

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi mobile pembelajaran metode Tilawati yang dapat digunakan secara mandiri oleh siswa TPQ. Aplikasi dirancang agar mampu menyajikan materi pembelajaran secara digital, interaktif, dan mudah diakses tanpa ketergantungan pada buku cetak maupun koneksi internet.

Pengembangan sistem menggunakan pendekatan *Agile Software Development*, dengan alasan metode ini memungkinkan proses pengembangan dilakukan secara iteratif dan adaptif. Setiap *sprint* fokus pada penyelesaian satu atau dua fitur inti, sehingga sistem dapat terus dievaluasi dan ditingkatkan berdasarkan kebutuhan yang muncul.

Pengumpulan data dilakukan secara manual dari buku cetak Tilawati, karena tidak tersedia *dataset* digital resmi. Proses konversi dilakukan menggunakan teknologi *Optical Character Recognition* (OCR), kemudian diverifikasi secara manual dan disusun dalam format *JavaScript Object Notation* (JSON). Format ini memungkinkan data dapat dimuat langsung ke dalam aplikasi tanpa bergantung pada koneksi internet.

Jenis data yang berhasil dikembangkan meliputi:

1. Data Tilawati yang berisi baris ayat latihan per halaman yang diorganisasi berdasarkan jilid.
2. Data Materi yang berisi konten seperti hukum bacaan tajwid dan materi-materi lain sesuai dengan yang ada dalam buku cetak Tilawati.
3. Data Kuis yang berisi soal pilihan ganda untuk menguji pemahaman siswa terhadap materi.

Aplikasi dikembangkan menggunakan *framework* Flutter dan bahasa pemrograman Dart. Fitur-fitur utama yang berhasil diimplementasikan antara lain:

1. Tampilan daftar materi Tilawati berdasarkan jilid.

2. Halaman detail per ayat dan per baris sesuai struktur buku.
3. Navigasi antar halaman materi.
4. Kuis interaktif dengan penilaian otomatis.
5. Penyimpanan halaman terakhir yang dibaca dan hasil kuis secara *online* melalui integrasi dengan *database* Supabase.

Proses pengujian dilakukan pada tiap akhir iterasi pengembangan, untuk memastikan fungsionalitas dan tampilan sistem sesuai dengan tujuan awal. Hasil pengujian menunjukkan bahwa aplikasi mampu berfungsi dengan baik, memuat data secara cepat tanpa internet, dan memberikan pengalaman belajar yang lebih fleksibel.

Dengan demikian, aplikasi ini telah berhasil dikembangkan dan memenuhi kebutuhan siswa TPQ untuk belajar Tilawati secara mandiri. Aplikasi ini juga menunjukkan potensi untuk dikembangkan lebih lanjut, seperti penambahan fitur audio bacaan, latihan tajwid lanjutan, atau fitur yang lebih spesifik melalui layanan *backend* yang lebih lengkap.

4.2 IMPLEMENTASI DESAIN ANTARMUKA

Antarmuka pengguna (*user interface/UI*) dalam aplikasi belajar Tilawati dirancang dengan prinsip kesederhanaan, keterbacaan, dan kemudahan navigasi. Implementasi UI dibangun menggunakan *framework* Flutter, dengan pendekatan modular berbasis *widget*, sehingga tampilan dapat dikembangkan secara efisien dan responsif. Setiap halaman aplikasi menyesuaikan dengan alur pembelajaran digital bagi siswa TPQ dan admin sebagai pengelola data pengguna.

Seluruh tampilan antarmuka dirancang dan dibangun menggunakan *framework* Flutter, yang memungkinkan pengembangan UI responsif lintas platform. Komponen visual dikembangkan menggunakan *widget* standar *Flutter*, dipadukan dengan aset visual seperti ikon dan tipografi yang relevan dengan tema pendidikan Islami.

1. Halaman Pembuka dan Autentikasi

a. *Splash Screen*

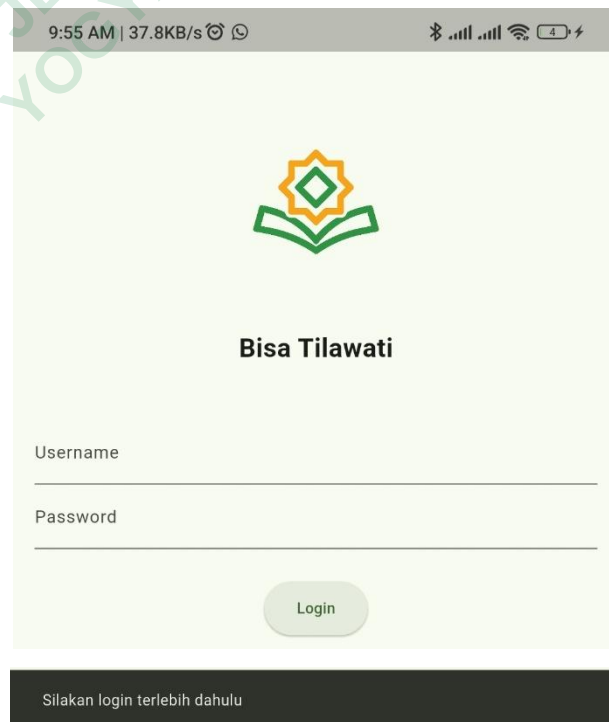
Menampilkan logo dan nama aplikasi sebagai tampilan pembuka.



Gambar 4.1 Halaman *Splash Screen*

b. Halaman *Login*

Pengguna memasukkan email dan kata sandi untuk mengakses sistem. Validasi kesalahan ditampilkan secara jelas jika data tidak sesuai.



Gambar 4.2 Halaman *Login*



Bisa Tilawati

Username

Password

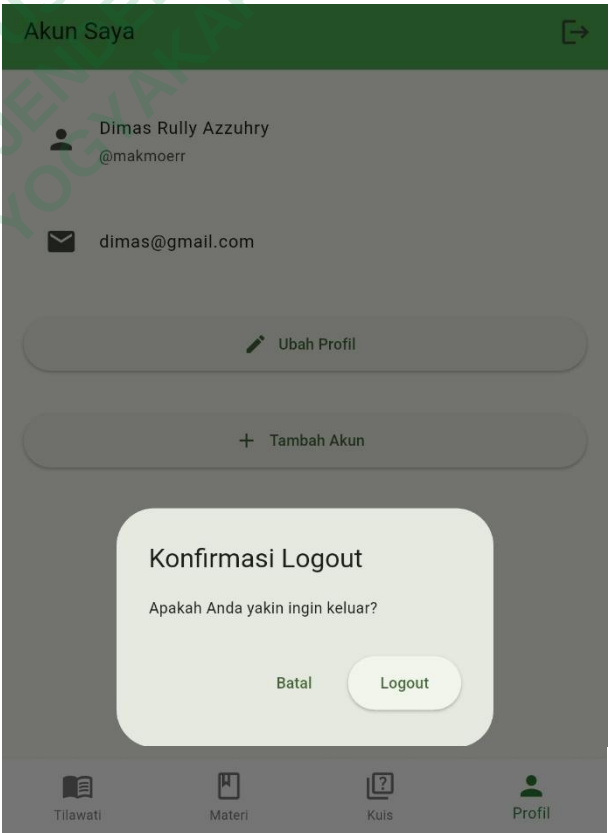
Login

Username atau password salah

Gambar 4.3 Halaman *Login* jika data tidak sesuai

c. Konfirmasi *Logout*

Pop-up konfirmasi muncul untuk mencegah *logout* tidak sengaja.



Akun Saya

Dimas Rully Azzuhry
@makmoerr

dimas@gmail.com

Ubah Profil

+ Tambah Akun

Konfirmasi Logout

Apakah Anda yakin ingin keluar?

Batal Logout

Tilawati Materi Kuis Profil

Gambar 4.4 Tampilan *Logout*

2. Menu Utama dan Navigasi Fitur

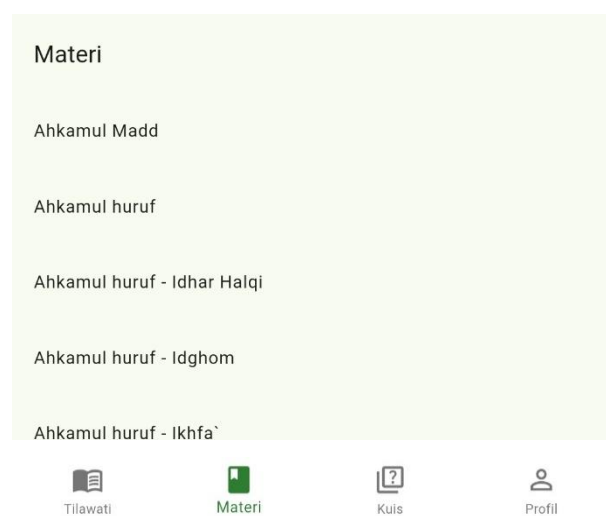
Setelah *login*, pengguna diarahkan ke halaman beranda fitur utama, yaitu:

a. Tilawati



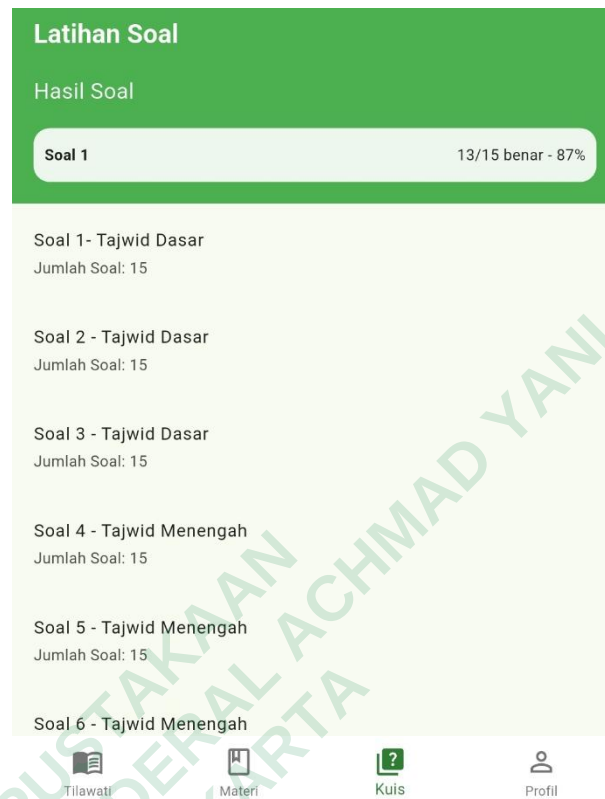
Gambar 4.5 Halaman Tilawati

b. Materi



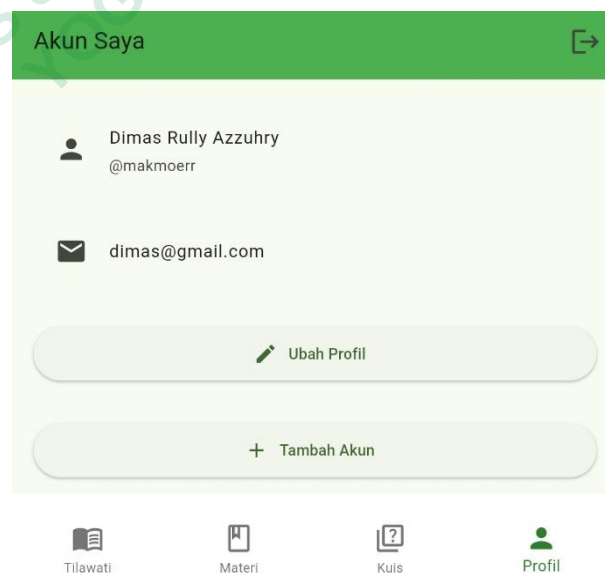
Gambar 4.6 Halaman Materi

c. Kuis



Gambar 4.7 Halaman Kuis

d. Profil Pengguna



Gambar 4.8 Halaman Profil

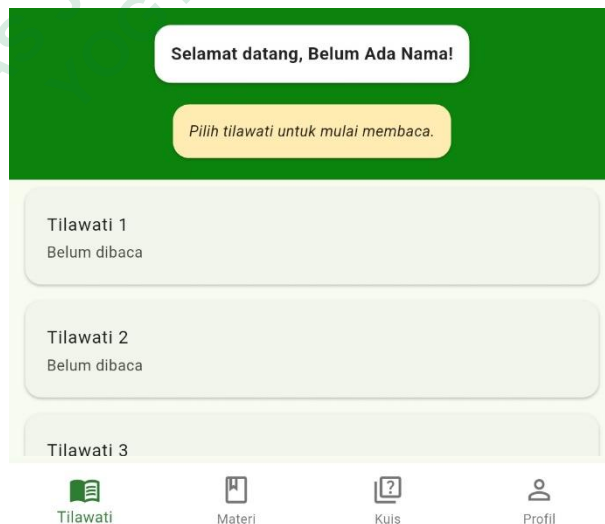
Navigasi dilakukan dengan *icon* menu dan tombol yang mudah dijangkau oleh pengguna.

3. Fitur Tilawati

Halaman Tilawati *List* menampilkan daftar jilid Tilawati berdasarkan tingkat belajar. Setiap kartu berisi nama jilid dan tombol akses detail.



Gambar 4.9 Halaman Tilawati



Gambar 4.10 Halaman Tilawati pada *User* baru

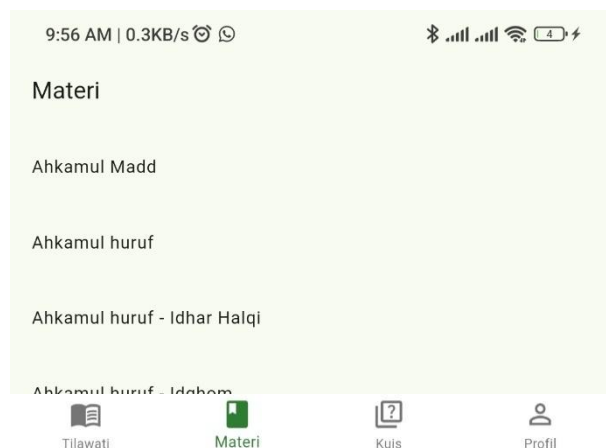
Halaman Tilawati Detail menampilkan baris-baris ayat dalam bentuk teks Arab sesuai struktur buku cetak. Dilengkapi fitur navigasi antar halaman.



Gambar 4.11 Halaman Tilawati Detail

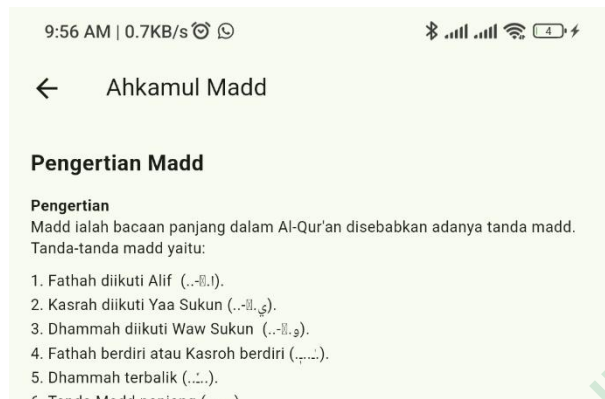
4. Fitur Materi

Halaman Materi berisi daftar topik materi tajwid dan hukum bacaan. Masing-masing topik menampilkan judul dan deskripsi singkat.



Gambar 4.12 Halaman Materi

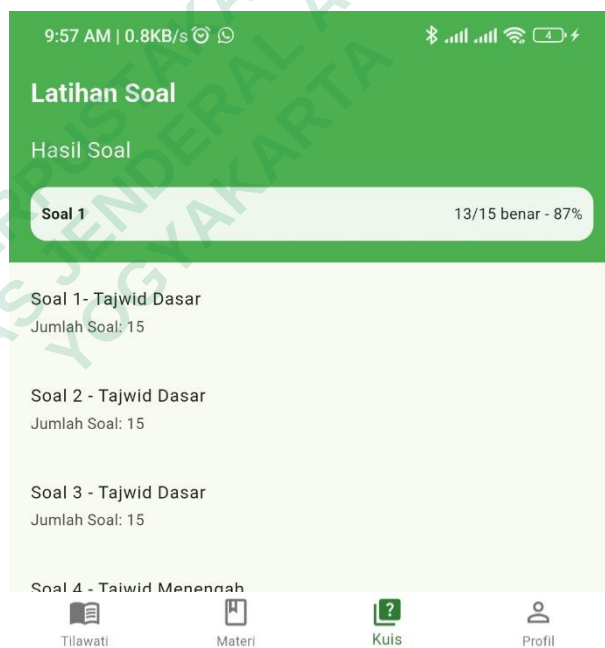
Halaman Materi Detail menyajikan isi materi secara lengkap, terdiri dari sub judul dan penjelasan dalam paragraf.



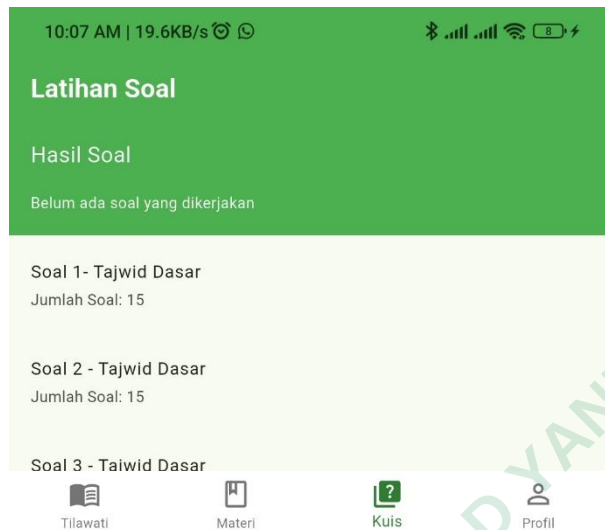
Gambar 4.13 Halaman Materi Detail

5. Fitur Kuis

Halaman Kuis *List* menampilkan daftar kuis berdasarkan topik. Jika belum ada hasil kuis, muncul status "belum dikerjakan".



Gambar 4.14 Halaman Kuis



Gambar 4.15 Halaman Kuis pada *User* baru

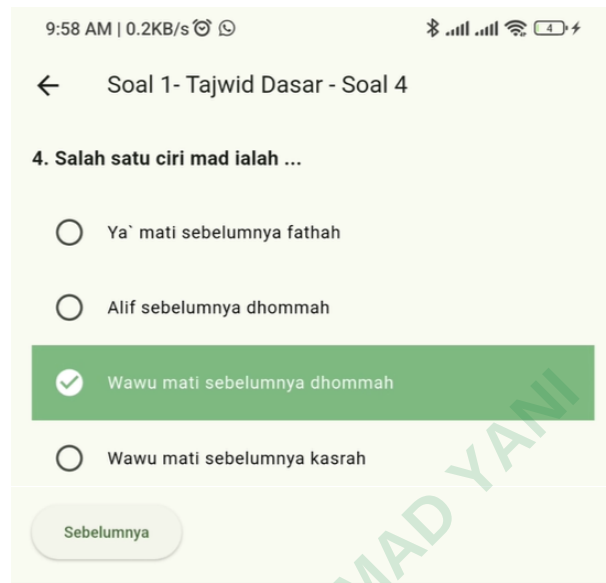
Halaman Kuis Detail menampilkan soal dan pilihan jawaban ganda. Tampilan sederhana memudahkan konsentrasi siswa.



Gambar 4.16 Halaman Kuis Detail

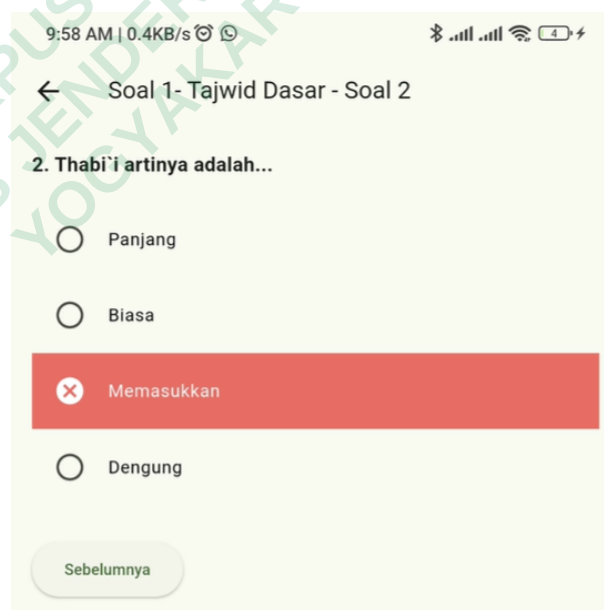
Fitur Hasil Kuis pada Halaman Kuis Detail:

- a. Jawaban benar ditandai secara visual.



Gambar 4.17 Halaman Kuis Detail jika jawaban benar

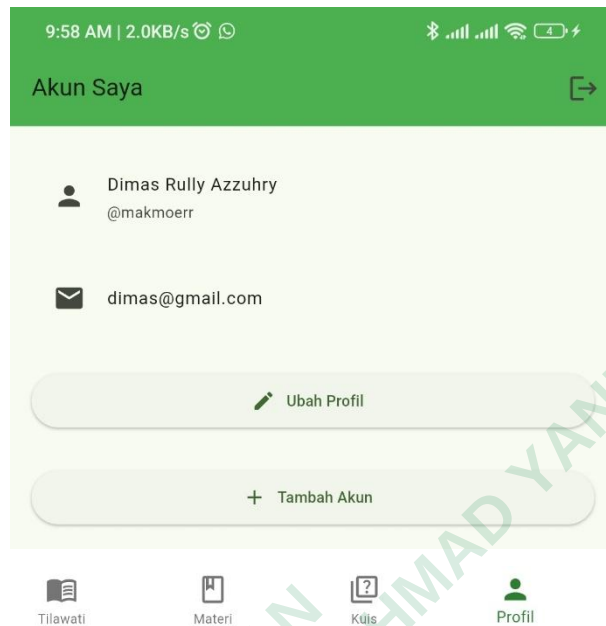
- b. Jawaban salah akan menandai pilihan yang salah.



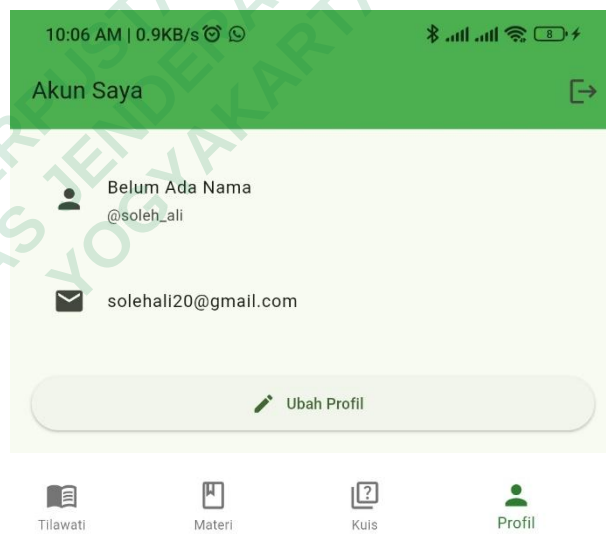
Gambar 4.18 Halaman Kuis Detail jika jawaban salah

6. Fitur Profil

Halaman Profil menampilkan data pengguna seperti nama, email, dan *role*.

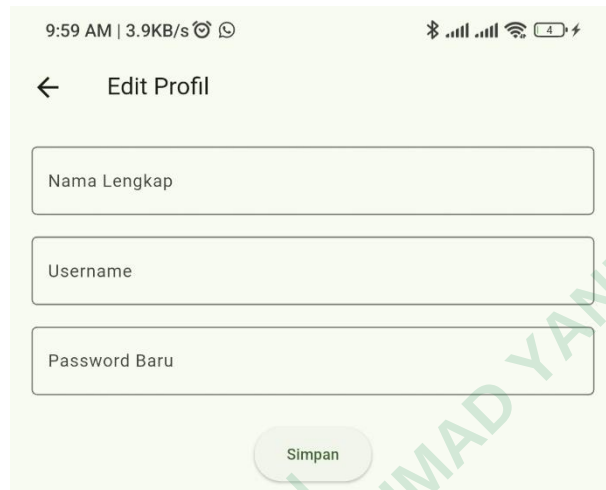


Gambar 4.19 Halaman Profil



Gambar 4.20 Halaman Profil pada *User* baru

Pada halaman *Update* Profil pengguna dapat mengubah nama atau kata sandi.



9:59 AM | 3.9KB/s

← Edit Profil

Nama Lengkap

Username

Password Baru

Simpan

Gambar 4.21 Halaman *Update* Profil



10:07 AM | 2.8KB/s

← Edit Profil

Nama Lengkap
Soleh Ali Al Harist

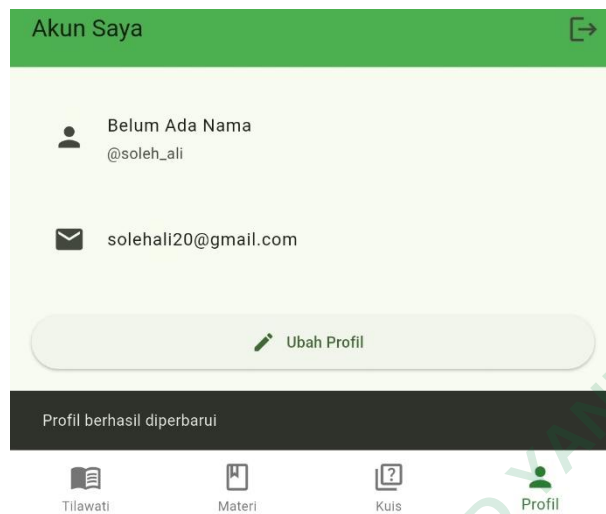
Username

Password Baru

Simpan

Gambar 4.22 Halaman *Update* Profil pada saat *input form*

Menampilkan *Feedback Update* ketika setelah diperbarui sehingga informasi langsung ditampilkan dengan notifikasi berhasil.

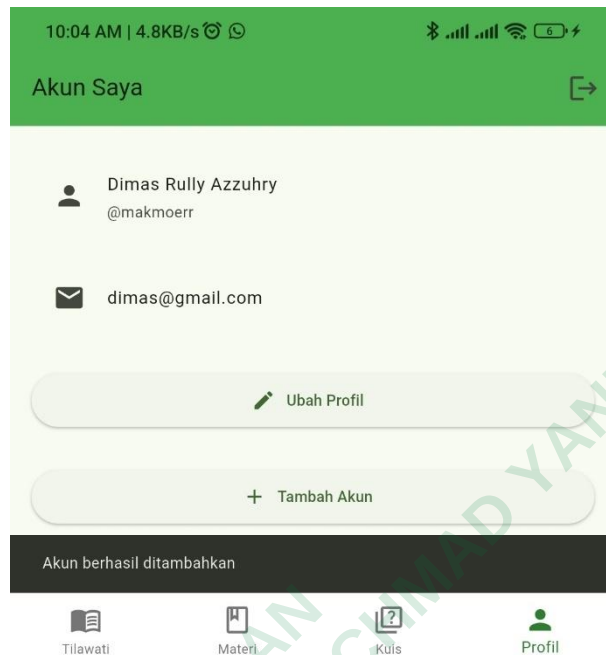


Gambar 4.23 *Feedback/Message* ketika profil diupdate

7. Fitur Tambah *User* (Admin/Guru)

Admin/Guru memiliki akses ke menu Tambah *User*.

Gambar 4.24 Halaman Tambah *User*



Gambar 4.25 *Feedback/Message* ketika *User* baru berhasil ditambahkan

Setelah data tersimpan, sistem memberikan notifikasi berhasil, dan *user* akan terdaftar dalam *database* Supabase.

4.3 BASIS DATA

Sistem aplikasi belajar Tilawati menggunakan dua pendekatan utama dalam pengelolaan basis data, yaitu penyimpanan data lokal berbasis JSON dan basis data daring menggunakan Supabase. Pemisahan ini bertujuan untuk menyeimbangkan antara efisiensi akses, ketersediaan *offline*, dan kebutuhan akan penyimpanan data dinamis.

4.3.1 Data Lokal (*Offline*) – Format JSON

Untuk data yang bersifat statis dan tidak sering berubah, seperti materi bacaan Tilawati, konten materi, dan soal kuis, digunakan struktur data *JavaScript Object Notation* (JSON). Data JSON disimpan secara lokal dalam direktori aplikasi dan dimuat menggunakan fungsi `rootBundle.loadString()` dari Flutter.

Beberapa jenis data yang disimpan secara lokal antara lain:

1. Tilawati *List* dan Detail untuk menyimpan daftar jilid dan halaman berisi baris-baris ayat.

2. Materi *List* dan *Detail* berisi kumpulan topik materi tajwid dan hukum bacaan yang sesuai dengan pembelajaran metode Tilawati.
3. Kuis *List* dan Kuis *Detail* untuk menyimpan soal pilihan ganda beserta opsi dan kunci jawaban.

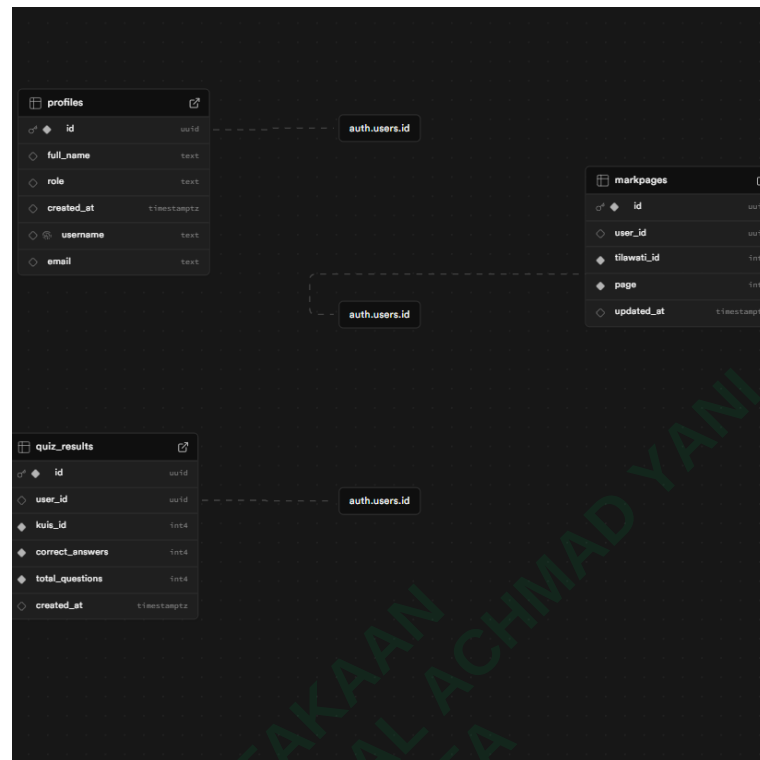
Dengan menggunakan format JSON, aplikasi dapat berjalan tanpa koneksi internet, sehingga cocok digunakan oleh siswa TPQ yang tidak selalu memiliki akses jaringan. Selain itu, struktur JSON mudah dikembangkan dan fleksibel dalam pemrosesan data.

4.3.2 Basis Data Daring (*Online*) – Supabase

Untuk data yang bersifat dinamis dan memerlukan autentikasi atau penyimpanan lintas perangkat, aplikasi menggunakan layanan basis data *cloud* Supabase. Supabase dipilih karena menyediakan fitur *real-time database*, autentikasi pengguna, dan integrasi langsung dengan Flutter melalui REST API dan SDK resmi.

Tabel-tabel utama yang disimpan di Supabase antara lain:

1. *user_profiles* untuk menyimpan informasi pengguna seperti nama, email, dan *role* (admin/siswa).
2. *quiz_results* untuk menyimpan hasil kuis yang dikerjakan oleh pengguna.
3. *mark_pages* untuk menyimpan halaman terakhir yang dibaca oleh masing-masing pengguna.



Gambar 4.26 Struktur Basis Data di Supabase

Selain itu, data pengguna (*Users*) disimpan di layanan Supabase *Authentication*. Data ini mencakup *email* dan *password* yang digunakan untuk proses *login*, serta dikaitkan dengan tabel *user_profiles* untuk menyimpan nama lengkap dan peran (admin atau siswa). Supabase *Authentication* juga menangani proses registrasi dan verifikasi *login* secara aman.

Data pada Supabase diakses menggunakan koneksi internet, dan ditangani melalui metode *asynchronous* dalam Flutter. Data ini juga memungkinkan pengembangan lanjutan seperti fitur akun guru, laporan hasil belajar, atau kolaborasi daring.

4.4 FITUR-FITUR SISTEM

Aplikasi belajar Tilawati yang dikembangkan memiliki beberapa fitur utama yang dirancang untuk menunjang proses pembelajaran mandiri siswa TPQ secara digital. Setiap fitur dirancang berdasarkan hasil analisis kebutuhan pengguna melalui metode *Design Thinking* dan direalisasikan secara bertahap melalui

pendekatan *Agile*. Berikut adalah deskripsi fitur-fitur utama yang telah diimplementasikan dalam sistem:

1. Halaman Login

Fitur *login* digunakan untuk mengautentikasi pengguna berdasarkan email dan kata sandi yang terdaftar. Sistem menggunakan Supabase *Authentication* untuk menyimpan dan memverifikasi data pengguna. Setelah berhasil *login*, pengguna akan diarahkan ke halaman utama sesuai perannya (admin atau siswa).

2. Tilawati

Fitur utama ini menyajikan daftar jilid Tilawati yang dapat dipilih oleh pengguna. Setelah memilih jilid, pengguna diarahkan ke halaman materi per halaman. Setiap halaman menampilkan baris-baris ayat dalam bentuk teks Arab sesuai dengan struktur asli buku cetak Tilawati. Fitur ini mendukung navigasi antar halaman dan memungkinkan siswa membaca secara bertahap.

3. Materi

Fitur materi menyediakan kumpulan teori pembelajaran, seperti hukum bacaan dan tajwid dasar. Setiap topik ditampilkan dalam bentuk daftar judul yang dapat diklik, kemudian ditampilkan isi materi secara terstruktur berdasarkan sub judul dan isi penjelasan. Fitur ini berfungsi sebagai referensi untuk memperkuat pemahaman siswa terhadap bacaan.

4. Kuis

Fitur kuis menyediakan soal pilihan ganda yang berfungsi sebagai latihan evaluatif setelah mempelajari materi. Kuis ditampilkan per topik dan terdiri dari pertanyaan serta beberapa opsi jawaban. Sistem akan memberikan skor otomatis berdasarkan pilihan yang benar, dan hasilnya disimpan secara daring menggunakan *database* Supabase.

5. Penanda halaman (*Mark Page*)

Fitur ini memungkinkan pengguna untuk secara otomatis menyimpan halaman terakhir yang dibaca pada materi Tilawati. Ketika aplikasi dibuka

kembali, sistem akan menampilkan halaman terakhir tersebut sebagai referensi, sehingga pengguna tidak perlu mencari ulang.

6. Hasil Kuis

Setelah menyelesaikan kuis, pengguna dapat melihat hasil skor yang diperoleh. Data ini disimpan ke dalam tabel *quiz_results* di Supabase dan dapat digunakan untuk memantau perkembangan belajar pengguna dari waktu ke waktu.

7. Profil User

Pengguna dapat melihat dan memperbarui data profilnya, seperti nama lengkap dan kata sandi. Informasi pengguna disimpan dalam tabel *user_profiles* dan terhubung dengan akun *login* pada Supabase *Authentication*.

8. Tambah Akun

Admin memiliki akses untuk menambahkan pengguna baru melalui fitur tambah akun. Admin dapat mengisi data seperti nama, email, dan peran pengguna (admin/guru atau siswa). Pembuatan akun dilakukan melalui pemanggilan Supabase *Edge Function* yang menangani proses pembuatan akun baru secara aman.

Fitur-fitur tersebut saling terintegrasi untuk mendukung tujuan utama aplikasi, yaitu memberikan pengalaman belajar Tilawati yang mandiri, mudah diakses, dan sesuai dengan kebutuhan siswa di lingkungan TPQ.

4.5 PEMBAHASAN

4.5.1 Pengumpulan Data

Data utama yang digunakan dalam aplikasi ini berasal dari buku cetak metode Tilawati. Karena tidak tersedia dalam bentuk digital resmi, proses pengumpulan data dilakukan secara manual dengan memindai halaman buku dan mengonversinya ke teks menggunakan *Optical Character Recognition* (OCR) dengan Google Lens. Setelah proses OCR, data diverifikasi ulang secara manual untuk memastikan akurasi teks Arab, terutama dalam baris-baris ayat Tilawati.

Untuk materi dan soal kuis, peneliti menyusun ulang struktur datanya berdasarkan referensi buku tajwid dasar. Semua data disimpan dalam format *JavaScript Object Notation* (JSON) agar mudah diproses di aplikasi.

4.5.2 Implementasi penggunaan FCA CLI

Aplikasi dikembangkan menggunakan arsitektur modular berbasis FCA CLI (*Feature-Component-Architecture*), yaitu pendekatan yang membagi *project* Flutter ke dalam tiga lapisan utama:

1. *Feature*: Menyimpan logika tiap fitur (Tilawati, Materi, Kuis, User, dll.).
2. *Component*: Menyimpan bagian *reusable* seperti tombol, *card*, dan dialog.
3. *Architecture*: Menyimpan konfigurasi, *router*, *theme*, dan koneksi Supabase.

FCA CLI digunakan untuk mempercepat pembuatan struktur folder dan *file*, sekaligus menjaga konsistensi dalam penulisan kode. Setiap fitur dikembangkan secara terpisah dan terintegrasi dalam satu alur navigasi. Pendekatan ini membuat proyek lebih rapi, *scalable*, dan mudah untuk dikembangkan lebih lanjut.

Langkah-langkah Implementasi FCA CLI di penelitian seperti pada langkah berikut:

1. Inisialisasi *Project*

Pembuatan *project* dengan menjalankan perintah `flutter create <project_app>` pada CMD. Untuk implementasi inisialisasi pada penelitian seperti pada kode berikut:

```
flutter create bisa_tilawati
```

Setelah kode tersebut dijalankan, Flutter akan membuat direktori proyek sesuai dengan struktur proyek Flutter.

2. Menambahkan Struktur FCA

FCA CLI digunakan untuk menghasilkan struktur folder dan file secara otomatis dengan perintah:

```
fca-cli add feature <feature_project>
```

Perintah tersebut akan membuat struktur:

```
lib/features/bisa_tilawati/
├── data/
├── domain/
└── presentation/
```

Direktori tersebut berisi:

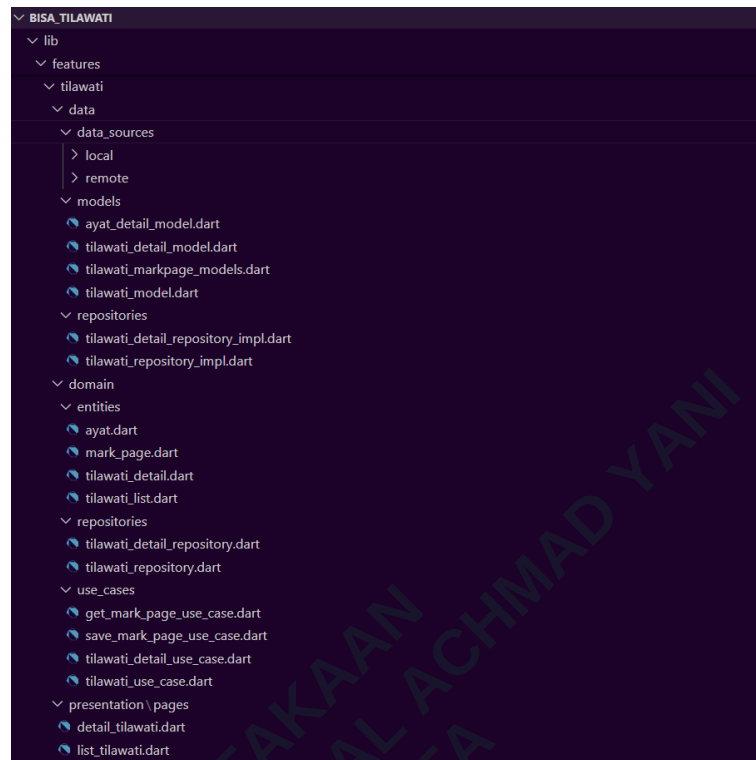
- a. Data berisi *data source* untuk mengambil data dari *database*, *models* untuk mendefinisikan model data ke sistem, dan *repository* sebagai penghubung antara direktori Data dan Domain.
- b. Domain berisi *entities* untuk mendefinisikan data yang ada pada sistem, *use case* yang berisi metode pemanggilan data untuk digunakan UI, dan *repository* yang terhubung pada *repository* yang ada pada direktori Data.
- c. Presentation yang berisi UI untuk menampilkan halaman.

Semua fitur yang digunakan dalam sistem ini menggunakan FCA CLI.

Berikut implementasi fitur yang dibuat dan data yang ada pada fitur:

1. Tilawati

Fitur Tilawati dibuat dengan menggunakan perintah `fca-cli add feature tilawati` di terminal CMD. Fitur Tilawati berisi direktori seperti pada Gambar 4.27.



Gambar 4.27 Direktori pada fitur Tilawati

Berdasarkan gambar struktur direktori proyek yang menggunakan arsitektur FCA, Fitur Tilawati dan terdiri dari tiga lapisan utama yaitu data, domain, dan *presentation*.

1. Data

Berisi *models* untuk merepresentasikan data dari *file* JSON, seperti daftar Tilawati (*tilawati_model.dart*), detail halaman (*tilawati_detail_model.dart*), dan penanda halaman (*tilawati_markpage_models.dart*). Serta *repository* yang berisi implementasi seperti (*tilawati_repository_impl.dart*) digunakan untuk mengambil data dari sumber lokal.

2. Domain

Digunakan untuk menyimpan entitas inti (*tilawati_list.dart*, *tilawati_detail.dart*, *mark_page.dart*) serta definisi *use_cases* seperti mengambil data *list* Tilawati, membuka detail halaman, dan menyimpan halaman terakhir yang dibaca.

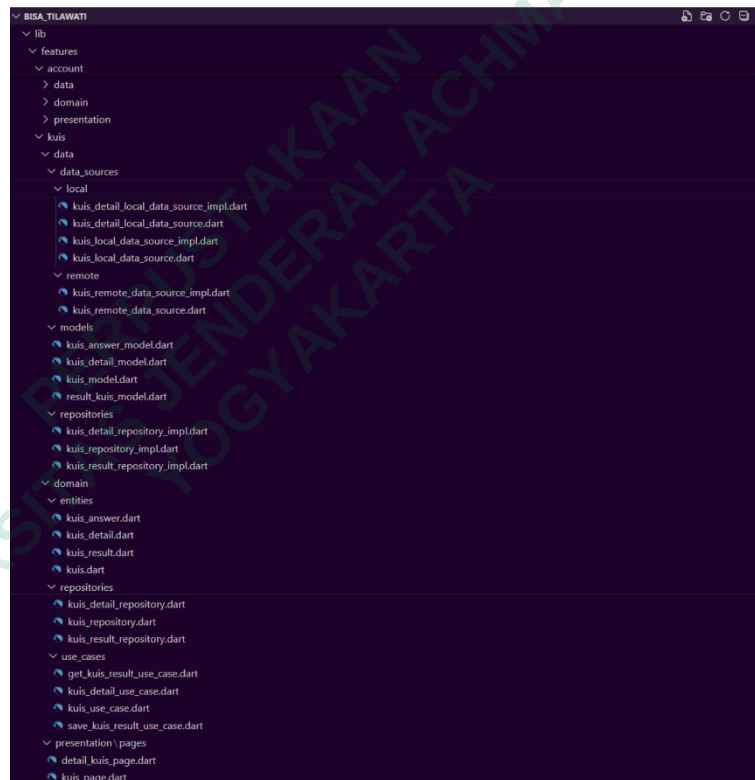
3. Presentation

Terdiri dari *file* UI seperti (*list_tilawati.dart*) dan (*detail_tilawati.dart*) yang menampilkan daftar jilid serta baris ayat per halaman.

Fitur ini membaca data JSON dari aset lokal dan menampilkannya secara dinamis. Penanda halaman terakhir juga disimpan, agar pengguna dapat melanjutkan bacaan secara berkelanjutan.

2. Kuis

Fitur Kuis dibuat dengan menggunakan perintah `fca-cli add feature kuis` di terminal CMD. Fitur Kuis berisi direktori seperti pada Gambar 4.28.



Gambar 4.28 Direktori pada fitur Kuis

Fitur Kuis dibangun dengan arsitektur FCA yang terdiri dari lapisan data, domain, dan *presentation*.

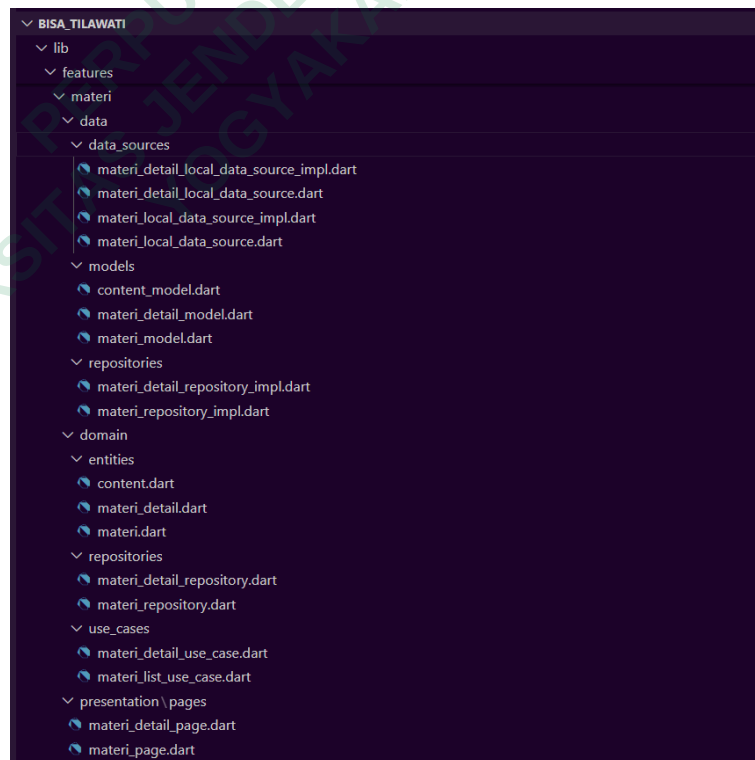
1. Lapisan Data untuk menangani pengambilan soal dari file JSON lokal serta pengiriman hasil kuis ke Supabase. Model data seperti (*kuis_model.dart*)

- dan (*kuis_result_model.dart*) yang merepresentasikan struktur soal dan hasil. *Repository* digunakan untuk mengelola akses data lokal dan *remote*.
2. Lapisan Domain berisi entitas seperti (*kuis.dart*) dan (*kuis_result.dart*), serta *use case* utama seperti (*kuis_use_case.dart*) untuk mengambil soal dan (*save_kuis_result_use_case.dart*) untuk menyimpan hasil ke Supabase.
 3. Lapisan Presentation menyediakan tampilan antarmuka berupa halaman daftar kuis (*kuis_page.dart*) dan halaman soal (*detail_kuis_page.dart*).

Dengan struktur ini, pengguna dapat mengakses soal secara *offline* dan menyimpan hasil secara daring. Pembagian layer juga mendukung kemudahan pengujian dan pengembangan sistem ke depannya.

3. Materi

Fitur Materi dibuat dengan menggunakan perintah `fca-cli add feature materi` di terminal CMD. Fitur Materi berisi direktori seperti pada Gambar 4.29.



Gambar 4.29 Direktori pada fitur Materi

Fitur Materi bertujuan menyajikan konten teori pembelajaran seperti hukum bacaan dan tajwid. Fitur ini disusun menggunakan arsitektur FCA, yang terbagi menjadi lapisan data, domain, dan *presentation*.

1. Lapisan Data

Folder *data_sources* berisi implementasi untuk membaca data materi dari *file* JSON lokal. Model data seperti (*materi_model.dart*), (*materi_detail_model.dart*), dan (*content_model.dart*) digunakan untuk memetakan struktur JSON ke dalam format objek Dart. *Repository* seperti (*materi_repository_impl.dart*) digunakan untuk mengambil data tersebut dari sumber lokal.

2. Lapisan Domain

Menyimpan entitas seperti (*materi.dart*) dan (*materi_detail.dart*), serta *use case* seperti (*materi_list_use_case.dart*) dan (*materi_detail_use_case.dart*), yang mengatur proses logika untuk mengambil daftar dan isi detail materi.

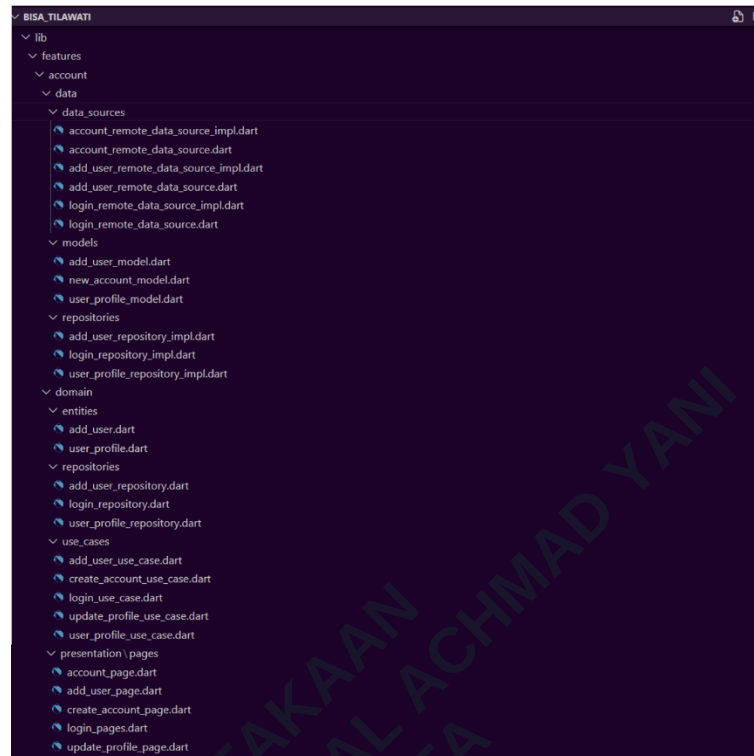
3. Lapisan Presentation

Berisi tampilan pengguna berupa (*materi_page.dart*) yang berisi daftar topik materi dan (*materi_detail_page.dart*) yang berisi isi konten materi per topik.

Fitur ini memungkinkan pengguna mengakses teori secara *offline*, dengan tampilan terstruktur berdasarkan judul, sub judul, dan isi materi.

4. Profil/Account

Fitur Tilawati dibuat dengan menggunakan perintah `fca-cli add feature account` di terminal CMD. Fitur Profil/Account berisi direktori seperti pada Gambar 4.30.



Gambar 4.30 Direktori pada fitur Profil (*Account*)

Fitur Profil (*Account*) berfungsi untuk mengelola data pengguna seperti *login*, pembuatan akun baru oleh admin/guru, serta pembaruan profil. Fitur ini disusun berdasarkan arsitektur FCA, yang terbagi dalam lapisan data, domain, dan *presentation*.

1. Lapisan Data

Folder *data_sources* menangani komunikasi *remote*, seperti *login*, pembuatan *user*, dan *update* profil menggunakan Supabase. Model seperti (*user_profile_model.dart*, *add_user_model.dart*, dan *new_account_model.dart*) merepresentasikan struktur data dari server. *Repository* berupa implementasi seperti data (*login_repository_impl.dart*) digunakan untuk mengelola proses autentikasi dan manipulasi data *user*.

2. Lapisan Domain

Berisi entitas seperti (*user_profile.dart*) dan (*add_user.dart*), serta *interface repository* dan *use case*, seperti:

- a. (*login_use_case.dart*) untuk *login* pengguna.
- b. (*add_user_use_case.dart*) untuk tambah *user* oleh admin/guru.

- c. (*update_profile_use_case.dart*) untuk mengubah nama atau *password*.

3. Lapisan Presentation

Berisi tampilan antarmuka pengguna seperti (*login_page.dart*, *add_user_page.dart*, dan *update_profile_page.dart*). Halaman ini terhubung langsung dengan *use case* yang bersangkutan untuk menjalankan proses autentikasi dan pengelolaan akun.

4.5.3 Implementasi Edge Function

Dalam proses pengembangan aplikasi belajar Tilawati, salah satu teknologi yang dimanfaatkan untuk mendukung fungsionalitas sistem secara aman dan efisien adalah Supabase Edge Function. Edge Function merupakan fitur dari Supabase yang memungkinkan pembuatan dan pemanggilan fungsi *backend* berbasis JavaScript/TypeScript yang dijalankan secara *serverless* di tepi jaringan (*edge*), dengan latensi rendah dan keamanan tinggi.

Fungsi ini digunakan dalam sistem untuk menangani proses pembuatan akun baru oleh admin, khususnya pada fitur Tambah *User*. Proses ini mencakup pembuatan email dan *password* yang langsung terdaftar di Supabase *Authentication*, sekaligus penyimpanan informasi tambahan seperti nama dan peran pengguna (admin atau siswa) ke dalam tabel *user_profiles*.

Alasan penggunaan Supabase Edge Function dalam proses ini adalah karena Supabase tidak mengizinkan pembuatan akun secara langsung dari sisi klien (*client-side*) untuk alasan keamanan. Selain itu Edge Function memungkinkan proses pendaftaran dilakukan melalui *backend* yang aman, dengan token akses terbatas. Fungsi ini juga dapat menangani dua proses sekaligus yaitu, register akun di *Authentication* dan *insert* data di tabel *user_profiles*.

Adapun alur prosesnya adalah sebagai berikut:

1. Admin mengisi *form* tambah user (nama, email, role).
2. Data tersebut dikirim ke *endpoint* Edge Function (*/create_user*).
3. Edge Function membuat akun baru di Supabase *Authentication*.

4. Jika berhasil, sistem otomatis menambahkan data ke tabel *user_profiles*.

Langkah implementasi untuk membuat Edge Function adalah menyiapkan *tools* dan *install package* yang dibutuhkan terlebih dahulu, terdapat beberapa *tools* dan *package* penting yang harus disiapkan, antara lain:

1. Supabase CLI dengan `npm install -g supabase`.
2. Node.js dan npm.
3. Docker dan Docker Compose.
4. *File .env* berisi konfigurasi Supabase yang didalamnya ada *URL* dan *Role Key*.

Untuk menjalankan dan menguji Supabase Edge Function secara lokal sebelum *dideploy* ke *cloud*, digunakan bantuan Docker agar *runtime* di Supabase tidak terganggu dan sesuai dengan spesifikasi Supabase. Selain itu juga agar dapat menjalankan Supabase lokal lengkap dengan layanan Auth, Database, dan Edge Function, sehingga seluruh proses *backend* dapat diuji tanpa koneksi ke server Supabase utama.

1. Untuk langkah pertama membuat Edge Function adalah inisialisasi proyek supabase dengan menjalankan perintah:

```
Supabase init
```

Perintah tersebut akan membuat struktur proyek Supabase berupa direktori yang didalamnya berisi (*config.toml*), direktori *functions*, dan direktori *migrations*.

2. Selanjutnya membuat *function create user* dengan menjalankan perintah:

```
supabase functions new create_user
```

Perintah tersebut akan membuat *function* di struktur proyek Supabase yang mana *function create user* akan dibuat di *indeks.ts*.

3. Lalu *file indeks.ts* diisi dengan logika untuk membuat akun baru dan menyimpan data di tabel *user_profiles*. Berikut cuplikan kode dari *indeks.ts*:

```

//Buat user via admin
const { data, error } = await supabase.auth.admin.createUser({
  email,
  password,
  user_metadata: { username, role },
  email_confirm: true,
});

if (error || !data?.user?.id) {
  return new Response(JSON.stringify({ error: error?.message
?? "Gagal membuat user" })), {
    status: 400,
    headers: { "Access-Control-Allow-Origin": origin },
  });
}

const userId = data.user.id;

//Tambahkan ke tabel profile
const { error: insertError } = await supabase
  .from("profiles")
  .insert([
    { id: userId, username, role, email }
  ]);

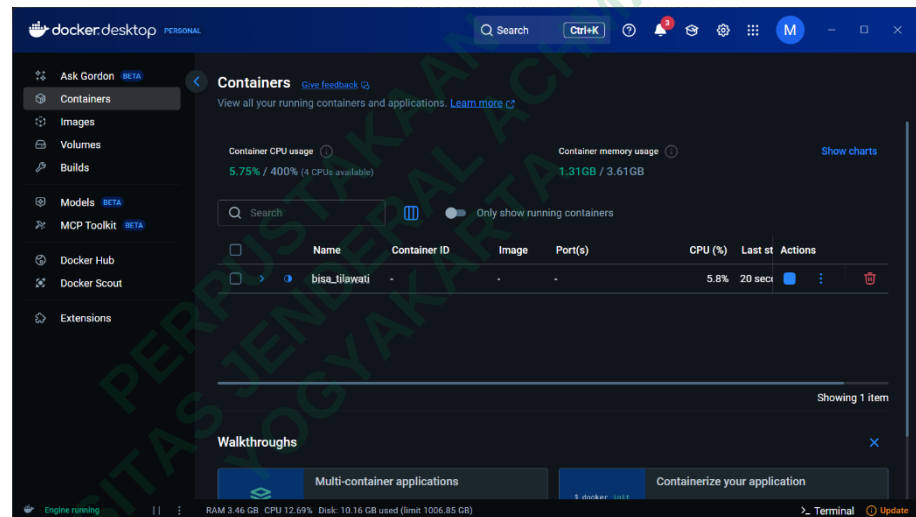
if (insertError) {
  return new Response(JSON.stringify({ error:
insertError.message })), {
    status: 400,
    headers: { "Access-Control-Allow-Origin": origin },
  });
}

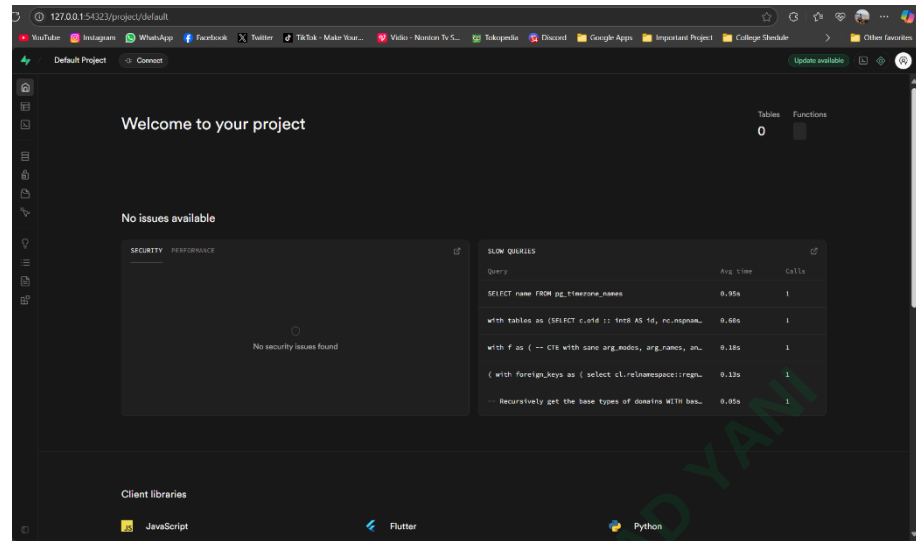
return new Response(JSON.stringify({ success: true })), {
  status: 200,
  headers: {
    "Content-Type": "application/json",
    "Access-Control-Allow-Origin": origin,
  },
};

```

Cuplikan kode tersebut adalah logika di *indeks.ts* untuk menambahkan data *User* dan *profile* ke Supabase.

4. Setelah menambahkan logika pada *indeks.ts*, lalu melakukan pengujian secara lokal dengan Docker yang mana sebelum memulai pengujian, Docker harus dijalankan. Dan setelah itu menjalankan perintah `supabase start` untuk menjalankan supabase secara lokal.





Gambar 4.33 Dashboard Supabase di server lokal

5. Menjalankan Edge Function di server lokal dengan menjalankan perintah `supabase functions serve create_user`. Edge Function dapat dipanggil dengan menggunakan `http://localhost:54321/functions/v1/create_user`.
6. *Deploy function* Supabase ke server cloud.

Setelah Edge Function selesai diuji di server lokal. *Function* di *deploy* dengan menjalankan perintah `supabase functions deploy create_user` di terminal CMD.

Sebelum dijalankan, harus mengonfigurasi *file .env* yang berisi URL dan *Role Key* untuk mengakses Supabase dengan menggunakan perintah `supabase secrets set --env-file .env` di terminal CMD.

```
C:\Users\azzuh\Documents\project\flutter\bisa_tilawati\supabase\functions\create_user>supabase secrets set --env-file .env
Using workdir C:\Users\azzuh\Documents\project\flutter\bisa_tilawati
Selected project: kiydkxkpfzlmypuhrc
Finished supabase secrets set.
C:\Users\azzuh\Documents\project\flutter\bisa_tilawati\supabase\functions\create_user>
```

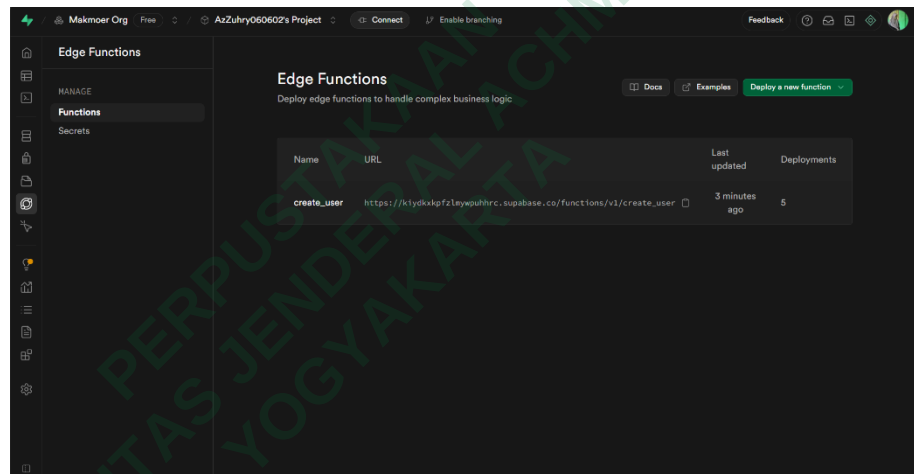
Gambar 4.34 Tampilan terminal CMD saat mengonfigurasi *file .env*

```

Download https://jsr.io/@supabase/functions-js/meta.json
Download https://deno.land/std@0.192.0/async/mod.ts
Download https://esm.sh/@supabase/auth-js@2.70.0/es2022/auth-js.mjs
Download https://esm.sh/@supabase/functions-js@2.4.4/es2022/functions-js.mjs
Download https://esm.sh/@supabase/node-fetch@2.6.15/es2022/node-fetch.mjs
Download https://esm.sh/@supabase/postgrest-js@1.19.4/es2022/postgrest-js.mjs
Download https://esm.sh/@supabase/realtime-js@2.11.15/es2022/realtime-js.mjs
Download https://esm.sh/@supabase/storage-js@2.7.1/es2022/storage-js.mjs
Download https://esm.sh/@supabase/supabase-js@2.50.2/es2022/supabase-js.mjs
Download https://deno.land/std@0.192.0/async/abortable.ts
Download https://deno.land/std@0.192.0/async/timeout.ts
Download https://deno.land/std@0.192.0/async/debounce.ts
Download https://deno.land/std@0.192.0/async/deferred.ts
Download https://deno.land/std@0.192.0/async/delay.ts
Download https://deno.land/std@0.192.0/async/mux_async_iterator.ts
Download https://deno.land/std@0.192.0/async/pool.ts
Download https://deno.land/std@0.192.0/async/tee.ts
Download https://deno.land/std@0.192.0/async/retry.ts
Download https://esm.sh/@supabase/node-fetch@2.6.14?target=es2022
Download https://esm.sh/isows@1.0.7?target=es2022
Download https://esm.sh/node/buffer.mjs
Download https://deno.land/std@0.192.0/_util/asserts.ts
Download https://esm.sh/isows@1.0.7/es2022/isows.mjs
Download https://jsr.io/@supabase/functions-js/2.4.4/meta.json
Download https://esm.sh/@supabase/node-fetch@2.6.13?target=es2022
Deploying Function: create_user (script size: 63.24kB)
Deployed Functions on project kiydkxkpfzlmypuhhrc: create_user
You can inspect your deployment in the Dashboard: https://supabase.com/dashboard/project/kiydkxkpfzlmypuhhrc/functions
C:\Users\azzuh\Documents\project\flutter\bisa_tilawati\supabase\functions\create_user>

```

Gambar 4.35 Tampilan terminal CMD saat mendeploy Edge Function



Gambar 4.36 Edge Function di server Supabase

Edge Function dapat diakses di Flutter dengan menggunakan cara pemanggilan seperti cuplikan kode berikut:

```

Future<void> createUser(AddUserModel user) async {
  final url = Uri.parse(
    'https://kiydkxkpfzlmypuhhrc.supabase.co/functions/v1/create_user',
  );

  final supabase = Supabase.instance.client;
  final jwt = supabase.auth.currentSession?.accessToken;

  final response = await http.post(
    url,
    headers: {

```

```

        'Content-Type': 'application/json',
        'Authorization': 'Bearer $jwt',
    },
    body: jsonEncode(user.toJson()),
  );
}

```

Implementasi ini telah diuji dan berhasil menjalankan fungsinya dengan baik. Admin dapat menambahkan pengguna baru tanpa harus melakukan proses registrasi manual dari sisi pengguna. Hal ini meningkatkan efisiensi dalam pengelolaan akun dan menjaga keamanan proses pembuatan akun baru.

Dengan memanfaatkan Edge Function, aplikasi menjadi lebih modular, terpisah antara *frontend* dan *backend*, serta memiliki kontrol keamanan yang lebih kuat dibandingkan pendekatan konvensional berbasis REST saja. Teknologi ini juga membuka peluang pengembangan fitur *server-side* lainnya, seperti validasi data atau pemrosesan hasil kuis lanjutan.

4.5.4 Kendala pada penelitian

Selama proses penelitian dan pengembangan aplikasi belajar Tilawati, terdapat beberapa kendala yang dihadapi, baik dari sisi teknis maupun non-teknis. Kendala-kendala tersebut antara lain:

1. Keterbatasan *Dataset* Digital

Salah satu tantangan utama adalah tidak tersedianya data Tilawati dalam bentuk digital. Seluruh isi materi dan ayat-ayat harus dikonversi secara manual dari buku cetak menggunakan bantuan teknologi *Optical Character Recognition* (OCR). Proses ini membutuhkan waktu dan ketelitian tinggi karena teks Arab tidak selalu terbaca sempurna oleh sistem OCR.

2. Struktur Data JSON yang Kompleks

Data yang digunakan untuk Tilawati, materi, dan kuis memiliki struktur hierarkis dan bervariasi. Hal ini memerlukan penyesuaian khusus dalam proses *parsing* dan penanganan data agar dapat ditampilkan sesuai urutan dan format yang diharapkan.

3. Implementasi Supabase Edge Function

Supabase tidak mengizinkan pembuatan akun baru secara langsung dari sisi *client*. Solusinya adalah dengan mempelajari dan mengimplementasikan Supabase Edge Function. Namun, proses ini cukup menantang karena membutuhkan pemahaman tentang *environment serverless* dan *deployment* berbasis Deno.

4. Masalah UI dan Responsivitas

Desain antarmuka perlu disesuaikan untuk berbagai ukuran layar, khususnya pada perangkat dengan resolusi kecil. Beberapa komponen UI harus dioptimalkan agar tetap nyaman dilihat dan digunakan oleh siswa TPQ.

5. Manajemen Token dan Autentikasi

Selama pengembangan, beberapa kali terjadi kesalahan terkait token Supabase, seperti token kadaluwarsa atau konfigurasi *environment* yang belum tepat. Hal ini mengganggu proses testing fitur *login* dan pembuatan akun.

6. Keterbatasan Waktu dan Perangkat Uji

Proses pengujian sistem dilakukan secara terbatas, hanya menggunakan satu perangkat Android dengan ukuran layar tertentu. Hal ini menyebabkan pengujian tidak mencakup berbagai variasi perangkat dan resolusi layar lainnya. Selain itu, penelitian mengalami keterlambatan karena perangkat utama (laptop) yang digunakan untuk proses pengembangan mengalami kerusakan. Kondisi ini mengharuskan peneliti untuk menunggu proses perbaikan dalam kurun waktu yang cukup lama, sehingga berdampak pada mundurnya jadwal pelaksanaan penelitian secara keseluruhan.

4.5.5 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur yang telah dikembangkan pada aplikasi belajar Tilawati berjalan sesuai dengan tujuan perancangan. Pengujian juga bertujuan untuk mengevaluasi sejauh mana sistem mampu memenuhi kebutuhan pengguna dari sisi fungsionalitas, antarmuka, dan integrasi data baik secara *offline* maupun *online*.

Metode yang digunakan dalam pengujian ini adalah *Black-box Testing*, yaitu metode yang berfokus pada hasil keluaran dari sistem tanpa menguji struktur internal kode. Pengujian dilakukan pada setiap fitur utama secara bertahap setelah penyelesaiannya dalam masing-masing *sprint*.

1. Lingkungan dan Perangkat Uji

- a. Platform: Android
- b. Perangkat: *Smartphone* Android versi 11
- c. Tools: *Flutter Debugger*, *Supabase Dashboard*, *Postman* (untuk uji Edge Function)
- d. Kondisi Koneksi: *Offline* untuk data lokal (JSON), dan *online* untuk login, hasil kuis, serta fitur akun

2. Fitur yang Diuji

- a. Login & Autentikasi

Melakukan pengujian form login, validasi email/sandi, dan navigasi sukses.

- b. Tilawati

Melakukan pengujian menampilkan daftar jilid, navigasi ke halaman detail, dan penyimpanan halaman terakhir.

- c. Materi

Melakukan pengujian daftar topik dan tampilan isi materi dengan struktur konten.

- d. Kuis

Melakukan pengujian menampilkan soal, validasi jawaban, penyimpanan nilai, dan menampilkan hasil.

- e. Profil

Melakukan pengujian melihat dan memperbarui nama serta kata sandi pengguna.

- f. Tambah *User* (Admin)

Melakukan pengujian *input* dan *submit form* tambah *user*, pemanggilan Edge Function, dan verifikasi ke Supabase.

Seluruh fitur diuji secara fungsional dan hasilnya menunjukkan bahwa sistem dapat berjalan dengan stabil. Data JSON dapat diakses dengan cepat dalam kondisi *offline*, sedangkan proses autentikasi dan penyimpanan daring ke Supabase berjalan lancar.

3. Pengujian Sistem Berdasarkan Kuesioner Pengguna

Sebagai bagian dari proses evaluasi sistem, pengujian aplikasi juga dilakukan dengan melibatkan pengguna akhir, yaitu guru dan siswa dari Taman Pendidikan Al Quran (TPQ). Pengujian dilakukan melalui formulir kuesioner berbasis *Google Form* yang berisi pernyataan-pernyataan mengenai kemudahan penggunaan, tampilan antarmuka, kelengkapan materi, serta efektivitas fitur kuis. Hasil tanggapan pengguna digunakan sebagai dasar untuk mengukur tingkat penerimaan dan kelayakan aplikasi secara praktis di lingkungan pembelajaran. Untuk melihat detail tanggapan dari setiap responden, termasuk pilihan jawaban dan saran yang diberikan, hasil lengkap kuesioner telah disertakan pada bagian lampiran laporan ini.

1. Jumlah Responden

Total responden yang mengisi formulir sebanyak 13 orang, terdiri dari:

- a. 3 guru TPQ
- b. 10 siswa kelas Tilawati 4, 5, dan 6

2. Pertanyaan kuesioner

Adapun daftar pertanyaan yang diberikan kepada responden adalah sebagai berikut:

- a. Aplikasi mudah digunakan dan tidak membingungkan.
Mengukur kemudahan penggunaan antarmuka oleh pengguna umum.
- b. Tampilan aplikasi menarik dan nyaman dilihat.
Menilai kualitas visual dan kenyamanan desain antarmuka.
- c. Materi Tilawati ditampilkan dengan jelas dan sesuai dengan buku.

Menguji kesesuaian konten Tilawati dengan referensi cetak.

- d. Fitur kuis membantu saya menguji pemahaman materi.

Mengukur efektivitas kuis sebagai alat evaluasi mandiri.

- e. Saya ingin terus menggunakan aplikasi ini untuk belajar Tilawati.

Menggambarkan keinginan responden untuk menggunakan aplikasi secara berkelanjutan.

Selain itu, kuesioner juga menyediakan kolom pertanyaan terbuka yang memungkinkan responden memberikan saran, masukan, atau kritik terhadap aplikasi.

3. Hasil Kuesioner (Rangkuman)

Berdasarkan jawaban skala pada setiap pernyataan, diperoleh kesimpulan sebagai berikut:

- a. Sebagian besar responden menyatakan aplikasi mudah digunakan dan tidak membingungkan, dengan mayoritas menjawab Setuju.
- b. Tampilan aplikasi dinilai nyaman dan menarik, dengan mayoritas menjawab Setuju.
- c. Materi Tilawati dinilai cukup lengkap, meskipun terdapat satu tanggapan Netral.
- d. Fitur kuis dinilai membantu siswa menguji pemahaman materi.
- e. Semua siswa menyatakan Setuju untuk terus menggunakan aplikasi ini dalam proses belajar Tilawati.

Beberapa tanggapan terbuka memberikan *insight* penting untuk pengembangan aplikasi ke depan:

1. Guru menyarankan penambahan fitur untuk melihat hasil tugas siswa, sehingga guru dapat melakukan pemantauan secara langsung terhadap aktivitas dan progres siswa.
2. Salah satu siswa menyarankan peningkatan tampilan Tilawati, agar tampilannya lebih jelas dan mudah dinavigasi.

Secara keseluruhan, hasil pengujian menunjukkan bahwa aplikasi telah diterima dengan baik oleh guru dan siswa sebagai media pembelajaran Tilawati. Aplikasi dinilai mudah digunakan, tampilannya menarik, dan fitur-fiturnya relevan dengan kebutuhan belajar. Namun, beberapa saran pengembangan, seperti fitur *monitoring* tugas dan perbaikan antarmuka Tilawati, menjadi catatan penting untuk pengembangan versi aplikasi berikutnya agar lebih sesuai dengan kebutuhan pengajaran di TPQ.

PERPUSTAKAAN
UNIVERSITAS JENDERAL ACHMAD YANI
YOGYAKARTA