di perangkat pengguna, aplikasi dapat diakses kapan saja tanpa perlu khawatir tentang ketersediaan jaringan. Hal ini tidak hanya meningkatkan kenyamanan pengguna tetapi juga memberikan aksesibilitas bagi mereka yang tinggal di daerah dengan konektivitas internet yang buruk[6].

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang sudah dijelaskan, maka dapat disimpulkan rumusan masalah pada penelitian ini adalah sebagai berikut.

1. Bagaimana mengimplementasikan arsitektur MVVM pada aplikasi Al-Quran berbasis offline mode?
2. Bagaimana mengintegrasikan state management Provider dengan arsitektur MVVM untuk pengelolaan data Al-Quran?
3. Bagaimana performa aplikasi Al-Quran setelah implementasi arsitektur MVVM dan state management Provider?

## **1.3. Tujuan**

Berdasarkan rumusan masalah diatas, maka tujuan dari penelitian ini adalah sebagai berikut.

1. Mengimplementasikan arsitektur MVVM pada aplikasi Al-Quran offline.
2. Mengintegrasikan state management Provider dengan arsitektur MVVM.
3. Menganalisis performa aplikasi setelah implementasi arsitektur dan state management yang dipilih.

#### **1.4. Manfaat**

Adapun manfaat dari penelitian ini adalah sebagai berikut.

1. Memberikan kontribusi dalam pengembangan aplikasi Al-Quran yang lebih terstruktur.
2. Menyediakan referensi implementasi arsitektur MVVM dan Provider untuk pengembang aplikasi serupa.
3. Menghasilkan aplikasi Al-Quran offline yang lebih efisien dan mudah dimaintain.

#### **1.5. Batasan Masalah**

Adapun batasan masalah dari pelaksanaan dan pembahasan penelitian ini adalah sebagai berikut.

1. Penelitian ini hanya berfokus pada pengembangan aplikasi Al-Quran berbasis offline mode.
2. Implementasi arsitektur yang digunakan adalah Model-View-ViewModel (MVVM).
3. State management yang diterapkan menggunakan library Provider.
4. Data Al-Quran yang digunakan disimpan secara lokal di perangkat pengguna.
5. Penelitian tidak mencakup fitur lain seperti pencarian lanjutan atau integrasi layanan eksternal.

6. Evaluasi performa aplikasi terbatas pada responsivitas dan efisiensi penggunaan memori.

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1. Penelitian Terdahulu**

Perkembangan aplikasi Al-Quran berbasis digital telah menarik perhatian banyak peneliti dalam beberapa tahun terakhir, dengan fokus pada solusi untuk meningkatkan aksesibilitas dan performa aplikasi. Sholeh, Ridhoni, dan Basuki (2022) mengembangkan aplikasi Al-Quran online dengan memanfaatkan REST API, yang memungkinkan integrasi data secara dinamis dari server ke aplikasi. Namun, pendekatan ini memiliki kelemahan, yaitu ketergantungan yang tinggi pada koneksi internet, yang dapat menjadi kendala bagi pengguna di wilayah dengan jaringan terbatas[7].

Penelitian lain yang relevan adalah yang dilakukan oleh Suryani dan Kurniawan (2022), yang menerapkan pola arsitektur MVVM pada aplikasi pengaduan masyarakat berbasis Android. Hasil penelitian ini menunjukkan bahwa pola MVVM mampu memisahkan logika bisnis dari antarmuka pengguna, sehingga meningkatkan keteraturan kode dan kemudahan pemeliharaan aplikasi. Konsep ini sangat relevan dalam pengembangan aplikasi Al-Quran digital untuk memastikan bahwa aplikasi tetap responsif dan mudah dikelola, meskipun implementasi MVVM pada aplikasi berbasis offline belum banyak dibahas dalam penelitian mereka.

Selain itu, Arif (2019) dalam penelitiannya mengenai implementasi pola arsitektur MVVM pada aplikasi SIAKAD berbasis mobile menunjukkan bahwa penggunaan MVVM dapat meningkatkan efisiensi pengelolaan data dan interaksi antar-komponen dalam aplikasi. Penelitian ini menjadi referensi penting dalam memahami bagaimana MVVM dapat diintegrasikan dengan state management Provider untuk mengelola data secara lebih efisien dalam aplikasi Al-Quran digital.

Berdasarkan penelitian-penelitian sebelumnya, penelitian ini bertujuan untuk mengembangkan aplikasi Al-Quran berbasis offline dengan mengintegrasikan arsitektur MVVM dan state management Provider, serta

mengatasi kelemahan yang ada, seperti ketergantungan pada koneksi internet dan performa aplikasi yang tidak optimal.

## **2.2. Landasan Teori**

### **2.2.1. MVVM (Model-View-ViewModel)**

Model-View-ViewModel (MVVM) adalah salah satu arsitektur yang digunakan dalam pengembangan aplikasi mobile untuk memisahkan logika bisnis (model) dari antarmuka pengguna (view). MVVM memungkinkan pengelolaan kode yang lebih terstruktur, modular, dan mudah diuji (Suryani & Kurniawan, 2022). ViewModel berfungsi sebagai penghubung antara model dan view, serta bertanggung jawab dalam menyediakan data untuk ditampilkan pada antarmuka pengguna.

### **2.2.2. State Management Provider**

Provider adalah salah satu teknik state management yang populer digunakan dalam pengembangan aplikasi menggunakan framework Flutter. Dengan Provider, pengembang dapat mengelola dan berbagi data antar-widget secara efisien. Nugroho (2020) menyebutkan bahwa Provider unggul dalam mengelola data secara dinamis, terutama pada aplikasi interaktif dengan fitur kompleks, seperti aplikasi Al-Qur'an digital. Hal ini memungkinkan perubahan data untuk langsung tercermin pada antarmuka tanpa menulis banyak kode tambahan.

### **2.2.3. Offline Mode Pada Aplikasi Mobile**

Mode offline merupakan fitur penting dalam aplikasi Al-Qur'an digital untuk memastikan data tetap dapat diakses tanpa koneksi internet. Implementasi mode offline menggunakan Realm sebagai database lokal memberikan keuntungan seperti performa tinggi, sinkronisasi data otomatis, dan kemampuan query yang efisien[9]. Realm mempermudah pengembang dalam mengelola data kompleks, seperti teks Al-Qur'an dan metadata yang terkait.

#### **2.2.4. Framework Flutter**

Flutter adalah framework open-source yang dikembangkan oleh Google untuk membangun aplikasi lintas platform (iOS, Android, Web) dengan satu basis kode. Flutter menggunakan bahasa pemrograman Dart dan memanfaatkan mesin rendering khusus untuk memberikan pengalaman pengguna yang konsisten di berbagai perangkat. Framework ini menawarkan performa tinggi dan efisiensi dalam pengembangan, serta mendukung fitur-fitur canggih seperti integrasi dengan state management, akses database lokal, dan mode offline[8].

#### **2.2.5. User Experience (UX)**

Dalam aplikasi Al-Qur'an berbasis mode offline, desain UX yang baik adalah kunci untuk memberikan pengalaman pengguna yang optimal. Penelitian menunjukkan bahwa desain antarmuka pengguna yang intuitif, responsif, dan mudah diakses sangat penting untuk aplikasi berbasis teks seperti aplikasi Al-Qur'an. Flutter mendukung desain antarmuka yang modern dan adaptif, sehingga meningkatkan pengalaman pengguna secara keseluruhan[10].

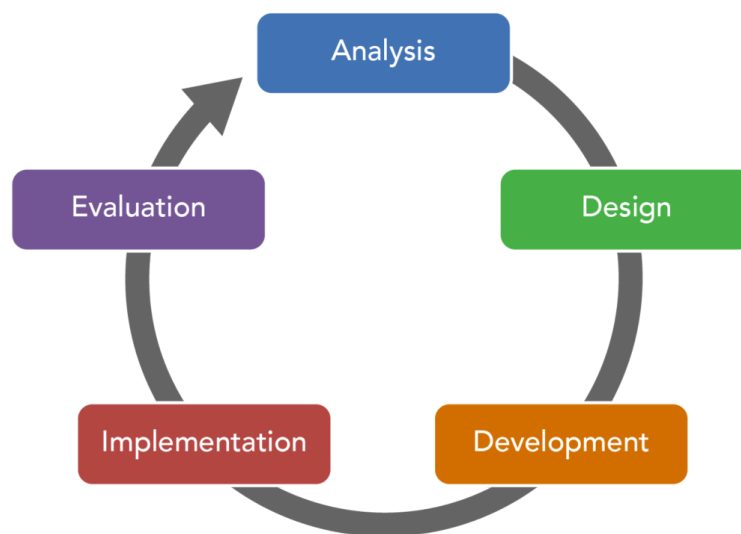


## BAB 3

### METODOLOGI

#### 3.1. Tahapan Penelitian

Penelitian ini dilaksanakan dengan mengikuti metodologi yang sistematis dan terstruktur untuk mencapai tujuan yang telah ditetapkan. Gambar 3.1.1 merupakan alur penelitian menggunakan metode ADDIE.



Gambar 3.1. Alur Metode Penelitian

Dalam alur penelitian ini, tahapan metode ADDIE dimulai dengan tahap Analysis (Analisis), di mana kebutuhan pengguna dan spesifikasi aplikasi diidentifikasi secara mendalam. Pada tahap ini, dilakukan analisis terhadap fitur utama yang diperlukan dalam aplikasi, seperti kemampuan membaca teks Al-Qur'an, terjemahan, tafsir, mode offline, dan sistem pencarian yang responsif. Selain itu, ditentukan pula teknologi yang akan digunakan, seperti Flutter sebagai framework pengembangan, arsitektur MVVM untuk memisahkan logika bisnis dan antarmuka, Provider sebagai state management, serta Realm untuk mendukung mode offline. Analisis ini dilakukan untuk memastikan aplikasi dapat memenuhi kebutuhan pengguna secara fungsional.

Tahap selanjutnya adalah Design (Perancangan), yang mencakup pembuatan *mockup*, *flowchart*, dan perancangan antarmuka pengguna (UI) yang intuitif. Selain itu, dilakukan perancangan arsitektur aplikasi menggunakan MVVM untuk memastikan modularitas dan keterpisahan yang baik antara komponen model, view, dan viewmodel. Pada tahap ini juga dirancang strategi implementasi mode offline, termasuk bagaimana data Al-Qur'an, terjemahan, dan tafsir disimpan secara lokal menggunakan Realm untuk mendukung aksesibilitas tanpa koneksi internet.

Tahap Development (Pengembangan) merupakan inti dari penelitian ini. Pada tahap ini, aplikasi mulai diimplementasikan berdasarkan desain yang telah dibuat. Framework Flutter digunakan untuk mengembangkan aplikasi dengan menggunakan arsitektur MVVM yang menghubungkan model, view, dan viewmodel. Provider diterapkan sebagai state management untuk mengelola aliran data secara efisien, sementara Realm digunakan sebagai database lokal untuk menyimpan data Al-Qur'an secara offline. Proses pengembangan juga melibatkan pengujian fungsional awal untuk memastikan setiap fitur berfungsi sesuai spesifikasi.

Tahap Implementation (Implementasi) melibatkan pengujian aplikasi yang telah dikembangkan pada perangkat pengguna. Pada tahap ini, aplikasi diujicobakan kepada sejumlah pengguna untuk mendapatkan umpan balik terkait performa, pengalaman pengguna, dan antarmuka aplikasi. Data hasil implementasi ini digunakan untuk mengevaluasi apakah aplikasi telah memenuhi kebutuhan dan tujuan penelitian.

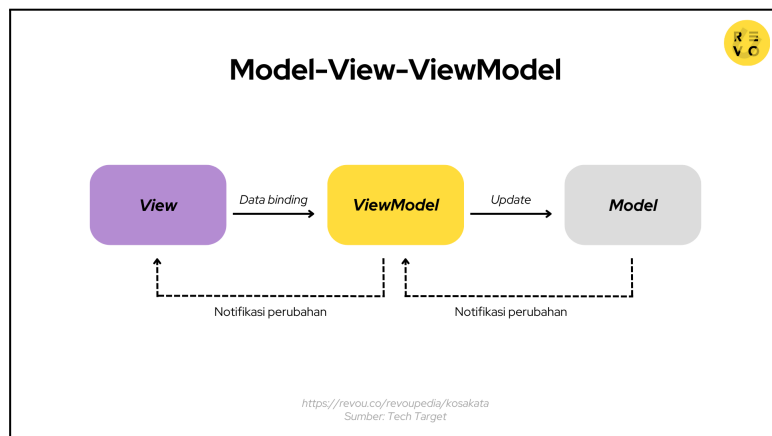
Tahap terakhir adalah Evaluation (Evaluasi), di mana performa aplikasi dianalisis secara menyeluruh berdasarkan hasil pengujian. Evaluasi ini mencakup pengukuran performa aplikasi, stabilitas mode offline, efektivitas state management Provider, serta keandalan arsitektur MVVM. Masukan yang diperoleh dari pengguna digunakan untuk melakukan perbaikan akhir sebelum aplikasi dirilis secara penuh.

### 3.2. Desain Sistem

Desain sistem pada penelitian ini bertujuan untuk menggambarkan alur kerja dan komponen utama yang digunakan dalam pengembangan aplikasi Al-Qur'an digital berbasis mode offline. Desain sistem ini dirancang agar aplikasi memiliki struktur yang modular, responsif, dan mendukung aksesibilitas data secara offline. Komponen utama desain sistem meliputi arsitektur Model-View-ViewModel (MVVM), penggunaan state management Provider, dan implementasi basis data lokal Realm Database. Berikut penjelasan rinci masing-masing komponen:

#### 3.2.1. Implementasi Arsitektur MVVM

Arsitektur MVVM diterapkan untuk memisahkan logika bisnis, antarmuka pengguna, dan pengelolaan data. Dengan pendekatan ini, setiap komponen memiliki peran spesifik yang membantu menciptakan aplikasi yang lebih terstruktur, modular, dan mudah dipelihara. Pendekatan ini juga mempermudah proses pengembangan aplikasi karena setiap lapisan bekerja secara terpisah, sehingga memungkinkan pembaruan atau pengujian dilakukan tanpa mempengaruhi komponen lainnya.



Gambar 3.2. Alur Arsitektur MVVM

Pada Gambar 3.2 ditampilkan alur kerja dari arsitektur MVVM, yang terdiri dari tiga komponen utama, yaitu Model, View, dan ViewModel. Arsitektur ini dirancang untuk memisahkan logika bisnis, pengelolaan data, dan antarmuka pengguna secara jelas, sehingga mempermudah proses

pengembangan aplikasi yang terstruktur dan dapat diandalkan. Ketiga komponen tersebut memiliki peran masing-masing sebagai berikut:

- **Model:** Bertugas mengelola logika bisnis dan data, seperti teks ayat Al-Qur'an, terjemahan, dan tafsir. Data ini diambil dari Realm Database dan diproses di bagian ini.
- **ViewModel:** Sebagai penghubung antara Model dan View. ViewModel menerima data dari Model, memprosesnya, dan menyediakan data untuk ditampilkan di antarmuka pengguna. ViewModel juga berperan dalam memproses interaksi pengguna, seperti pencarian surah atau penandaan halaman terakhir.
- **View:** Bagian antarmuka pengguna yang berfokus pada pengalaman pengguna. View hanya menerima data dari ViewModel tanpa mengolah logika bisnis, sehingga mempermudah desain tampilan aplikasi.

Pendekatan MVVM memastikan bahwa setiap komponen terpisah secara logis, meningkatkan fleksibilitas, skalabilitas, dan kemudahan pengujian aplikasi.

### 3.2.2. Implementasi State Management Provider

Provider digunakan sebagai solusi utama untuk state management dalam pengembangan aplikasi Flutter, memberikan cara yang efisien untuk mengelola dan berbagi data antar-widget[11]. Dengan memanfaatkan Provider, data dapat dikelola dengan struktur yang rapi dan diperbarui secara dinamis, sehingga mempermudah pengembang dalam membuat aplikasi yang responsif dan interaktif. Salah satu keunggulan utama dari Provider adalah kemampuannya untuk menyederhanakan proses pengelolaan state, terutama pada aplikasi dengan fitur kompleks seperti pencarian ayat, penandaan halaman terakhir, atau pengaturan preferensi pengguna[12].

Komponen penting dalam Provider adalah ChangeNotifier, yang bertugas mendeteksi dan mengelola perubahan data. Setiap kali data mengalami perubahan, ChangeNotifier akan memberi tahu View terkait

untuk memperbarui tampilannya, sehingga aplikasi selalu menyajikan informasi terkini kepada pengguna. Selain itu, ada Consumer, yang berfungsi sebagai penghubung antara View dan ViewModel. Consumer memastikan bahwa setiap perubahan data yang terjadi di ViewModel dapat langsung diterapkan pada View tanpa memerlukan pembaruan manual. Dengan pendekatan ini, pengelolaan data menjadi lebih efisien, dan View dapat tetap ringan karena tidak perlu memuat logika bisnis.

Keunggulan lain dari Provider adalah efisiensinya dalam memastikan performa aplikasi tetap optimal, meskipun terdapat banyak fitur yang berjalan secara bersamaan[12]. Misalnya, fitur-fitur seperti pengelolaan daftar ayat favorit, navigasi antarsurah, atau penyesuaian tema aplikasi dapat dilakukan tanpa menyebabkan beban berat pada aplikasi. Dengan menggunakan Provider, aplikasi mampu merespons perubahan data secara real-time, meningkatkan pengalaman pengguna secara keseluruhan. Hal ini sangat penting dalam aplikasi seperti Al-Qur'an digital, di mana interaksi pengguna sering terjadi dan membutuhkan tampilan data yang cepat serta akurat.

### **3.2.3. Implementasi Realm Local Database**

Realm Local Database digunakan sebagai solusi penyimpanan data lokal yang efisien dalam pengembangan aplikasi Flutter, menawarkan pendekatan yang lebih cepat dan lebih mudah dibandingkan dengan SQLite tradisional[13]. Dengan memanfaatkan Realm, data dapat disimpan secara lokal dengan struktur berorientasi objek, memungkinkan pengembang untuk mengelola data kompleks dengan cara yang lebih intuitif dan mudah dipelihara. Salah satu keunggulan utama dari Realm adalah kemampuannya untuk menyediakan operasi CRUD yang cepat dan efisien, terutama pada aplikasi yang membutuhkan penyimpanan data offline seperti cache konten, riwayat aktivitas pengguna, atau pengaturan aplikasi.

Komponen penting dalam Realm adalah Realm Object, yang berfungsi sebagai model data yang mendefinisikan struktur dan hubungan antar objek dalam database. Setiap kali terjadi perubahan pada data, Realm

secara otomatis mengelola pembaruan dan sinkronisasi, sehingga data selalu konsisten di seluruh aplikasi. Selain itu, Realm juga menyediakan fitur Realm Results, yang memungkinkan pengambilan dan pemfilteran data secara reaktif. Ketika data dalam database berubah, Results akan secara otomatis memperbarui dirinya, memastikan bahwa aplikasi selalu menampilkan data terkini tanpa perlu melakukan query ulang secara manual.

## DAFTAR PUSTAKA

- [1] Riyadhi, I. M., Purnamasari, I., & Prihandani, K. (2023). Penerapan Pola Arsitektur Mvvm Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android. *INFOTECH journal*, 9(1), 147-158.
- [2] Arif, M., Musthafa, A., & Muriyatmoko, D. (2019). Implementasi Pola Arsitektur Model-View-ViewModel (MVVM) pada Sistem Informasi Akademik Universitas Darussalam Gontor Berbasis Mobile.
- [3] Tetiana, V., Kusumo, D. S., & Andrian, M. (2023). Analisis Pengaruh Pola Arsitektur Model View ViewModel (MVVM) terhadap Kinerja Aplikasi Mobile dengan Menerapkan Application Programming Interface (API) Covid 19. *eProceedings of Engineering*, 10(3).
- [4] Prasetyo, E., & Raharjo, B. (2021). *Penerapan Design Pattern MVVM dan Clean Architecture pada Pengembangan Aplikasi Agree Partner*. Jurnal Automata, 2(2), 123–134.
- [5] Siregar, R. (2021). *Implementasi Pola Arsitektur MVVM pada Front-End Aplikasi Mobile UGFoodHub untuk User Profile Customer Menggunakan State Management Provider Flutter Berbasis Android*. Universitas Gunadarma.
- [6] Misniati, M., Hidayat, R., & Suharmon, S. (2024). Pengembangan Asesmen Pendidikan Agama Islam dan Budi Pekerti Bentuk Quizizz offline Mode Kertas tentang Keragaman pada Materi Al-Qur'an dan Hadits di SDN 13 Pasar Remaja Kota Sawahlunto. *Indonesian Research Journal on Education*, 4(4), 266-275.
- [7] Sholeh, M., Ridhoni, I. W., & Basuki, U. J. (2022). Pengembangan Aplikasi Alquran Online dengan Memanfaatkan Rest Api. *Device*, 12(2), 51-59.
- [8] Lovric, L., Fischer, M., Röderer, N., & Wunsch, A. (2023). Evaluation of the Cross-Platform Framework Flutter Using the Example of a Cancer Counselling App. In *ICT4AWE* (pp. 135-142).
- [9] Chen, Y., Huang, J., & Zhang, W. (2019). An Efficient Offline Mode Implementation Using Realm Database in Mobile Applications. *Journal of Mobile Computing and Applications*, 15(4), 33-42.

- [10] Singh, N., Kumar, R., & Jain, A. (2020). Enhancing Mobile App User Experience Using Flutter: A Framework Perspective. *International Journal of Interactive Mobile Technologies*, 14(13), 120-135.
- [11] Husain, I., Purwantoro, P., & Carudin, C. (2023). Analisis Performa State Management Provider Dan Getx Pada Aplikasi Flutter. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(2), 1417-1422.
- [12] Prayoga, R. R., Syalsabila, A., Munawar, G., & Jumiyan, R. (2021). Performance Analysis of BLoC and Provider State Management Library on Flutter. *Jurnal Mantik*, 5(3), 1591-1597.
- [13] Andersson, T. (2018). Analysis and quantitative comparison of storage, management, and scalability of data in Core Data system in relation to Realm.