



## **Pengembangan Aplikasi Mobile Muhasabah Untuk Evaluasi Diri Harian Berbasis Kotlin Android Dengan Pendekatan *Self-Assessment***

**Raafi Hilmi, Mulia Rahmayu\***

Teknik dan Informatika, Teknologi Informasi, Universitas Bina Sarana Informatika, DKI Jakarta, Indonesia

Email: <sup>1</sup>raafihilmi90@gmail.com, <sup>2,\*</sup>mulia.mlh@bsi.ac.id

(\* : coresponding author)

**Abstrak**—Muhasabah atau refleksi diri merupakan proses penting dalam membentuk kesadaran diri dan meningkatkan kualitas hidup seseorang, namun praktiknya belum menjadi kebiasaan sehari-hari di kalangan masyarakat, khususnya generasi muda. Kurangnya kesadaran, keterbatasan waktu, serta belum tersedianya media digital yang mendukung proses muhasabah secara praktis menjadi alasan utama. Penelitian ini bertujuan untuk merancang aplikasi Muhasabah Harian berbasis Android yang mampu memfasilitasi pengguna untuk melakukan evaluasi diri secara mandiri melalui pendekatan *self-assessment*. Metode pengembangan yang digunakan adalah model *Waterfall*, dengan tahapan analisis kebutuhan, desain, implementasi, pengujian, dan dukungan. Teknik pengumpulan data dilakukan melalui studi pustaka, observasi terhadap aplikasi serupa, serta wawancara dengan guru dan siswa di SDN Panunggangan 2 Pinang. Sistem dirancang dengan arsitektur MVVM menggunakan *Jetpack Compose* dan bahasa pemrograman *Kotlin*. Aplikasi mendukung dua mode akses, yaitu tamu (*offline*) dengan penyimpanan lokal melalui *Room Database* dan pengguna terdaftar (*online*) dengan penyimpanan di *Firebase Firestore*. Hasil dari penelitian ini adalah sebuah aplikasi yang memiliki fitur pengisian refleksi, riwayat evaluasi, grafik perkembangan, dan notifikasi pengingat muhasabah. Aplikasi ini diharapkan dapat menjadi media refleksi yang praktis dan efektif.

**Kata Kunci:** Muhasabah; Aplikasi Android; Self-Assessment; Firebase; Room Database

**Abstract**—*Self-reflection (muhasabah) is crucial for fostering self-awareness and improving one's quality of life, yet it has not become a daily habit—particularly among younger generations. Limited awareness, time constraints, and the lack of practical digital tools are key obstacles. This study aims to design an Android-based "Muhasabah Harian" application that enables users to conduct independent self-evaluation through a self-assessment approach. Development follows the Waterfall model, covering requirement analysis, design, implementation, testing, and support. Data were gathered through literature review, observation of similar applications, and interviews with teachers and students at SDN Panunggangan 2 Pinang. The system adopts the MVVM architecture, Jetpack Compose, and Kotlin. The application offers two modes: guest (offline) with local storage via Room Database and registered user (online) with storage in Firebase Firestore. Key features include reflection entry, evaluation history, progress graphs, and reminder notifications. The resulting application is expected to provide a practical and effective medium for daily self-reflection.*

**Keywords:** *Muhasabah; Android Application; Self-Assessment; Firebase; Room Database*

### **1. PENDAHULUAN**

Perkembangan teknologi digital dan peningkatan akses internet di Indonesia menunjukkan pertumbuhan yang sangat pesat dalam sepuluh tahun terakhir. Kondisi ini menjadikan internet sebagai bagian yang tidak terpisahkan dari kehidupan remaja, baik untuk kebutuhan komunikasi, pendidikan, maupun hiburan. Namun, kemudahan akses tersebut juga membawa konsekuensi berupa munculnya perilaku adiktif terhadap internet. Dampak adiksi internet tidak hanya bersifat individual, tetapi juga memengaruhi berbagai aspek kehidupan. Pada ranah akademik, adiksi internet menyebabkan prokrastinasi dan penurunan prestasi belajar. Secara psikologis, remaja rentan mengalami kecemasan dan depresi. Sedangkan pada aspek sosial muncul kecenderungan menarik diri dari lingkungan serta konflik interpersonal yang berkaitan dengan perubahan konektivitas fungsi otak pada remaja adiksi internet [1]. Salah satu metode Pemeliharaan Kesehatan Mental menurut islam adalah dengan menggunakan metode Psikoterapi Sufistik yang didalamnya terdapat berbagai macam terapi dan salah satunya adalah terapi muhasabah [2]. Dengan mengacu pada perkembangan psikososial tersebut maka seorang remaja awal yang mampu menghadapi dan memenuhi tugas perkembangannya akan mengalami suatu bentuk keakraban dan kemudahan dalam membentuk attachment dengan berbagai aspek kehidupannya [3].

Muhasabah diri adalah koreksi terhadap kesalahan, kelemahan, perbuatan, sikap dan peninjauan sejenisnya pada diri sendiri atau sebuah cara yang biasa digunakan pada proses pemulihan atau penyembuhan atau terapi dalam prosesnya dilakukan oleh sendiri mengikuti nalurinya. *Muhasabah diri dalam perspektif hadis* bagi jiwa dan kesehatan mental sangat baik dilakukan karena cara ini adalah cara yang alami. Proses muhasabah diri meliputi hati yang tenang dan nyaman atau bahkan sesuai apa yang Rasulullah ajarkan seperti Sholat, Dzikirullah, mengasingkan diri (*taffakur*) atau juga banyak ibadah lainnya [4]. Terapi muhasabah adalah suatu bentuk refleksi diri yang melibatkan pemeriksaan pikiran, perasaan, dan perilaku seseorang untuk mendapatkan wawasan tentang pola pribadi dan mengidentifikasi area yang perlu diperbaiki. Bentuk terapi ini berakar pada filsafat Islam dan dapat diterapkan pada berbagai aspek kehidupan, termasuk *self-motivation* [5]. Muhasabah juga memainkan peran penting dalam pengembangan karakter. Dengan secara rutin mengevaluasi diri, seseorang dapat menjadi lebih jujur, rendah hati, dan bertanggung jawab. Ini membantu dalam menghindari kesalahan yang sama di masa depan dan mendorong kemajuan pribadi serta spiritual [6].



Namun dalam kenyataannya, tidak semua orang mampu melakukan muhasabah secara rutin. Kehidupan yang semakin dinamis, terutama di kalangan generasi muda, sering kali membuat aktivitas introspeksi ini terlupakan. Aktivitas yang padat, gaya hidup cepat, serta kurangnya kesadaran untuk mengevaluasi diri menyebabkan banyak orang terjebak dalam rutinitas tanpa pernah meninjau kembali apakah apa yang mereka lakukan sudah sesuai dengan nilai-nilai kebaikan. Hal ini sejalan dengan pendapat [7], yang menyatakan bahwa generasi muda mengalami kesulitan dalam melaksanakan refleksi diri secara konsisten.

Lebih jauh lagi, masyarakat Indonesia khususnya generasi muda, sangat akrab dengan penggunaan perangkat digital dalam aktivitas sehari-hari. Akan tetapi, belum banyak ditemukan aplikasi yang memanfaatkan pendekatan teknologi tersebut sebagai sarana dalam menanamkan nilai-nilai pembentukan jati diri dan pengembangan kepribadian melalui praktik muhasabah [8] telah menunjukkan bahwa penggunaan aplikasi Android bisa membantu masyarakat dalam mengevaluasi diri secara efisien dan efektif, khususnya dalam konteks kesehatan. Konsep serupa dapat diterapkan dalam konteks muhasabah diri agar lebih mudah diakses oleh masyarakat luas.

Android adalah sistem operasi berbasis *Linux* kernel yang dirancang untuk mendukung perangkat *mobile* dengan antarmuka layar sentuh, contohnya *smartphone* dan tablet [9]. Android menawarkan lingkungan pengembangan terbuka yang memungkinkan pengembangan untuk menciptakan berbagai aplikasi secara fleksibel dan inovatif. Sifat *open-source* dari Android memungkinkan pengembangan melakukan eksplorasi dan modifikasi dengan mudah, serta mendapat dukungan luas dari komunitas global. Hal ini membuat Android menjadi pilihan populer dalam pengembangan aplikasi *mobile*, termasuk dalam penelitian ini yang bertujuan untuk merancang aplikasi muhasabah diri berbasis Android.

Pada penelitian terdahulu yang dilakukan oleh Musfichin tahun 2023 menghasilkan dua studi, hasil studi pertama diperoleh pengertian bahwa muhasabah merupakan kesadaran individu untuk menghitung dan mengoreksi akan aktifitas diri yang telah dan akan dijalani untuk tujuan yang lebih baik. Sementara hasil studi kedua menunjukkan bahwa muhasabah memiliki tiga dimensi antara lain: Kesadaran diri, Menghitung Diri, dan Mengoreksi Diri. Implikasi penelitian ini dibahas dalam konteks pengembangan psikologi islam [10]. Penerapan metode Muhasabah An-Nafs dilaksanakan dengan menerapkan beberapa tahapan yakni, merancang tahapan-tahapan akan diimplementasikan dalam pelaksanaannya dan disesuaikan dengan tahapan akan lazim terjadi dengan metode Muhasabah An-Nafs. Adapun tahapan-tahapan tersebut adalah Takhalli, Tahalli dan Tajalli [11]

Pada penelitian yang dilakukan oleh Maretta pada tahun 2024, konseling berbasis muhasabah cukup efektif dalam untuk menurunkan perilaku menyimpang seperti membolos, merokok, dan bullying, serta meningkatkan aspek spiritualitas, kedisiplinan, dan empati siswa. Temuan ini mendukung pentingnya integrasi muhasabah dalam strategi konseling untuk meningkatkan pembentukan karakter dan perilaku positif di lingkungan sekolah [12]. Penelitian lainnya yang dilakukan oleh Guntoro dan Ardiansyah menjelaskan bahwa peningkatan kemampuan diri berawal dari upaya manusia untuk melakukan evaluasi atau muhasabah terhadap amal dan aktivitas yang dilakukan. Muhasabah juga dapat mempengaruhi fluktuasi keimanan dalam beribadah dengan evaluasi ibadah dapat digunakan sebagai pendukung manusia untuk membantu mengontrol dan mengelola ibadah dalam memotivasi umat muslim dengan cara meriwayatkan ibadah harian yang telah dilakukan. Hasil dari penelitian ini adalah sebuah aplikasi *Mutaba'ah* yang diharapkan dapat memotifasi pengguna dalam meningkatkan ibadah dengan melihat grafik ibadah, riwayat ibadah dan melihat persentase pengguna pada grup yang tersedia [13].

Dalam merepresentasikan perangkat lunak sistem yang dibuat menggunakan UML. UML memiliki berbagai jenis diagram yang dikelompokkan menjadi dua klasifikasi utama, yakni diagram struktur (*structural diagrams*) dan diagram perilaku (*behaviour diagrams*). Menurut [14] Dalam perancangan aplikasi ini, digunakan beberapa jenis diagram yang termasuk ke dalam kedua kategori tersebut, Penerapan UML memiliki kontribusi besar terhadap dokumentasi teknis serta proses pengembangan sistem yang lebih terstruktur. Dengan penerapan diagram UML yang konsisten, pengembang dapat mengurangi risiko kesalahan selama tahap implementasi, karena setiap kebutuhan dan proses sistem telah tergambar secara jelas sejak tahap perancangan [15].

Penelitian ini terfokus pada tahap perancangan serta pengembangan aplikasi Muhasabah diri yang dibangun menggunakan platform *Android* dengan pendekatan *self-assessment*. Dalam penelitian yang dilakukan oleh [16] disebutkan bahwa *self-assessment* memberikan kesempatan kepada siswa untuk menilai dan mengevaluasi kualitas belajar secara mandiri, sehingga mereka dapat secara langsung mengidentifikasi kelemahan dan kekuatan dalam pembelajaran, serta meningkatkan kemampuan untuk mengambil keputusan terkait perbaikan diri secara terus-menerus. Metode pengembangan yang digunakan adalah model *Waterfall*. Metode *Waterfall* terbagi ke dalam lima tahapan utama, yaitu: tahap analisis, perancangan atau desain, pelaksanaan atau implementasi, pengujian, hingga pemeliharaan. Metodologi tersebut cocok digunakan dalam pengembangan sistem yang sudah memiliki kebutuhan yang jelas di awal, karena alurnya memungkinkan dokumentasi dan control yang lebih baik dalam setiap tahapannya [17].

Teknik pengumpulan data dilakukan melalui studi pustaka, observasi terhadap aplikasi serupa, serta wawancara dengan guru dan siswa di SDN Panunggangan 2 Pinang. Aplikasi ini ditujukan untuk pengguna umum yang ingin melakukan evaluasi diri secara mandiri melalui perangkat *mobile*. Fitur yang dikembangkan dalam aplikasi mencakup pencatatan refleksi harian, penilaian terhadap diri sendiri berdasarkan indikator tertentu, serta pelacakan perkembangan evaluasi dari waktu ke waktu. Aplikasi akan dibangun menggunakan bahasa pemrograman *Kotlin* melalui *Android Studio*, serta memanfaatkan *Firebase* sebagai media autentikasi pengguna dan penyimpanan data berbasis *cloud*. *Firebase* sangat efektif digunakan untuk aplikasi yang membutuhkan interaksi data secara langsung dengan pengguna,

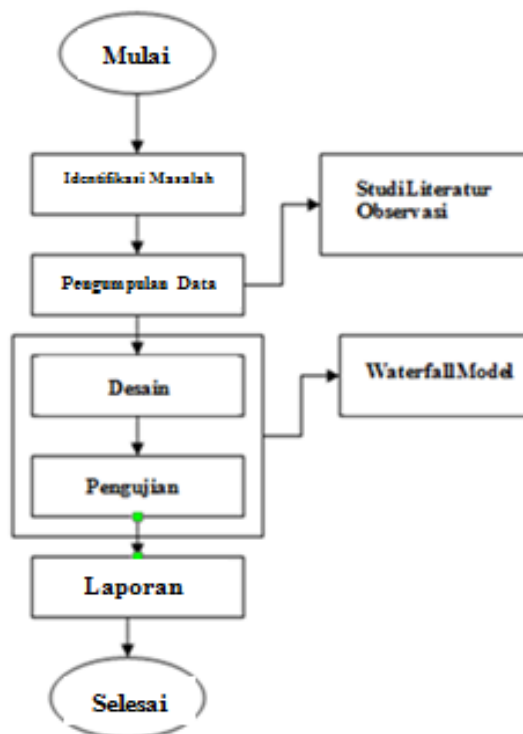
karena kecepatan dan kemudahan dalam sinkronisasi data yang disediakannya [18]. Selain layanan penyimpanan data, *Firebase* juga menawarkan fitur autentikasi pengguna yang mendukung proses login melalui berbagai metode, seperti email, Google, dan media sosial lainnya. *Firebase Authentication* memberikan kemudahan kepada pengembang dalam proses pengelolaan identitas pengguna tanpa perlu membangun mekanisme keamanan sendiri. Penggunaan *Firebase* terbukti dapat mempercepat proses pengembangan aplikasi dan meningkatkan keamanan data pengguna, sehingga sangat cocok untuk aplikasi muhasabah yang memerlukan akses aman dan sinkronisasi data secara langsung [19].

Aplikasi ini hanya dikembangkan untuk sistem operasi *Android* dan tidak mencakup platform *iOS* maupun web. Selain itu, aplikasi tidak menyediakan fitur interaksi antar pengguna, konsultasi psikologis, maupun analisis berbasis kecerdasan buatan. Seluruh proses muhasabah dilakukan secara pribadi oleh pengguna dan ditujukan sebagai alat bantu individu, bukan sebagai sarana evaluasi oleh pihak eksternal. Tujuan dari penelitian ini yaitu membangun suatu aplikasi Muhasabah dalam Bahasa Indonesia yang tidak hanya berfungsi sebagai alat bantu spiritual, aplikasi ini diharapkan juga sebagai media untuk membangun kesadaran diri, memperbaiki akhlak, meningkatkan produktivitas, dan membentuk pribadi yang lebih baik secara menyeluruh. Sejalan dengan hasil penelitian [20], penerapan metode asesmen diri (*self-assessment*) pada pembelajaran terbukti mampu mendorong peserta didik untuk lebih menyadari kekuatan dan kelemahan dirinya sendiri, meningkatkan kepercayaan diri, serta menumbuhkan sikap jujur dan objektif dalam melakukan evaluasi terhadap diri. Hal ini menunjukkan bahwa integrasi metode *self-assesment* dalam media digital seperti aplikasi muhasabah berpotensi menjadi sarana efektif untuk membantu individu dalam melakukan refleksi diri secara mandiri dan berkelanjutan.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Dalam melakukan penelitian ini, ada beberapa tahapan yang dilakukan antara lain yang terlihat di gambar 1 berikut ini:



**Gambar 1.** Tahapan Penelitian

Adapun penjelasan dari gambar 1 tahapan penelitian ini adalah:

- a. Identifikasi Masalah  
Penelitian dasarnya diawali dengan melakukan analisis dan identifikasi masalah yang terjadi pada masyarakat khususnya remaja dalam penggunaan Muhasabah diri, dan bagaimana cara pemecahan masalah yang terjadi sehingga penggunaan pada Aplikasi Muhasabah tersebut dapat berfungsi dengan baik.
- b. Pengumpulan data  
Pada tahapan kali ini yaitu metode pengumpulan data yang dilakukan dengan cara studi literatur dan observasi, studi literatur untuk mencari berbagai referensi yang berkaitan dengan penelitian yang dilakukan dan observasi dengan mengamati beberapa aplikasi serupa yang telah ada di platform Google Play Store, baik dari segi fitur, tampilan, maupun kekurangan yang dapat dikembangkan.



### c. *Waterfall* Model

*Waterfall* adalah metode atau model pengembangan perangkat lunak yang dipilih dalam penelitian ini, yakni suatu pendekatan berurutan yang melibatkan sejumlah tahapan utama di antaranya:

#### 1. Analisa Kebutuhan Software

Tahapan ini dilakukan untuk mengidentifikasi kebutuhan sistem yang akan dibangun. Analisis dilakukan berdasarkan hasil studi pustaka, observasi terhadap aplikasi sejenis, serta wawancara dengan pihak sekolah. Dari hasil analisis ini diperoleh informasi mengenai fitur-fitur penting yang diperlukan dalam aplikasi muhasabah diri, seperti pencatatan refleksi harian, indikator evaluasi, dan penyimpanan histori evaluasi.

#### 2. Desain

Tahapan desain meliputi perancangan antarmuka pengguna (*user interface*), perancangan struktur data, serta alur interaksi pengguna dengan sistem. Desain dilakukan menggunakan bantuan diagram UML seperti *Use Case*, *Activity*, dan *Sequence Diagram*. Desain ini menjadi acuan dalam proses implementasi aplikasi agar sesuai dengan kebutuhan yang telah dijelaskan.

#### 3. Implementasi

Pembuatan aplikasi dilakukan melalui pemanfaatan perangkat lunak *Android Studio* dan dikembangkan menggunakan bahasa pemrograman *Kotlin* yang sesuai untuk platform *Android*.

#### 4. Pengujian

Setelah aplikasi selesai dibuat, dilakukan pengujian untuk memastikan bahwa seluruh fitur berjalan sesuai dengan fungsinya. Pengujian dilakukan secara manual dengan mencoba setiap fitur utama dan mencatat hasilnya. Jika ditemukan kesalahan (bug), akan dilakukan perbaikan sebelum aplikasi dinyatakan siap digunakan.

#### 5. Pemeliharaan

Tahap ini merupakan bentuk dukungan terhadap aplikasi yang telah dibangun, baik dalam bentuk perawatan, perbaikan jika terjadi kesalahan, maupun pengembangan lebih lanjut sesuai kebutuhan pengguna. Tahap ini dilakukan jika aplikasi akan terus dikembangkan secara berkelanjutan.

### d. Laporan

Tahap ini merupakan tahap pengumpulan semua hasil rancangan sistem yang telah dilakukan dan dijadikan menjadi laporan penelitian.

## 3. ANALISA DAN PEMBAHASAN

Pada penelitian ini *Waterfall* adalah metode atau model pengembangan perangkat lunak yang dipilih dalam penelitian ini, yakni suatu pendekatan berurutan yang melibatkan sejumlah tahapan utama di antaranya:

### 3.1 Analisis Kebutuhan

Analisis kebutuhan dibutuhkan supaya dapat merumuskan fitur dan elemen penting yang harus tersedia dalam aplikasi muhasabah diri berbasis *Android*. Kebutuhan ini terdiri atas dua kategori utama, yaitu:

#### a. Kebutuhan Fungsional

Berikut kebutuhan fungsional yang dirumuskan yaitu akses dengan akun google, akses sebagai tamu, halaman beranda, fitur muhasabah harian, riwayat dan grafik perkembangan, motivasi dan kutipan harian dan notifikasi pengingat.

#### b. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional menjelaskan karakteristik teknis, operasional, dan kualitas aplikasi yang perlu dipenuhi dalam perancangan dan pengembangan aplikasi. Kebutuhan ini mencakup platform android, tampilan antarmuka yang sederhana, bahasa indonesia, keamanan data, konektivitas internet (opsional), dan ukuran aplikasi ringan

### 3.2 Desain

Perancangan basis data dalam aplikasi Muhasabah Harian menggunakan pendekatan hybrid, yaitu menggabungkan penyimpanan online melalui *Firebase Firestore* untuk pengguna login (akun Google) dan *offline* melalui *Room (SQLite)* untuk pengguna tamu (*anonymous login*). Pendekatan ini memungkinkan aplikasi berfungsi dalam kondisi daring maupun luring dengan tetap menjaga kelengkapan dan integritas data.

#### a. Struktur Database

Basis data aplikasi terdiri dari tabel-tabel berikut:

##### 1. Tabel Muhasabah

Tabel Muhasabah digunakan untuk menyimpan data Muhasabah diri dari pengguna. Tabel ini terdiri dari:

Nama Database	: Muhasabah_Diri
Nama Tabel	: muhasabah
Primary Key	: id
Jumlah Field	: 9

**Tabel 1.** Muhasabah

No	Nama Kolom	Type Data	Constraint	Keterangan
1	<i>id</i>	<i>INTEGER</i>	<i>PRIMARY KEY</i>	ID unik lokal (auto increment)
4	<i>type</i>	<i>TEXT</i>		Jenis muhasabah
5	<i>activity</i>	<i>TEXT</i>		Deskripsi aktivitas
6	<i>obstacle</i>	<i>TEXT</i>		Kendala yang dihadapi
7	<i>note</i>	<i>TEXT</i>		Catatan tambahan
8	<i>mood</i>	<i>TEXT</i>		Suasana hati
9	<i>score</i>	<i>INTEGER</i>		Nilai penilaian (1-10)

2. Tabel Pengguna

Tabel pengguna digunakan untuk menyimpan pengguna. Tabel ini terdiri dari:

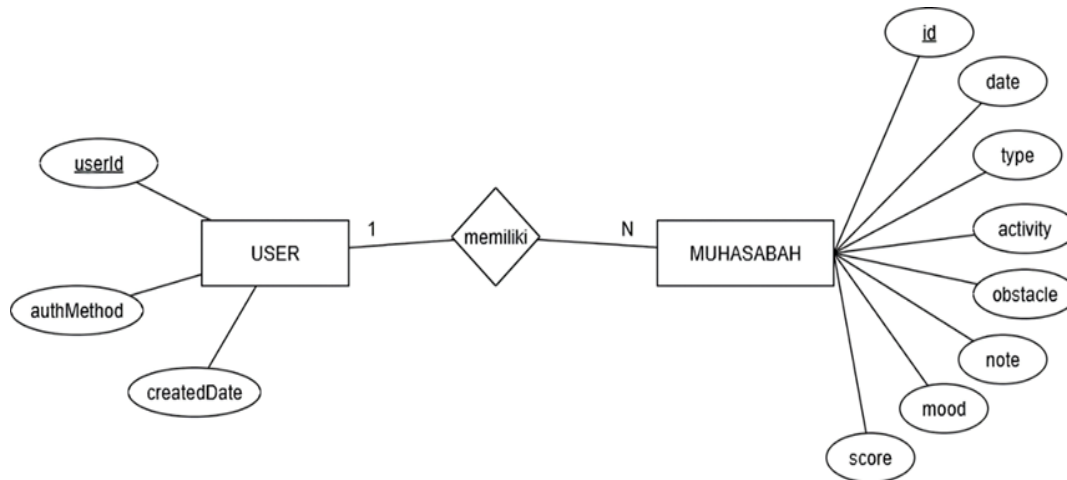
- Nama Database : Muhasabah\_Diri
- Nama Tabel : pengguna
- Primary Key : *userId*
- Jumlah Field : 3

**Tabel 1.** Pengguna (*Firestore Authentication*)

No	Nama Kolom	Type Data	Keterangan
1	<i>userId</i>	<i>String</i>	ID unik pengguna ( <i>Primary Key</i> )
2	<i>authMethod</i>	<i>String</i>	Metode autentikasi ( <i>anonymous/google</i> )
3	<i>createdDate</i>	<i>Timestamp</i>	Tanggal pembuatan akun

b. *Entity Relationship Diagram (ERD)*

Struktur relasi antar entitas pada basis data *Firestore (Online)* dan *Room (Offline)* digambarkan dalam bentuk ERD dibawah ini:

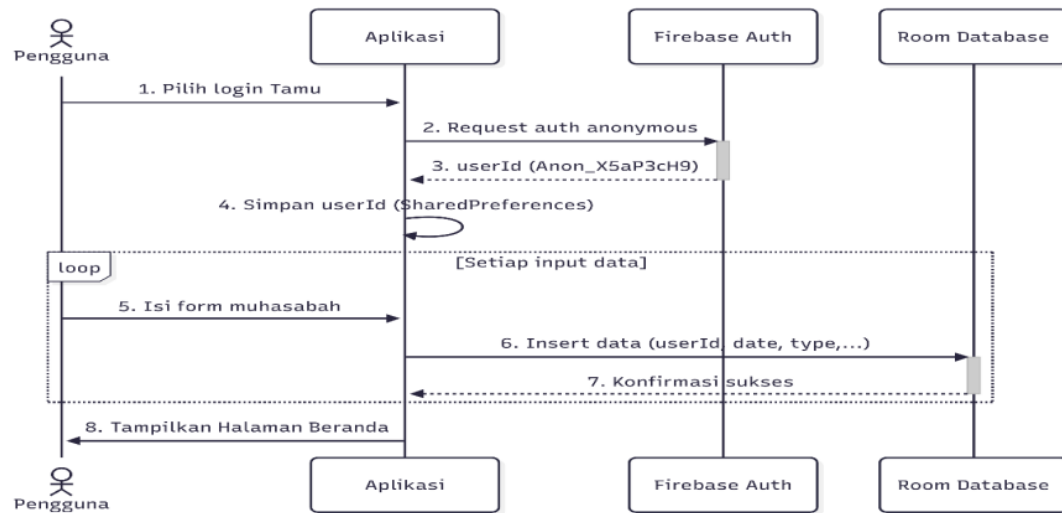


**Gambar 2.** *Entity-Relationship Diagram (ERD)*

Pada gambar 2 menjelaskan setiap entitas *USER* dapat memiliki banyak entitas *MUHASABAH*, tetapi satu entitas *MUHASABAH* hanya dimiliki oleh satu *USER*. Hal ini tergambar dari notasi kardinalitas “1” pada sisi *USER* dan “N” pada sisi *MUHASABAH*. Atribut *userId* yang terdapat pada entitas *MUHASABAH* berperan sebagai *foreign key*, yang merujuk pada atribut *userId* di entitas *USER*. Dengan demikian, setiap data catatan muhasabah selalu terasosiasi dengan pengguna tertentu yang mencatatkannya.

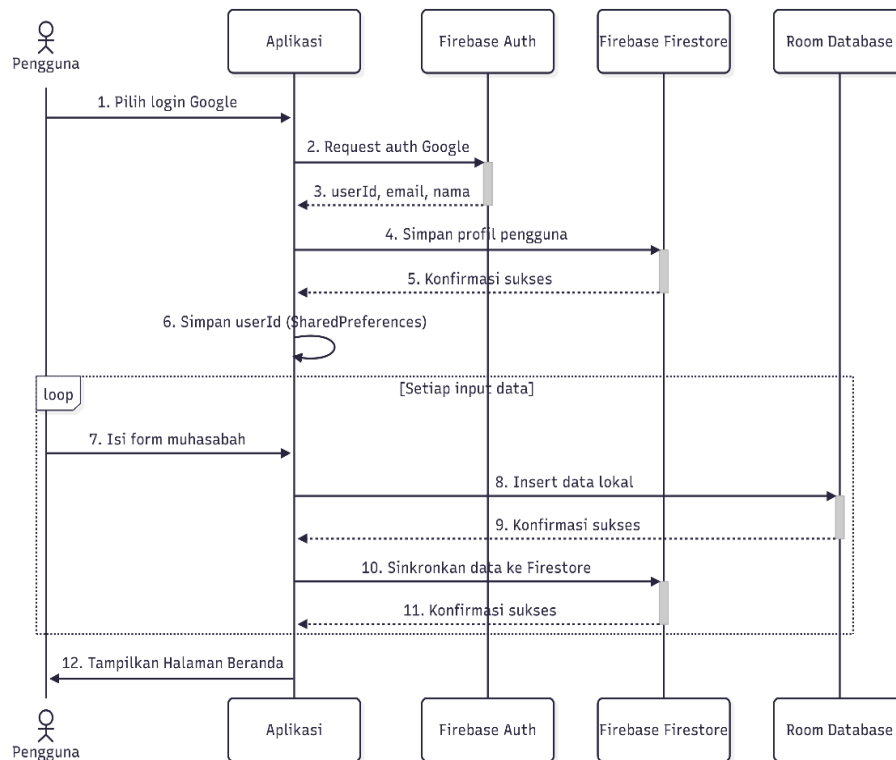
c. *Sequence Diagram*

Pada gambar 3 menjelaskan ketika pengguna memilih untuk masuk sebagai tamu, aplikasi secara otomatis melakukan permintaan autentikasi sebagai anonim ke *Firestore*. Setelah permintaan diproses, *Firestore* akan mengembalikan ID pengguna unik yang khusus untuk sesi tamu tersebut. ID pengguna ini kemudian disimpan secara lokal di perangkat pengguna sebagai referensi. Selanjutnya, setiap data muhasabah yang dimasukkan atau diolah oleh pengguna akan disimpan secara eksklusif di dalam database lokal menggunakan framework *Room*, tanpa dikirimkan ke server eksternal. Terakhir, pengguna akan diarahkan ke aplikasi menuju halaman beranda.



**Gambar 3.** Sequence Diagram Login Tamu dan Penyimpanan Lokal

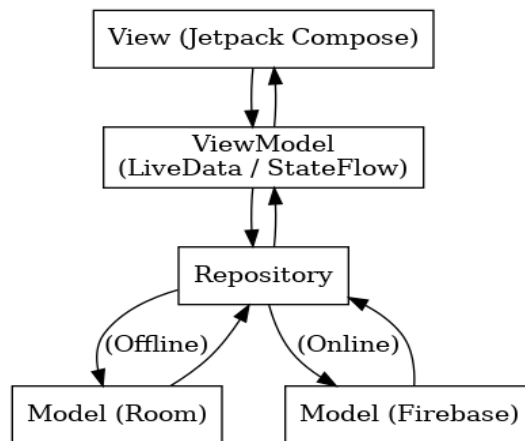
Sedangkan pada gambar 4 menjelaskan ketika pengguna memilih masuk menggunakan akun *Google*, aplikasi akan menginisiasi proses autentikasi melalui *Firebase*. Sistem kemudian memverifikasi kredensial *Google* pengguna dan *Firebase* mengembalikan data identitas yang mencakup ID pengguna unik, alamat email, serta nama lengkap. Data profil ini langsung disimpan di *Firestore* (database cloud *Firebase*), dan aplikasi menerima konfirmasi ketika penyimpanan profil berhasil dilakukan. Secara paralel, ID pengguna disimpan secara lokal di perangkat untuk referensi cepat. Selanjutnya, setiap data muhasabah yang diinput pengguna pertama kali disimpan di database lokal perangkat (menggunakan *Room*) untuk menjamin akses offline. Setelah tersimpan secara lokal, data tersebut segera disinkronkan ke *Firestore* untuk *backup cloud* dan akses multi-perangkat. Terakhir, pengguna diarahkan ke halaman beranda setelah berhasil disimpan baik secara lokal maupun di *cloud*.



**Gambar 4.** Sequence Diagram Login Google dan Sinkronisasi Firestore

*d. Software Architecture*

Perancangan arsitektur perangkat lunak dalam pengembangan aplikasi muhasabah diri ini menggunakan pola *Model-View-ViewModel (MVVM)* yang disesuaikan dengan kerangka kerja *Jetpack Compose* di Android. Pola ini dipilih karena memberikan pemisahan yang jelas antara logika presentasi dan logika bisnis, serta mendukung penerapan arsitektur modular dan skalabel.



**Gambar 5.** Arsitektur Aplikasi Muhasabah Diri

Pada gambar 5 menjelaskan bahwa secara umum, arsitektur MVVM terdiri atas tiga komponen utama, yaitu:

a. Model

Model berfungsi sebagai representasi data dan logika bisnis aplikasi. Pada aplikasi ini, model mencakup data hasil evaluasi pengguna, entitas *Room Database* untuk mode *offline*, serta interaksi dengan *Firestore* untuk mode *online*. Termasuk dalam model adalah entitas, *Data Access Objects (DAO)*, serta kelas helper yang memproses total skor evaluasi.

b. View

*View* merupakan bagian antarmuka pengguna (*user interface*) yang ditampilkan di perangkat. Dalam aplikasi ini, *View* dibangun menggunakan *Jetpack Compose* yang memungkinkan pengembangan UI secara deklaratif, efisien, dan responsif. *View* akan merepresentasikan halaman beranda, formulir pertanyaan, statistik, dan riwayat muhasabah.

c. ViewModel

*ViewModel* bertugas sebagai penghubung antara *View* dan *Model*. Komponen ini menerima input dari *View*, memproses data melalui model, dan mengembalikan hasilnya ke *View*. *ViewModel* juga menyimpan state agar antarmuka tidak kehilangan data saat terjadi rotasi layar atau perubahan konfigurasi.

Aplikasi mendukung dua skenario penggunaan:

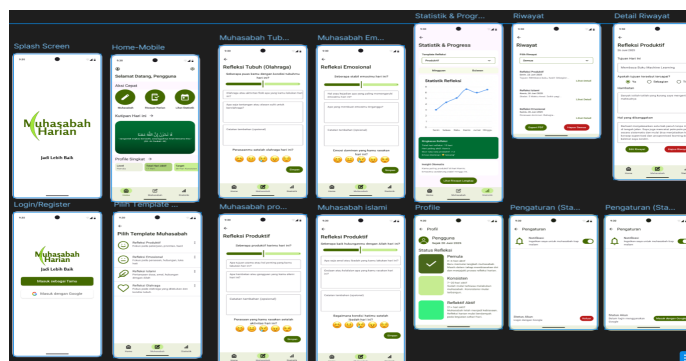
a. *Online*, untuk pengguna login dengan Google, data akan disimpan di *Firestore Authentication* dan *Firestore Realtime Database*.

b. *Offline*, untuk pengguna tamu, data akan disimpan secara lokal menggunakan *Room (ORM untuk SQLite)*

## 4. IMPLEMENTASI

### 4.1 Hasil Perancangan

Perancangan antarmuka pengguna (*User Interface*) merupakan tahapan penting dalam pengembangan aplikasi, karena berkaitan langsung dengan kenyamanan dan kemudahan pengguna dalam berinteraksi dengan sistem. Antarmuka yang baik harus mampu menyampaikan fungsi aplikasi secara intuitif dan menarik tanpa membingungkan pengguna. Pada aplikasi muhasabah ini, desain antarmuka dirancang menggunakan perangkat lunak *Figma*, dengan memakai *Material Design 3* sebagai acuan dalam desain sistemnya yang umum digunakan pada aplikasi Android. Fokus utama perancangan adalah kesederhanaan navigasi, konsistensi visual, dan kemudahan akses oleh semua kalangan usia. Adapun tampilan *User Interface* dari aplikasi Muhasabah sebagai berikut :



**Gambar 6.** Rancangan Aplikasi Muhasabah Harijan

Pada gambar 6 menjelaskan rancangan aplikasi Muhasabah tampilan awal aplikasi menampilkan *splash screen* sederhana dengan logo “Muhasabah Harian” di tengah layer. Setelah itu, pengguna diarahkan ke halaman *login* dengan dua pilihan akses yaitu sebagai tamu (*offline*) atau *login* dengan *Google* (*online*). Kedua opsi disajikan secara jelas dan mudah dijangkau. Tampilan beranda menampilkan sambutan kepada pengguna sesuai mode akses, baik login maupun tamu. Tersedia tiga menu utama yaitu Muhasabah, Riwayat Harian, dan Statistik. Selain itu, ditampilkan kutipan harian dari ayat Al-Qur’an yang berubah secara otomatis setiap hari untuk menambah nilai reflektif dari aplikasi ini. Tampilan menu muhasabah memperlihatkan empat kategori refleksi yang dapat dipilih pengguna, yaitu Refleksi Produktif, Emosional, Islami, dan Olahraga. Setiap kategori dirancang untuk mengarahkan pengguna melakukan evaluasi harian sesuai dengan aspek kehidupan yang relevan, guna meningkatkan kesadaran diri dan kualitas aktivitas hariannya.

Tampilan riwayat evaluasi berfungsi untuk menampilkan seluruh data refleksi pengguna berdasarkan kategori yang dipilih. Pada bagian ini, pengguna juga dapat melakukan ekspor seluruh data ke dalam bentuk file PDF, serta menghapus seluruh catatan refleksi dengan satu tombol. Fitur konfirmasi disediakan untuk menghindari penghapusan yang tidak disengaja. Tampilan pengaturan pada aplikasi menampilkan pengelolaan notifikasi dan status akun pengguna. Gambar sebelah kiri memperlihatkan tampilan saat pengguna masuk menggunakan akun *Google*. Pada bagian bawah halaman, informasi akun seperti alamat email akan ditampilkan, dan tersedia tombol untuk keluar dari akun. Tombol saklar notifikasi dapat diaktifkan atau dinonaktifkan sesuai keinginan pengguna.

## 4.2 Pengujian

Pengujian pada aplikasi muhasabah harian dilakukan dengan pendekatan pengujian perangkat lunak yang melibatkan dua metode utama, yaitu *unit testing* dan *UI testing*. Pengujian ini bertujuan untuk memastikan bahwa fungsi-fungsi utama aplikasi bekerja sesuai dengan yang diharapkan, baik dari segi logika internal maupun tampilan antarmuka pengguna.

### 4.2.3 Unit Testing

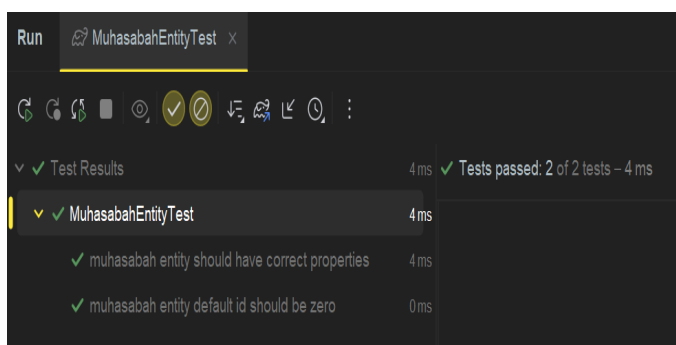
*Unit testing* dilakukan pada bagian terkecil dari kode, yaitu kelas atau fungsi yang berdiri sendiri, untuk memastikan bahwa setiap bagian menghasilkan output yang sesuai ketika diberikan input tertentu. Dalam aplikasi ini dilakukan tiga unit test, yaitu:

#### a. *MuhasabahEntityTest*

Pada gambar 7 pengujian dilakukan terhadap struktur data *MuhasabahEntity* untuk memastikan bahwa data yang dimasukkan ke dalam entitas sesuai dengan nilai-nilai atribut yang didefinisikan. Salah satu pengujian menguji apakah ID secara default bernilai 0 saat tidak ditentukan.

```
class MuhasabahEntityTest {  
  
    @Test  
    > fun `muhasabah entity should have correct properties`() {...}  
    |  
  
    @Test  
    fun `muhasabah entity default id should be zero`() {  
        val entity = MuhasabahEntity(  
            date = "2023-10-10",  
            type = "Harian",  
            activity = "Membaca",  
            obstacle = "Kurang konsentrasi",  
            note = "Harus lebih fokus",  
            mood = "Baik",  
            score = 80  
        )  
  
        assertEquals( expected: 0, entity.id)  
    }  
}
```

Gambar 7. MuhasabahEntityTest



Gambar 8. Hasil MuhasabahEntityTest

Pada gambar 8 menunjukkan hasil dari pengujian *MuhasabahEntity Test* yang menunjukkan kesesuaian hasil yang di dapat.

b. *RiwayatViewModelTest*

Pada gambar 9 pengujian ini memastikan bahwa *ViewModel RiwayatViewModel* dapat mengambil data dari repositori dan memetakannya menjadi list yang ditampilkan di layar riwayat. Pengujian juga memastikan bahwa nilai deskripsi dan tanggal dihasilkan dengan benar sesuai data *MuhasabahEntity*.

```
@OptIn(ExperimentalCoroutinesApi::class)
class RiwayatViewModelTest {

    private val testDispatcher = StandardTestDispatcher()
    private lateinit var repository: MuhasabahRepository
    private lateinit var viewModel: RiwayatViewModel

    @Before
    fun setup() {...}

    @After
    fun tearDown() {...}

    @Test
    fun `riwayat StateFlow emits mapped RiwayatItem list`() = runTest {...}
}
```

Gambar 9. RiwayatViewModelTest

```
✓ ✓ Test Results 1 sec 470ms
  ✓ ✓ RiwayatViewModelTest 1 sec 470ms
    ✓ riwayat StateFlow emits mapped RiwayatItem list 1 sec 470ms
```

Gambar 10. Hasil RiwayatViewModelTest

Pada gambar 10 menunjukkan hasil dari pengujian *RiwayatViewModel Test* yang menunjukkan kesesuaian hasil yang di dapat.

c. *MuhasabahViewModelTest*

Pada gambar 11 fokus pengujian berada pada proses penyimpanan refleksi (*saveReflection*) ke dalam repositori. Fungsi ini diuji apakah berhasil memanggil fungsi *insert* dengan data yang sesuai dan memicu callback *onSuccess*.

```
@Test
fun `saveReflection calls repository and onSuccess`() = runTest {
    var onSuccessCalled = false

    val answers = mapOf<String, Any>{
        "activity" to "Meditation",
        "obstacle" to "Distraction",
        "note" to "Felt calm",
        "mood" to "😊",
        "score" to 0.8f
    }

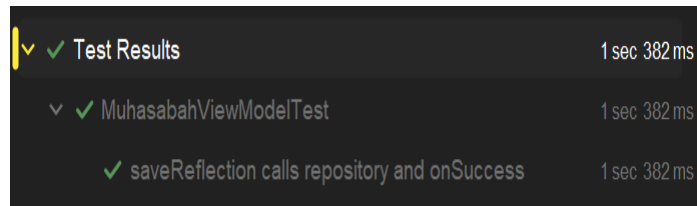
    viewModel.saveReflection(type: "Reflection", answers) {
        onSuccessCalled = true
    }

    testScheduler.advanceUntilIdle()

    coVerify {
        repository.insertReflection(match<MuhasabahEntity> {
            it.type == "Reflection" &&
            it.activity == "Meditation" &&
            it.obstacle == "Distraction" &&
            it.note == "Felt calm" &&
            it.mood == "😊" &&
            it.score == 8 // 0.8 * 10
        })
    }

    assert(onSuccessCalled)
}
```

Gambar 11. MuhasabahViewModelTest



**Gambar 12.** Hasil MuhasabahViewModelTest

Pada gambar 12 menunjukkan hasil dari pengujian MuhasabahViewModel Test yang menunjukkan kesesuaian hasil yang di dapat.

### 4.2.3 UI Testing

Pengujian tampilan pengguna dilakukan menggunakan framework *Jetpack Compose Test*, dengan memanfaatkan *ComposeTestRule*. Pengujian dilakukan terhadap *LoginScreen* sebagai pintu masuk utama aplikasi. Terdapat berbagai skenario pengujian yang dilakukan:

#### a. Tampilan UI

Pada gambar 13 memastikan bahwa semua elemen UI, seperti logo, tombol "Masuk sebagai Tamu", dan tombol "Masuk dengan Google", ditampilkan dengan benar.

```
@Test
fun loginScreen_displaysAllUIElements() {
    composeTestRule.setContent {
        LoginScreen()
    }

    composeTestRule.onNodeWithContentDescription(label: "Muhasabah Harian Logo").assertIsDisplayed()
    composeTestRule.onNodeWithText(text: "Masuk sebagai Tamu").assertIsDisplayed()
    composeTestRule.onNodeWithText(text: "Masuk dengan Google").assertIsDisplayed()
    composeTestRule.onNodeWithContentDescription(label: "Google Logo").assertIsDisplayed()
}
```

**Gambar 13.** Test Tampilan UI Login

#### b. Login Sebagai Tamu

Pada gambar 14 menguji skenario ketika pengguna menekan tombol "Masuk sebagai Tamu", baik saat berhasil maupun gagal. Saat terjadi kegagalan *login* (misal karena koneksi internet), pesan kesalahan ditampilkan kepada pengguna.

```
@Test
fun loginScreen_guestLoginFailure_showsErrorMessage() = runTest {
    val errorMessage = "Network error"
    val exception = Exception(errorMessage)

    every {...} answers {
        lambda<Exception> -> Unit {}.invoke(exception)
    }

    composeTestRule.setContent {...}
    composeTestRule.onNodeWithText(text: "Masuk sebagai Tamu").performClick()
    composeTestRule.waitForIdle()
    composeTestRule.onNodeWithText(text: "Login tamu gagal: $errorMessage").assertIsDisplayed()
}
```

**Gambar 14.** Test Masuk Sebagai Tamu

#### c. Login dengan Google

Pada gambar 15 menguji keberhasilan *login Google* dan kegagalan ketika objek *FirebaseUser* bernilai *null*. Pengujian juga memastikan *callback* berhasil terpanggil saat proses *login* berhasil.

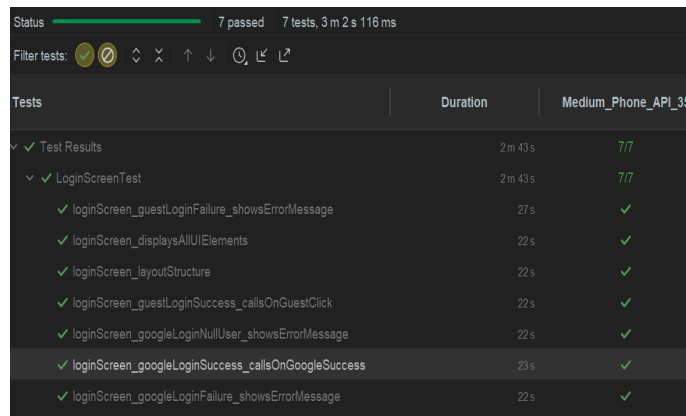
```
@Test
fun loginScreen_googleLoginSuccess_callsOnGoogleSuccess() = runTest {
    var googleSuccessCalled = false
    val mockUser = mockk<FirebaseUser>()

    coEvery { AuthManager.cacheUser(any(), any()) } just Runs
    every { ... } answers {
        lambda<FirebaseUser?> -> Unit().invoke(mockUser)
    }

    composeTestRule.setContent { ... }
    composeTestRule.onNodeWithText( text: "Masuk dengan Google").performClick()
    composeTestRule.waitForIdle()
    assert(googleSuccessCalled)
}
```

Gambar 15. Test Masuk dengan Google

Pada gambar 16 telah dilakukan pengujian yang menyeluruh baik dari sisi logika internal dan interaksi pengguna, aplikasi ini diharapkan dapat berjalan dengan stabil, memenuhi fungsionalitas yang dirancang, dan memberikan pengalaman pengguna yang optimal



Tests	Duration	Medium_Phone_API_35
Test Results	2 m 43 s	7/7
LoginScreenTest	2 m 43 s	7/7
loginScreen_guestLoginFailure_showsErrorMessage	27 s	✓
loginScreen_displaysAllUIElements	22 s	✓
loginScreen_layoutStructure	22 s	✓
loginScreen_guestLoginSuccess_callsOnGuestClick	22 s	✓
loginScreen_googleLoginNullUser_showsErrorMessage	22 s	✓
loginScreen_googleLoginSuccess_callsOnGoogleSuccess	23 s	✓
loginScreen_googleLoginFailure_showsErrorMessage	22 s	✓

Gambar 16. Hasil LoginScreenTest

## 5. KESIMPULAN

Perangkat lunak Muhasabah Harian ini berhasil dirancang dan dikembangkan menggunakan pendekatan hybrid yang menggabungkan penyimpanan data secara lokal (*Room*) dan *cloud* (*Firebase Firestore*). Hal ini memberikan keleluasaan kepada pengguna untuk tetap dapat menggunakan aplikasi meskipun tanpa koneksi internet. Implementasi arsitektur *Model-View-ViewModel* (MVVM) bersama dengan *framework Jetpack Compose* telah memudahkan proses pengembangan, pemeliharaan, serta meningkatkan keterpisahan antara logika aplikasi dan tampilan antarmuka. Pengujian perangkat lunak yang dilakukan, baik melalui unit testing maupun UI testing, mengindikasikan bahwa komponen utama program bekerja sesuai harapan dan sejalan dengan tujuan perancangan. Aplikasi telah diuji pada berbagai skenario login, baik sebagai tamu maupun dengan akun Google, serta telah dilakukan sinkronisasi data dengan Firebase untuk pengguna yang login dengan akun Google, dan penyimpanan lokal untuk pengguna tamu. Aplikasi Muhasabah Harian diharapkan dapat menjadi sarana bantu dalam proses evaluasi diri (muhasabah) yang lebih praktis, terstruktur, dan menyenangkan, serta dapat meningkatkan kesadaran pengguna terhadap pentingnya refleksi harian.

## REFERENCES

- [1] N. F. Ilahiya, H. Tajiri, and D. Sadiyah, "Efektivitas Muhasabah Sebagai Metode Konseling Sufistik Dalam Menurunkan Perilaku Adiktif Internet Pada Siswa SMKN 1 Ciamis," vol. 9, no. 3, pp. 86–96, 2026.
- [2] Yumna, G. K. Wardana, H. Fauziah, Muslikhin, S. S. Bachtiar, and E. Novitasari, "Implementasi Terapi Muhasabah Sebagai upaya Memelihara Kesehatan Mental dalam Usaha Penanggulangan Adiksi Narkoba (Studi Lapangan Lapas Narkotika Cirebon)," *Syifa al-Qulub J. Stud. Psikoterapi Sufistik*, vol. 6, no. 1, pp. 95–112, 2021.
- [3] Ulya, "Minaret Journal of Religious Studies," *Minaret J. Relig. Stud.*, vol. 1 (02), pp. 111–126, 2023.
- [4] Asmara Wijaya and Repa Hudan Lisalam, "Muhasabah Diri dalam Perspektif Hadis," *Ikhlash J. Ilm. Pendidik. Islam*, vol. 2, no. 3, pp. 13–19, 2025.
- [5] P. Pujiyanto, "Aplikasi Self Assesment Checkup Covid-19 Berbasis Android," *J. Digit. Teknol. Inf.*, vol. 3, no. 1, p. 28, 2020.
- [6] A. H. Cholili, A. Z. Zainuddin, T. S. Rohma, C. M. Alfayn, N. Dewi, and M. Mahbubi, "Penanganan Prokrastinasi Mahasiswa melalui Muhasabah Diri," *J. Theory Pract. Islam. Guid. Couns.*, vol. 1, no. 1, pp. 1–13, 2024.



- [7] I. I. J. Rifka Alkhilyatul Ma'rifat, I Made Suraharta, "No Title 濟無No Title No Title No Title," vol. 2, pp. 306–312, 2024.
- [8] M. Uin and A. Banjarmasin, "Muhasabah sebagai Pengembangan Psikologi Islam," *J. Stud. Insa.*, vol. 11, no. 2, pp. 93–105, 2023.
- [9] I. I. J. Rifka Alkhilyatul Ma'rifat, I Made Suraharta, "No Title 濟無No Title No Title No Title," vol. 2, no. 3, pp. 306–312, 2024.
- [10] M. D. Kurniasanti, "Muhasabah Sebagai Upaya Mengatasi Perilaku Menyimpang Melalui Pendekatan Konseling Individual," *NATHIQIYYAH J. Psikol. Islam*, vol. 7, pp. 205–216, 2024.
- [11] J. Sarjana, T. Informatika, P. Studi, T. Informatika, and U. Ahmad, "P engembangan Aplikasi Mutaba ' ah u ntuk Evaluasi Ibadah Harian Berbasis Android," vol. 7, no. 2, pp. 31–37, 2019.
- [12] S. Narulita, A. Nugroho, and M. Z. Abdillah, "Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS) Universitas Nasional Karangturi Semarang , Indonesia ( deskripsi ) dan perancangan sistem , khususnya pada pemrogr," no. 3, pp. 244–256, 2024.
- [13] W. Widyatmoko and N. Pamungkas, "Pemodelan Unified Modeling Language pada Sistem Aplikasi Pariwisata (SiAP)," *J. Bumigora Inf. Technol.*, vol. 4, no. 1, pp. 73–84, 2022.
- [14] A. Digi Maulana Syah, "Application of Self Assessment Techniques in PAI Learning," *J. Pendidik. Indones.*, vol. 1, no. 02, pp. 81–86, 2023.
- [15] Y. S. Rahayu, Y. Saputra, and D. Irawan, "Implementasi Metode Waterfall Pada Pengembangan Sistem Informasi Mobile E-Disarpus," *Zo. J. Sist. Inf.*, vol. 6, no. 2, pp. 523–534, 2024.
- [16] R. A. Setyawan, "Penerapan Firebase Realtime Database Pada Aplikasi Catatan Harian Diabetes Melitus," *J. Inform. Komputer, Bisnis dan Manaj.*, vol. 22, no. 1, pp. 1–9, 2024.
- [17] A. A. M. Suradi, A. Syam, S. Alam, A. Bahtiar, and A. R. Johanis, "Pemanfaatan Firebase Realtime Database Dalam Perancangan Aplikasi Penilaian Siswa SMK Negeri 2 Pangkep Secara Realtime," *JUKI J. Komput. dan Inform.*, vol. 6, no. 2, pp. 146–154, 2024.
- [18] S. Fitriyah, A. Wijayanti, and M. R. Purwaningrum, "Pelaksanaan Self Assessment Peserta Didik pada Pembelajaran Bahasa Indonesia Sekolah Dasar," *J. Inovasi, Eval. dan Pengemb. Pembelajaran*, vol. 4, no. 1, pp. 121–127, 2024.