

RANCANG BANGUN APLIKASI THE HAFIZ GUNA MENINGKATKAN HAFALAN AL-QUR'AN BERBASIS MOBILE

(Design And Development Of The Hafiz Mobile-Based Application To Improve Al-Qur'an Memorization)

Muhammad Rizqi Aditia, Nadiyah Agitha*, Sri Endang Anjarwani

^[1]Dept Informatics Engineering, Mataram University

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: rizqiaditia27@gmail.com, [nadiya,endang]@unram.ac.id

Abstract

The Quran is a holy book that serves as the foundation for the lives of Muslims. Therefore, it is a duty for Muslims to study the Quran and understand its content from an early age. One way that Muslims can understand and practice the content of the Quran is by memorizing it. According to data from the Ministry of Religion of the Republic of Indonesia in 2020, there are 26,973 Islamic boarding schools (pesantren) spread throughout Indonesia. With the increasing number of pesantren in Indonesia, it is expected that many people will become memorizers of the Quran or Hafiz. However, 126 Hafiz face difficulties when memorizing outside of the pesantren environment. This is due to lack of time for review, lack of memorizing partners, and lack of motivation. Based on this problem, the author proposes the idea of developing a mobile application called The Hafiz with Extreme Programming method and the Flutter framework. The application features useful features such as audio of the Quran, a feature for testing memorization. The application has been tested with unit testing, widget testing and received Mean Opinion Score value of 4.56, indicating that The Hafiz application runs well and meets user needs.

Keywords: Mobile Application, Qur'an Memorization, Muroja'ah, Hafiz, Extreme Programming

*Penulis Korespondensi

1. PENDAHULUAN

1.1. Latar Belakang

Al-Qur'an merupakan kitab suci untuk melatar belakangi kehidupan umat manusia yang memeluk agama Islam. Al-Qur'an juga menjadi petunjuk untuk dipelajari yang akan membantu umat muslim menemukan nilai-nilai yang dapat dijadikan pedoman untuk menyelesaikan permasalahan hidup. Oleh karenanya, merupakan sebuah kewajiban bagi umat muslim untuk mempelajari Al-Qur'an dan memahami isi dari Al-Qur'an sejak dini[1]. Salah satu cara yang dapat dilakukan oleh umat muslim untuk memahami dan mengamalkan isi Al-Qur'an adalah dengan Menghafalkannya[2].

Menghafal Al-Quran mempunyai manfaat yang besar. Selain merupakan perbuatan yang mulia, menghafal Al-Qur'an juga dapat membuka pintu-pintu kebaikan. Imam Nawawi dalam kitab *At-Tibyan Fi Adabi Hamalati* Al-Qur'an menjelaskan beberapa manfaat menghafal Al-Qur'an, di antaranya para penghafal Al-Qur'an memiliki daya ingat yang tajam

dan bersih intuisinya[3]. Begitu banyak keuntungan lain yang didapat para penghafal Al-Qur'an.

Seorang penghafal Al-Qur'an sangat dianjurkan untuk menjaga hafalannya sebagaimana sabda Rasulullah ﷺ yang artinya, "*Jagalah (hafalan) Al-Qur'an. Karena demi Dzat yang jiwa Muhammad berada di tangan-Nya, Al-Qur'an jauh lebih cepat hilang melebihi unta yang diikat*" (HR. Bukhori).

Hadist tersebut menjadi dorongan agar para penghafal Al-Qur'an senantiasa menjaga hafalannya. Menjaga hafalan Al-Qur'an dapat dilakukan dengan mengulang-ulang ayat yang sudah dihafal atau disebut dengan murojaah.

Salah satu lingkungan yang sangat mendukung untuk menghafal Al-Qur'an adalah institusi pendidikan pondok pesantren atau pondok tahfiz. Setiap tahun, jumlah pondok pesantren di Indonesia semakin meningkat. Menurut data Kementerian Agama Republik Indonesia tahun 2020, tercatat bahwa terdapat 26.973 pondok pesantren yang tersebar di seluruh Indonesia[4], dengan bertambahnya jumlah pesantren di Indonesia tentu diharapkan untuk

menghasilkan banyak penghafal Al-Qur'an atau Hafiz. Namun, terdapat berbagai permasalahan yang dialami para penghafal. Berdasarkan hasil survei yang dilakukan penulis terhadap 157 responden yang pernah menghafal Al-Qur'an baik di pondok pesantren ataupun di tempat lain, sebanyak 126 responden setuju bahwa dengan tidak berada pada lingkungan yang mendukung untuk menghafal seperti pondok pesantren menyebabkan menurunnya kualitas hafalan. Menurunnya kualitas hafalan disebabkan oleh beberapa hal, 79 responden mengalami permasalahan terkait kurangnya waktu mengulang hafalan, 84 responden kekurangan kawan atau guru sebagai tempat menyetor atau menguji hafalan dan 78 responden kekurangan motivasi untuk terus menjaga hafalannya.

Seiring dengan perkembangan teknologi, mobilitas masyarakat pun ikut meningkat, hal ini ditandai dengan semakin maraknya penggunaan perangkat bergerak atau *mobile device* seperti *smartphone* berbasis android[5].

Berangkat dari permasalahan yang dialami para penghafal Al-Qur'an serta melihat hasil survei di mana 137 responden setuju dengan pengembangan aplikasi *mobile* untuk menyelesaikan permasalahan tersebut, penulis mengusulkan gagasan **Rancang Bangun Aplikasi The Hafiz Guna Meningkatkan Hafalan Al-Qur'an Berbasis Mobile**. Sebelumnya, penulis telah mengembangkan aplikasi The Hafiz berbasis *website* dengan alamat "thehafiz.id" sebagai penelitian awal dan mendapat respon yang sangat positif dari para penghafal Al-Qur'an. Aplikasi The Hafiz berbasis *mobile* merupakan pemutakhiran dari *website* "thehafiz.id" agar dapat diakses secara *offline* oleh pengguna.

Pengembangan aplikasi The Hafiz berbasis *mobile* menggunakan metode *Extreme Programming* yang merupakan metode pengembangan perangkat lunak yang cenderung mengadopsi pendekatan berorientasi objek. Metode ini juga efektif untuk pengembangan aplikasi yang mengalami perubahan yang cepat[6], sehingga aplikasi The Hafiz dapat dikembangkan dengan efisien, praktis dan fleksibel. Aplikasi The Hafiz memiliki fitur-fitur yang bermanfaat seperti audio Al-Qur'an yang dapat diputar berulang-ulang untuk memudahkan dalam menambah ataupun menguatkan hafalan, fitur untuk menguji hafalan dengan suara ataupun dengan tulisan agar pengguna dapat menguji hafalan secara mandiri guna meningkatkan kekuatan hafalannya, dan fitur notifikasi pengingat hafalan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan, maka rumusan yang dapat diambil adalah bagaimana

merancang dan membangun aplikasi The Hafiz berbasis *mobile* untuk meningkatkan hafalan Al-Qur'an.

1.3. Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah merancang dan membangun aplikasi berbasis *mobile* yang efektif dalam membantu para penghafal Al-Qur'an dalam mengatasi masalah-masalah yang mereka hadapi dalam menjaga maupun menambah hafalannya menggunakan metode *Extreme Programming*.

2. TINJAUAN PUSTAKA

2.1. Dasar Teori

Dalam upaya mencapai tujuan penelitian ini, peneliti merujuk kepada sejumlah teori yang berkaitan dengan dua aspek penting, yaitu menghafal Al-Qur'an dan pengembangan aplikasi *mobile*.

2.1.1. Al-Qur'an

Al-Qur'an adalah kitab suci umat islam yang diwahyukan oleh Allah *Subhanallahu wa ta'ala* kepada Nabi Muhammad ﷺ melalui malaikat Jibril, Al-Qur'an terdiri dari 30 juz, 114 surah, dan 6.236 ayat. Setiap muslim diwajibkan untuk mempelajari Al-Qur'an karena Al-Qur'an merupakan pedoman kehidupan dunia dan akhirat yang kesucian dan kemurniannya dijaga oleh Allah[7]. Selain itu, membaca Al-Qur'an juga termasuk amal ibadah[8].

2.1.2. Muroja'ah

Muroja'ah merupakan praktik mengulang-ulang hafalan Al-Qur'an yaitu dengan mengulang ayat atau surah yang telah dihafal sebelumnya, dengan tujuan agar hafalan tersebut semakin kuat. Tanpa *muroja'ah* hafalan akan mudah hilang dari ingatan[9].

2.1.3. Android

Android merupakan sebuah sistem operasi *open source* yang digunakan untuk perangkat *mobile* berbasis Linux[10]. Android awalnya dikembangkan oleh Android Inc. Kemudian diakuisisi oleh Goggle pada tahun 2005. Pada tahun 2007, Android secara resmi dirilis bersamaan dengan pembentukan *Open Handset Alliance*, kemudian pada tahun 2008 *smartphone* Android pertama mulai dijual[11]. Sebagai salah satu produk dari perusahaan raksasa Google, Android menjadi salah satu komponen penting di sektor pasar *smartphone*[12]. Android terus dikembangkan sampai saat ini versi terbaru Android adalah 12.0 yang dirilis pada Oktober 2021. Antarmuka Android bersifat

Android memberikan kebebasan kepada pengembang perangkat lunak *mobile*, itulah yang menjadi salah satu alasan mengapa Android menjadi sangat populer di kalangan pengguna maupun produsen *smartphone*.

2.1.4. Dart

Dart adalah bahasa pemrograman yang berbasis dengan konsep pemrograman berorientasi objek (OOP) yang dikembangkan oleh google, bahasa ini pertama kali diresmikan pada konferensi GOTO di Denmark pada tahun 2011. Dart dapat digunakan untuk membuat aplikasi *mobile*, *game*, *web*, *IoT*, dan *back-end* (CLI)[13]. *Syntax* bahasa pemrograman dart menggunakan C-Style *syntax* sehingga dart mirip dengan bahasa pemrograman javascript, java, swift, dan C. Dart juga bersifat *open source* serta mendukung pengembangan aplikasi berbagai platform seperti Android, iOS, dan web hanya dengan satu *codebase*.

2.1.5. Flutter

Flutter adalah sebuah *framework* yang menggunakan bahasa pemrograman Dart yang digunakan untuk membuat aplikasi *cross-platform* dengan *performa* tinggi. Flutter dirilis secara publik oleh Google pada tahun 2016. *Framework* Flutter dapat memberikan *performa* setara dengan aplikasi *native* karena Flutter tidak bergantung pada *widget* OEM pada perangkat melainkan *me-render* setiap komponen menggunakan *engine* dengan *performa* tinggi miliknya[14]. *Engine* milik flutter dibuat dengan bahasa C/C++. Selain itu, flutter juga mendukung *hot-reload* pada proses pengembangan aplikasi.

2.1.6. SQLite Database

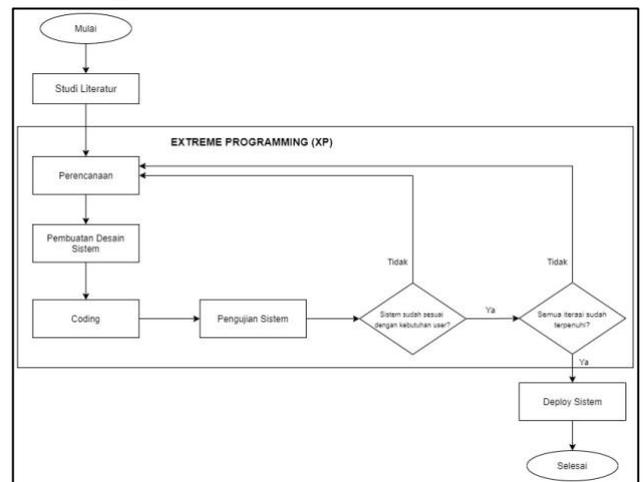
SQLite merupakan sistem manajemen basis data relasional yang tersedia secara *default* di platform Android[15]. SQLite berukuran relatif ringan dan bersifat *ACID-compliant*. Seperti sistem basis data pada umumnya, SQLite juga mendukung beragam tipe data seperti TEXT, INTEGER, dll.

2.1.7. Extreme Programming (XP)

Extreme Programming (XP) merupakan salah satu metode pengembangan perangkat lunak yang mengadopsi pendekatan pemrograman berorientasi objek (OOP)[16]. XP merupakan bagian dari *agile methods*. Metode XP cocok digunakan oleh tim berskala kecil sampai medium, metode ini juga bersifat adaptif dan fleksibel yang di mana dapat memudahkan tim jika terjadi perubahan *requirement* pada proses pengembangan aplikasi.

3. METODOLOGI

Dalam penelitian ini, penulis menggunakan metode pengembangan *Extreme Programming* (XP) yang terdiri dari 4 tahapan yaitu perencanaan (*planning*), perancangan (*design*), pengkodean (*coding*), dan pengujian (*testing*). Setiap tahapan tersebut mengalami serangkaian iterasi atau siklus pendek yang memungkinkan peneliti untuk meningkatkan dan melakukan perbaikan aplikasi. Berikut merupakan tahap-tahap pengembangan sistem yang diilustrasikan dengan diagram alir pada Gambar 1.



Gambar 1. Metodologi pengembangan sistem

Pada Gambar 1 dapat dilihat Metodologi pengembangan *Extreme Programming* (XP). Pada metodologi ini, terdapat iterasi berulang sehingga dapat dilakukan perbaikan sebelum aplikasi dirilis, pada penelitian ini dilakukan 2 iterasi. Hal inilah yang membedakan XP dengan metode pengembangan tradisional seperti *waterfall*. Selain itu, dalam XP digunakan *user stories* pada tahap desain sistem yang dapat menjabarkan kebutuhan pengguna secara lebih baik dan *refactoring* yang dilakukan secara teratur untuk menjaga kebersihan dan kejelasan kode program.

3.1. Studi Literatur

Tahap awal dalam penelitian ini adalah studi literatur, di mana peneliti mengumpulkan data dan informasi serta memahami teori-teori dari buku, jurnal, dan skripsi yang berkaitan dengan pengembangan aplikasi untuk penghafal Al-Qur'an. Hasil dari tahapan ini berupa data dan informasi yang relevan sebagai pendukung dalam pembuatan aplikasi The Hafiz.

3.2. Planning

Setelah melakukan studi literatur, tahap selanjutnya adalah melakukan perencanaan atau *planning*. Pada tahap ini peneliti menganalisis permasalahan serta kebutuhan yang diperlukan dalam pengembangan aplikasi The Hafiz. Penelitian ini berfokus pada pencarian solusi untuk permasalahan yang dialami para penghafal Al-Qur'an. Pada tahap ini penulis mengumpulkan data-data yang diperlukan.

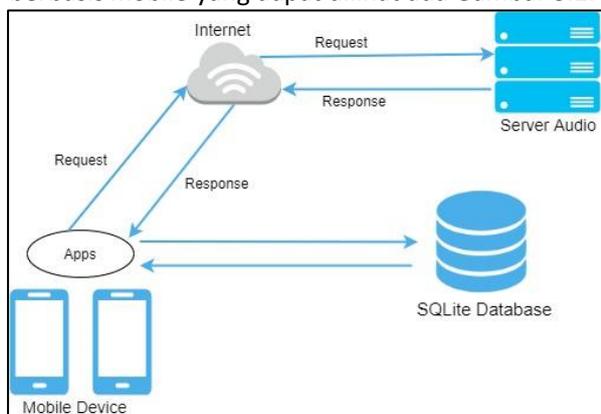
berupa *database* Al-Qur'an Indonesia, audio Al-Qur'an, dan kuesioner penelitian. Selain itu, peneliti menentukan *requirement* pengguna dan ditulis dalam bentuk *user stories*.

3.3. Design

Setelah semua kebutuhan pembangunan sistem terkumpul, tahap selanjutnya adalah tahap *design* yang merupakan tahap pembuatan perancangan sistem. Pada penelitian ini, sistem dirancang dengan konsep berorientasi objek menggunakan *Unified Modelling Language* (UML) dan *Entity Relationship Diagram* (ERD).

3.3.1. Arsitektur Sistem

Berikut merupakan Arsitektur Aplikasi The Hafiz berbasis *mobile* yang dapat dilihat ada Gambar 3.2.



Gambar 2. Arsitektur Sistem

3.3.2. Use Case Diagram

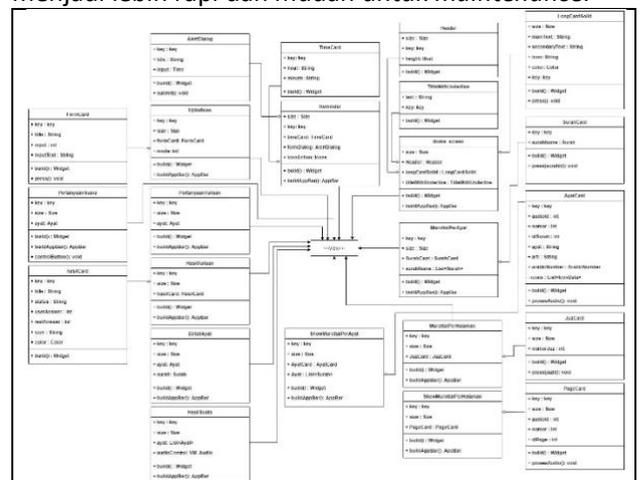
Berikut merupakan *use case* diagram Aplikasi The Hafiz berbasis *mobile* yang menggambarkan hubungan antara pengguna yang merupakan masyarakat umum dengan sistem yang dibuat dapat dilihat pada Gambar 3.

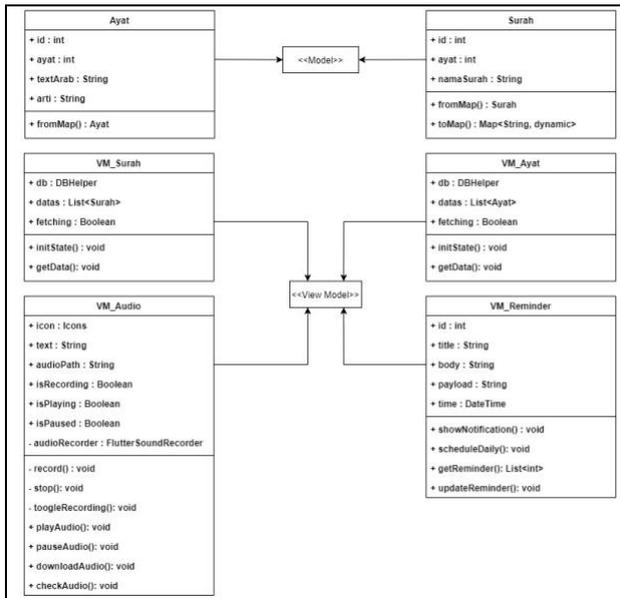


Gambar 3. Use Case Diagram

3.3.3. Class Diagram

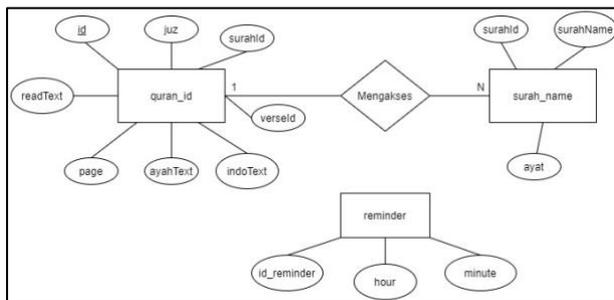
Gambar 4 merupakan *class* diagram pada aplikasi The Hafiz berbasis *mobile*. Aplikasi ini dibangun dengan menerapkan arsitektur MVVM (*Model, View, View Model*) karena pada arsitektur ini proses *testing* mudah dilakukan serta lebih mudah untuk penambahan fitur aplikasi ke depannya[17]. Terdapat 3 komponen pada arsitektur ini yaitu *view* merupakan komponen yang terkait dengan *user interface* aplikasi, *model* merupakan representasi dari data yang digunakan, dan *view model* merupakan komponen yang bertugas untuk berinteraksi dengan *model* serta menyediakan data untuk *view*. Arsitektur ini memisahkan *code business logic* dengan *interface* aplikasi sehingga *code* menjadi lebih rapi dan mudah untuk *maintenance*.





Gambar 4. Class diagram

3.3.4. Entity Relationship Diagram



Gambar 5. Entity Relationship Diagram

Gambar 5 merupakan Entity Relationship Diagram (ERD) aplikasi The Hafiz berbasis *mobile* yang menggambarkan relasi antar entitas. Dimana pada ERD tersebut terdapat tiga entitas yaitu *quran_id*, *surah_name*, dan *reminder*.

3.4. Coding

Setelah melalui tahap merancang arsitektur sistem dan desain aplikasi, tahap berikutnya adalah implementasi rancangan sistem ke dalam bentuk aplikasi menggunakan Bahasa Pemrograman. Proses *coding* untuk pembuatan aplikasi ini dilakukan oleh peneliti sendiri. Aplikasi akan dibangun berbasis Android dengan menggunakan Bahasa Pemrograman Dart. Dalam proses *coding* ini dibutuhkan beberapa *tools* berupa *framework* Flutter sebagai arsitektur *code* aplikasi dan *Visual Studio Code* sebagai *code editor*. Tahap Selanjutnya melakukan konfigurasi aplikasi dengan *database offline* yang digunakan yaitu SQLite.

3.5. Testing

Setelah implementasi aplikasi dengan Bahasa Pemrograman selesai, tahap berikutnya adalah pengujian sistem. Sistem diuji dengan menggunakan Unit Testing, *Widget Testing*, dan *Mean Opinion Score* (MOS). Unit testing digunakan untuk menguji komponen dasar aplikasi seperti *class* dan *function* sudah berjalan dengan baik dan sesuai yang diharapkan, sedangkan *Widget Testing* berfungsi untuk menguji komponen dari *user interface* pada aplikasi seperti tombol, kolom teks, dan lain-lain dapat berfungsi dengan baik. Selanjutnya metode *Mean Opinion Score* (MOS) digunakan untuk menguji sistem dengan meminta 27 orang penghafal Al-Qur'an menilai fungsionalitas beserta komponen-komponen pada *user interface* setiap fitur di aplikasi The Hafiz dengan angka mulai dari 1 (buruk) sampai 5 (sangat baik).

Setelah pengujian dilakukan, peneliti akan melakukan evaluasi berdasarkan masukan pengguna. Jika hasil evaluasi menunjukkan adanya kekurangan pada aplikasi maka peneliti akan memulai iterasi baru untuk mengatasi masalah-masalah tersebut.

3.6. Implementasi Sistem

Setelah semua tahap berhasil dilakukan, tahap terakhir adalah melakukan implementasi sistem pada kehidupan sehari-hari. Pada penelitian ini dibatasi sampai tahap pengujian.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Extreme Programming

Pada tahap implementasi, peneliti mulai menerapkan rancangan Pembangunan aplikasi sesuai dengan metode *extreme programming* dan rancangan yang sudah dibuat sebelumnya.

4.1.1. Planning

Pada tahap *planning*, peneliti telah merumuskan lingkup masalah yang akan diteliti, bahan yang diperlukan, dan menyusun rencana *user stories* dan *iteration plan* untuk pengembangan sistem dalam penelitian. Dengan demikian, peneliti telah mengumpulkan daftar kebutuhan sistem yang diperlukan yaitu *database* Al-Qur'an Indonesia, Audio bacaan Al-Qur'an per ayat dan per halaman, dan hasil kuesioner penelitian.

4.1.2. Design

Setelah peneliti mengumpulkan kebutuhan dan merancang desain UML untuk pengembangan aplikasi The Hafiz, selanjutnya pada tahap ini akan dilakukan

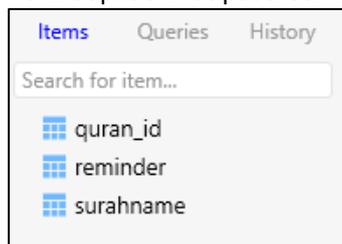
perancangan *interface* aplikasi sesuai dengan *user stories* yang sudah dibuat.

4.1.3. Coding

Pada tahap ini dilakukan implementasi perencanaan dan desain ke dalam bentuk pengembangan aplikasi The Hafiz berbasis *mobile*. Pengembangan aplikasi merujuk pada *user stories* dan *iteration plan* yang sudah dibuat.

4.1.3.1 Implementasi Database

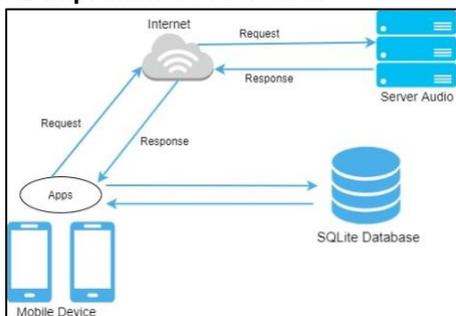
Berikut merupakan implementasi *database* dari aplikasi The Hafiz dapat dilihat pada Gambar 6.



Gambar 6. Struktur Database

Gambar 6 merupakan struktur dari *database* dengan nama "thehafiz-app.db" untuk aplikasi The Hafiz berbasis *mobile*. Struktur *database* ini terdiri dari 3 tabel yaitu tabel *quran_id*, *reminder* dan *surahname* di mana tabel *quran_id* dan *reminder* saling terhubung satu sama lain.

4.1.3.2 Implementasi Arsitektur



Gambar 7. Implementasi Arsitektur

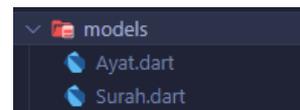
Gambar 7 Menggambarkan teknologi apa saja yang terdapat pada arsitektur aplikasi. Terdapat penggunaan *SQLite Database*, internet, server audio dan perangkat *Mobile* pengguna. Untuk dapat menjalankan aplikasi ini, pengguna membutuhkan *smartphone* yang terhubung dengan internet agar dapat mengunduh audio Al-Qur'an dari server saat pertama kali diputar. Penggunaan *SQLite database* bertujuan untuk menyimpan data Al-Qur'an Indonesia secara *offline*, sehingga pengguna tidak perlu terhubung ke internet untuk membaca maupun melakukan uji hafalan Al-Qur'an. Aplikasi The Hafiz

dibuat dengan bahasa pemrograman Dart dan *framework* Flutter.

4.1.3.3 Implementasi Class

Implementasi selanjutnya adalah implementasi Class dari Aplikasi The Hafiz yang diimplementasikan ke dalam coding menggunakan tools visual studio code dan *framework* Flutter. Pengembangan aplikasi ini menerapkan pola arsitektur MVVM (Model, View, ViewModel) yang bertujuan agar code dapat lebih terorganisir dan lebih mudah untuk pengembangan. Berikut merupakan implementasi class dari aplikasi The Hafiz.:

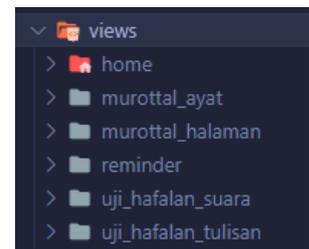
1. Class Model



Gambar 8. Class Model

Gambar 8 merupakan implementasi dari *Class model*. Class ini berfungsi untuk menyimpan data dan logika bisnis yang terkait dengan aplikasi. *Class* ini hanya bertanggung jawab untuk menyimpan dan mengelola data serta melakukan operasi-operasi yang dibutuhkan untuk mengolah data tersebut.

2. Class View



Gambar 9. Class View

Gambar 9 merupakan implementasi dari *Class view*. Class ini berfungsi untuk mengelola tampilan serta bertanggung jawab untuk menampilkan data yang diperoleh dari model kepada pengguna dan menerima masukan dari pengguna.

3. Class ViewModel

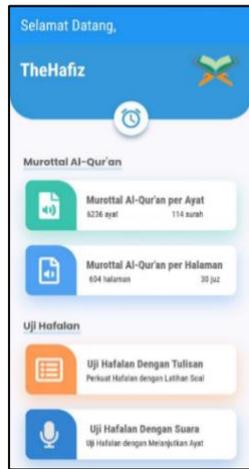
Gambar 10 di bawah ini merupakan implementasi dari *Class ViewModel*. Class ini berfungsi untuk mengolah data dari Model sehingga siap untuk ditampilkan ke pengguna melalui View. *ViewModel* juga bertanggung jawab untuk menjembatani komunikasi antara Model dan View. *ViewModel* tidak terlibat langsung dengan tampilan aplikasi, tetapi ia menyediakan data dan logika yang diperlukan oleh View untuk ditampilkan ke pengguna.



Gambar 10. Class ViewModel

4.1.3.4 Implementasi Interface

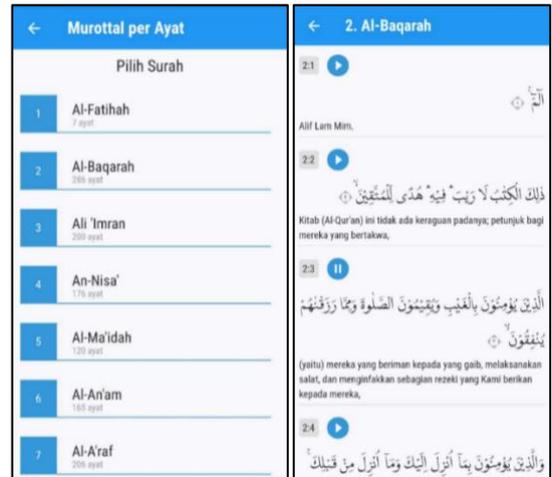
Berikut merupakan implementasi *interface* aplikasi yang telah dibuat.



Gambar 11. Halaman Utama Aplikasi

Gambar 11 merupakan implementasi halaman utama yang akan muncul pertama kali saat aplikasi The Hafiz dibuka, melalui halaman ini pengguna dapat mengakses semua fitur yang tersedia.

Gambar 12 di bawah ini merupakan implementasi halaman untuk mendengarkan audio murottal Al-Qur'an per ayat. Pada halaman ini, pengguna akan memilih surah yang ingin diputar, jika audio dari surah tersebut belum diunduh maka pengguna akan diminta untuk mengunduh terlebih dahulu, kemudian sistem akan menampilkan proses pengunduhan audio, jika sudah selesai maka audio otomatis diputar.

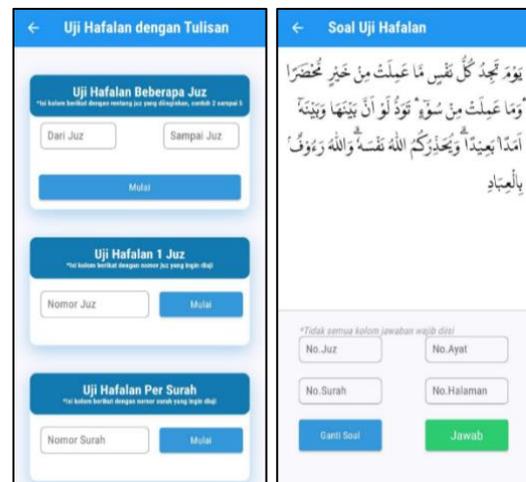


Gambar 12. Halaman murottal Al-Qur'an per ayat.



Gambar 12. Halaman murottal Al-Qur'an per halaman.

Gambar 12 merupakan implementasi halaman untuk mendengarkan audio murottal Al-Qur'an per halaman. Pada halaman ini, pengguna akan memilih juz yang ingin diputar, jika audio halaman yang terdapat pada juz tersebut belum diunduh maka pengguna akan diminta untuk mengunduh terlebih dahulu, jika sudah maka audio bisa langsung diputar.



Gambar 13. Halaman uji hafalan dengan tulisan

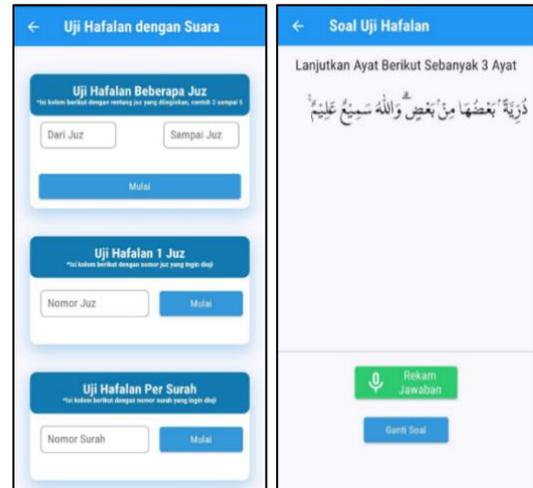
Gambar 13 merupakan implementasi halaman uji hafalan dengan tulisan. Pada halaman ini, pengguna akan mengisi nomor juz atau surah yang ingin diuji, kemudian sistem akan menampilkan satu ayat soal secara acak beserta kolom jawaban pada halaman berikutnya.



Gambar 14. Halaman hasil uji hafalan dengan tulisan

Gambar 14 merupakan implementasi halaman untuk melihat hasil uji hafalan dengan tulisan. Pada halaman ini ditampilkan hasil jawaban uji hafalan dengan tulisan dalam bentuk tabel, jika benar maka tabel akan berwarna hijau, jika salah maka tabel akan berwarna merah beserta kunci jawaban.

Gambar 15 di bawah ini merupakan implementasi halaman untuk melakukan uji hafalan dengan suara. Sistem akan menampilkan satu ayat soal secara acak dan tombol untuk merekam jawaban pada halaman berikutnya. Jika sudah selesai menjawab, pengguna dapat menekan tombol stop.



Gambar 15. Halaman uji hafalan dengan suara



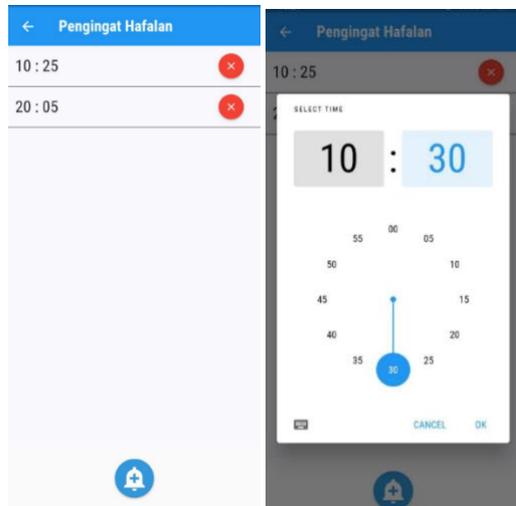
Gambar 16. Halaman uji hafalan dengan suara

Gambar 16 merupakan implementasi halaman untuk melihat hasil uji hafalan dengan suara. Pada halaman ini ditampilkan tiga ayat lanjutan dari soal dan rekaman suara pengguna sehingga pengguna bisa mengoreksi jawabannya secara mandiri. Halaman ini akan muncul secara otomatis ketika pengguna menekan tombol stop rekaman pada halaman sebelumnya.

Gambar 17 merupakan implementasi halaman untuk melihat detail ayat soal uji hafalan. Pada halaman ini ditampilkan informasi detail mengenai ayat yang menjadi soal uji hafalan dengan tulisan. Informasi yang disediakan berupa nomor juz, nama surah, nomor ayat, nomor halaman, dan terjemahan Bahasa Indonesia.



Gambar 17. Halaman detail ayat soal



Gambar 18. Halaman Kelola jadwal notifikasi pengingat hafalan

Gambar 18 merupakan implementasi halaman untuk melihat dan mengelola jadwal notifikasi pengingat hafalan. Pada halaman ini, pengguna dapat melihat, menghapus, dan menambah jadwal notifikasi. Notifikasi akan ditampilkan oleh sistem sesuai dengan jam dan menit yang sudah diatur pengguna.

4.1.4. Testing

Setelah proses implementasi selesai dilakukan, proses selanjutnya adalah pengujian. Pengujian pada aplikasi The Hafiz menggunakan metode *Unit Testing*, *Widget Testing*, dan *Mean Opinion Score (MOS)*. *Unit testing* berfungsi untuk menguji komponen dasar aplikasi seperti *class* dan *function* apakah sudah berjalan dengan baik dan sesuai yang diharapkan sedangkan *Widget Testing* berfungsi untuk menguji komponen dari *user interface* pada aplikasi seperti

tombol, kolom teks, dan lain lain dapat berfungsi dengan baik. Selanjutnya metode *Mean Opinion Score (MOS)* digunakan untuk menguji sistem dengan meminta 27 orang penghafal Al-Qur'an menilai fungsionalitas beserta komponen-komponen pada *user interface* setiap fitur di aplikasi The Hafiz dengan angka mulai dari 1 (buruk) sampai 5 (sangat baik)..

1. Hasil Pengujian Unit Testing

Metode pengujian menggunakan metode *Unit testing* merupakan metode untuk menguji komponen dasar aplikasi seperti *class* dan *function* apakah sudah berjalan dengan baik dan sesuai yang diharapkan. Dari sembilan skenario pengujian dengan *unit testing*, seluruh skenario mendapatkan nilai hasil "Diterima". Dengan begitu, dapat disimpulkan bahwa seluruh *function* pada aplikasi dapat berjalan dengan baik.

2. Hasil Pengujian Widget Testing

Metode pengujian menggunakan metode *Widget testing* merupakan metode yang berfungsi untuk menguji komponen dari *user interface* pada aplikasi seperti tombol, kolom teks, dan lain-lain dapat berfungsi dengan baik. Dari 25 skenario pengujian dengan *widget testing*, seluruh skenario mendapatkan nilai hasil "Diterima". Dengan begitu, dapat disimpulkan bahwa semua komponen tampilan pada aplikasi The Hafiz dapat berjalan sesuai yang diharapkan.

3. Hasil Pengujian Mean Opinion Score (MOS)

Selanjutnya metode pengujian Mean Opinion Score (MOS) digunakan untuk menguji sistem dengan meminta 27 orang masyarakat muslim yang pernah menghafal Al-Qur'an untuk menilai fungsionalitas setiap fitur di aplikasi The Hafiz melalui kuesioner dan memberikan masukan. Hasil penilaian dan masukan dari responden akan dikumpulkan dan dihitung untuk mendapatkan kesimpulan seberapa layak sistem dapat digunakan. Jika responden memberikan masukan terkait kekurangan dalam aplikasi, peneliti akan melakukan perbaikan pada aplikasi yang dikelompokkan ke dalam beberapa iterasi perbaikan. Kuesioner penilaian terdiri dari 12 pertanyaan sebagai berikut:

- Apakah tampilan aplikasi mudah dimengerti?
- Apakah fitur murottal Al-Qur'an per ayat mudah untuk dioperasikan?
- Apakah fitur murottal Al-Qur'an per halaman mudah untuk dioperasikan?

- d. Apakah kedua fitur murottal Al-Qur'an dapat membantu dalam mempermudah menghafal?
- e. Apakah fitur uji hafalan dengan tulisan mudah dioperasikan?
- f. Apakah fitur uji hafalan dengan tulisan dapat membantu dalam menguji hafalan?
- g. Apakah fitur uji hafalan dengan suara mudah dioperasikan?
- h. Apakah fitur uji hafalan dengan suara dapat membantu dalam menguji hafalan?
- i. Apakah kedua fitur uji hafalan dapat membantu dalam memperkuat hafalan Al-Qur'an?
- j. Apakah fitur pengingat hafalan mudah digunakan?
- k. Secara keseluruhan, apakah aplikasi mudah digunakan?
- l. Apakah aplikasi sudah sesuai dengan kebutuhan pengguna?

Responden akan diminta untuk menjawab kuesioner dengan pilihan jawaban sebagai berikut:

SS : Sangat setuju

S : Setuju

C : Cukup

TS : Tidak Setuju

STS : Sangat Tidak Setuju

Berdasarkan masukan responden, didapatkan beberapa kekurangan pada aplikasi sehingga peneliti perlu melakukan perbaikan dan penambahan fitur agar aplikasi sesuai dengan kebutuhan pengguna. Perbaikan dilakukan sebanyak dua iterasi.

A. Iterasi pertama

Pada iterasi pertama, dilakukan perbaikan pada fitur uji hafalan dengan tulisan. Berdasarkan penuturan salah satu pengguna melalui kuesioner, terdapat kesalahan pada halaman jawaban uji hafalan dengan tulisan di mana nomor halaman dan nomor ayat dari jawaban pengguna tertukar saat ditampilkan. Setelah melakukan analisa, peneliti menemukan kesalahan kode dan melakukan perbaikan sehingga jawaban pengguna pada bagian nomor halaman dan ayat dapat ditampilkan dengan benar.

B. Iterasi kedua

Setelah perbaikan iterasi pertama selesai, selanjutnya dilakukan perbaikan pada fitur murottal Al-Qur'an per ayat. Berdasarkan penuturan pengguna melalui kuesioner, pengguna mengalami kesulitan dalam mencari ayat terakhir yang dibaca dan meminta untuk ditambahkan fitur penanda ayat.

Berdasarkan penuturan tersebut, peneliti menambahkan icon pada komponen AyatCard sebagai

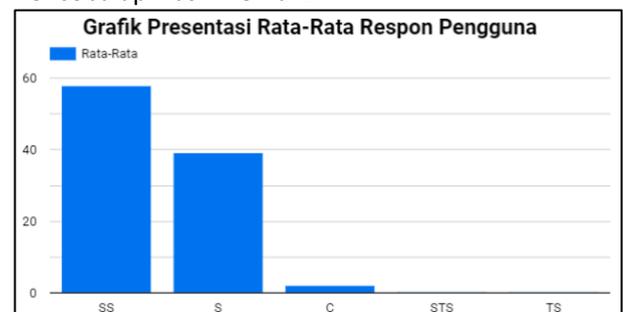
tombol untuk menandai ayat yang diinginkan dan nama surah beserta nomor ayat yang ditandai akan ditampilkan di halaman utama aplikasi, pengguna juga dapat menekan icon panah kanan untuk langsung dibawa ke ayat tersebut.

Tabel 1 merupakan hasil pengujian dengan menyebarkan kuesioner kepada 27 pengguna yang merupakan masyarakat muslim yang pernah menghafal Al-Qur'an. Didapatkan hasil rata-rata responden yang menjawab sangat setuju adalah 57,95%, setuju adalah 39,15%, dan cukup adalah 2,15%.

Tabel 1. Hasil pengujian *Mean Opinion Score*

Pertanyaan	SS		S		C		TS		STS	
	Total	%	Total	%	Total	%	Total	%	Total	%
Pertanyaan 1	18	66.7	9	33.3	0	0	0	0	0	0
Pertanyaan 2	16	59.3	11	40.7	0	0	0	0	0	0
Pertanyaan 3	16	59.3	11	40.7	0	0	0	0	0	0
Pertanyaan 4	17	63	9	33.3	1	3.7	0	0	0	0
Pertanyaan 5	16	59.3	10	37	1	3.7	0	0	0	0
Pertanyaan 6	15	55.6	10	37	2	7.4	0	0	0	0
Pertanyaan 7	16	59.3	11	40.7	0	0	0	0	0	0
Pertanyaan 8	15	55.6	10	37	2	7.4	0	0	0	0
Pertanyaan 9	14	51.9	13	48.1	0	0	0	0	0	0
Pertanyaan 10	17	63	10	37	0	0	0	0	0	0
Pertanyaan 11	15	55.6	12	44.4	0	0	0	0	0	0
Pertanyaan 12	15	55.6	11	40.7	1	3.7	0	0	0	0
Rata-rata		57.95%		39.15%		2.15%		0		0

Gambar 19 merupakan Grafik persentase rata-rata jawaban pengguna yang telah menggunakan mencoba aplikasi The Hafiz.



Gambar 19. Presentasi rata-rata jawaban pengguna.

Setelah mendapatkan hasil persentase jawaban pengguna pada keseluruhan pertanyaan, dapat ditentukan skala yang akan digunakan untuk penentuan nilai MOS. Tabel 2 berikut menunjukkan keterangan skala yang akan digunakan:

Tabel 2. Skala *Mean Opinion Score*.

MOS	Keterangan	Bobot Nilai	Kualitas
SS	Sangat setuju	5	Excellent
S	Setuju	4	Good
C	Cukup	3	Fair
TS	Tidak setuju	2	Poor
STS	Sangat tidak setuju	1	Bad

Tabel 3. Tabel hasil perhitungan *mean opinion score*.

Pertanyaan	SS(5)	S(4)	C(3)	TS(2)	STS(1)	Mean
Pertanyaan 1	18	9	0	0	0	4.67
Pertanyaan 2	16	11	0	0	0	4.59
Pertanyaan 3	16	11	0	0	0	4.59
Pertanyaan 4	17	9	1	0	0	4.59
Pertanyaan 5	16	10	1	0	0	4.56
Pertanyaan 6	15	10	2	0	0	4.48
Pertanyaan 7	16	11	0	0	0	4.59
Pertanyaan 8	15	10	2	0	0	4.48
Pertanyaan 9	14	13	0	0	0	4.52
Pertanyaan 10	17	10	0	0	0	4.63
Pertanyaan 11	15	12	0	0	0	4.56
Pertanyaan 12	15	11	1	0	0	4.52
MOS (Mean Opinion Score)						4,56

Tabel 3 menunjukkan hasil perhitungan MOS yang dilakukan pada 27 responden, hasil tersebut menunjukkan bahwa aplikasi The Hafiz berbasis mobile memiliki nilai MOS sebesar 4,56. Berdasarkan skala yang telah ditetapkan, nilai ini menunjukkan bahwa aplikasi tersebut memiliki kualitas yang baik untuk digunakan karena memiliki nilai di atas 4,0.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian dan pengembangan sistem yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. Pengembangan aplikasi The Hafiz dengan metode *Extreme Programming* dapat berjalan dengan efektif hingga selesai. Waktu pengerjaannya pun sesuai estimasi yang telah direncanakan. Metode ini sesuai dengan pengembangan aplikasi dalam waktu cepat dan butuh perubahan dalam masa pengembangan. Pengembangan aplikasi juga didukung dengan pemanfaatan *framework* Flutter yang memiliki banyak *package* pendukung dan penggunaan arsitektur MVVM dapat membantu peneliti dalam menyelesaikan aplikasi dengan lebih cepat.
2. Pengembangan aplikasi The Hafiz dalam penelitian ini memudahkan pengguna dalam menambah dan menjaga hafalan Al-Qur'an dan sudah sesuai dengan kebutuhan pengguna. Dibuktikan dengan hasil kuesioner yang telah diisi oleh 27 pengguna, di mana rata-rata yang responden menjawab sangat setuju adalah 57.95%, setuju adalah 39.15% dan cukup adalah 2.15%.
3. Berdasarkan hasil kuesioner yang telah diisi oleh 27 pengguna, diperoleh nilai *Mean Opinion Score* sebesar 4,56. Berdasarkan skala yang telah ditetapkan, nilai ini menunjukkan bahwa aplikasi tersebut memiliki kualitas yang baik untuk digunakan karena memiliki nilai di atas 4,0.

5.2. Saran

Saran yang dapat diberikan oleh peneliti untuk membuat aplikasi The Hafiz menjadi lebih baik lagi di masa depan, yaitu sebagai berikut:

1. Untuk pengembangan selanjutnya, peneliti dapat mengembangkan aplikasi dalam platform IOS agar dapat digunakan oleh semua kalangan pengguna perangkat mobile.
2. Untuk pengembangan aplikasi selanjutnya, peneliti dapat menambahkan metode uji hafalan baru berupa kuis atau permainan agar proses uji hafalan dapat lebih menarik beserta kecerdasan buatan untuk pengujian hafalan agar dapat menampilkan soal uji hafalan sesuai dengan titik lemah hafalan pengguna.

DAFTAR PUSTAKA

- [1] S. Orba Manullang, Mardani, and Aslan, "the Effectiveness of Al-Quran Memorization Methods for Millennials Santri During Covid-19 in Indonesia," *J. Pendidik. Islam*, vol. 4, no. 2, pp. 195–207, 2021, [Online]. Available: <http://ejournal.ikhac.ac.id/index.php/NAZHRUNA/>
- [2] A. A. Ghofur, "DAMPAK PENURUNAN HAFALAN AL-QUR'AN PADA PERILAKU ALUMNI PONDOK PESANTREN AL-HIKMAH KARANGGEDE BOYOLALI," 2020.
- [3] M. Oktapiani, "Tingkat Kecerdasan Spiritual Dan Kemampuan Menghafal Al-Qur'an," *Tahdzib Al-Akhlaq J. Pendidik. Islam*, vol. 3, no. 1, pp. 95–108, 2020, doi: 10.34005/tahdzib.v3i1.861.
- [4] K. A. Indonesia, "Pangkalan Data Pondok Pesantren." <https://ditpdpontren.kemenag.go.id/pdpp/statistik> (accessed Mar. 14, 2022).
- [5] A. F. Ramadhan, A. D. Putra, and A. Surahman, "Aplikasi Pengenalan Perangkat Keras Komputer Berbasis Android Menggunakan Augmented Reality (AR)," *J. Teknol. dan Sist. Inf.*, vol. 2, no. 2, pp. 24–31, 2021, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/JTSI>
- [6] I. Carolina and A. Supriyatna, "Penerapan Metode Extreme Programming dalam Perancangan Aplikasi Perhitungan Kuota SKS Mengajar Dosen," *J. IKRA-ITH Inform.*, vol. 3, no. 1, pp. 106–113, 2019.
- [7] A. A. Amrullah, *Tafsir Al-Azhar Jilid 5*, 5th ed. PUSTAKA NASIONAL PTE LTD SINGAPURA, 1999.
- [8] E. Iryani, "Al- Qur'an Dan Ilmu Pengetahuan Eva Iryani 1," *J. Ilm. Univ. Batanghari Jambi*, vol. 17, no. 3, p. 70, 2017.
- [9] R. Nurhidayah, "EKPERIMENTASI BIMBINGAN BELAJAR DENGAN TEKNIK TOKEN ECONOMY UNTUK MENINGKATKAN MUROJAAH JUS 30

- PADA SANTRI DI PONPES TAHFIDZUL QUR'AN AL AWWABIN BANDAR LAMPUNG," 2020.
- [10] E. Maiyana, "Pemanfaatan Android Dalam Perancangan Aplikasi Kumpulan Doa," *J. Sains dan Inform.*, vol. 4, no. 1, pp. 54–65, 2018, doi: 10.22216/jsi.v4i1.3409.
- [11] Y. Yudhanto and A. Wijayanto, "Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio," 2017. https://books.google.co.id/books?hl=id&lr=&id=98phDwAAQBAJ&oi=fnd&pg=PP1&dq=android+aplikasi&ots=OG7eYau6Zz&sig=KEZt7WGFNeHLLu8zJrJvfMkqXJw&redir_esc=y#v=onepage&q=android+aplikasi&f=false (accessed May 08, 2022).
- [12] P. S. Dewi and S. Sintaro, "Mathematics Edutainment Dalam Bentuk Aplikasi Android," *Triple S*, vol. 2, no. 1, pp. 1–11, 2019.
- [13] W. Wiryosukiro, "Belajar Bahasa Pemrograman Dart Untuk Pemula – Part 1," *Emka.Web.Id*, no. 1, pp. 2–5, 2019, [Online]. Available: <https://emka.web.id/tutorial/2019/belajar-bahasa-pemrograman-dart-untuk-pemula-part-1/>
- [14] W. Wu, "Flutter Vs React," no. March, 2018, [Online]. Available: <https://www.theseus.fi/bitstream/handle/10024/146232/thesis.pdf?sequence=1>
- [15] H. Fujita and E. Herrera-Viedma, "New Trends in Intelligent Software Methodologies, Tools and Techniques ... - Google Buku," 2018. https://books.google.co.id/books?hl=id&lr=&id=dl1xDwAAQBAJ&oi=fnd&pg=PA3&dq=sqlite+database&ots=uChSMHxip9&sig=y_yW4LstgnFiND-j7qie2wVK-XE&redir_esc=y#v=onepage&q=sqlite+database&f=false (accessed May 11, 2022).
- [16] A. Supriyatna, "Metode Extreme Programming Pada Pembangunan Web Aplikasi Seleksi Peserta Pelatihan Kerja," *J. Tek. Inform.*, vol. 11, no. 1, pp. 1–18, 2018, doi: 10.15408/jti.v11i1.6628.
- [17] E. Arribe, Aryanto, and R. Asrianto, "Aplikasi E-Marketplace Menggunakan Arsitektur Mvvm (Model-View-Viewmodel) Berbasis Android," *J. Fasilkom*, vol. 11, no. 2, pp. 75–78, 2021, doi: 10.37859/jf.v11i2.2762.