

BAB 3

METODE PENELITIAN

Bab ini menjelaskan metodologi penelitian yang digunakan untuk mengembangkan aplikasi belajar Tilawati berbasis mobile. Metode yang digunakan meliputi *Design Thinking* pada tahap analisis dan perancangan, serta *Agile Development* untuk proses pengembangan sistem. Bab ini juga mencakup bahan dan alat penelitian, alur penelitian, serta proses pengumpulan data dan pengujian sistem.

Pemilihan metode *Design Thinking* dan *Agile Software Development* dalam penelitian ini didasarkan pada karakteristik proyek yang menekankan pada pemahaman mendalam terhadap kebutuhan pengguna serta fleksibilitas tinggi dalam proses pengembangan perangkat lunak.

Design Thinking dipilih karena berfokus pada pengguna dan merupakan pendekatan yang berorientasi pada pengguna (*user-centered design*) dengan menempatkan empati sebagai dasar dari proses analisis dan perancangan. Metode ini digunakan pada tahap awal penelitian untuk menggali kebutuhan pengguna aplikasi secara langsung, terutama siswa dan guru TPQ yang memiliki kebutuhan spesifik dalam pembelajaran Al Quran metode Tilawati.

Proses observasi, wawancara, serta pengujian terhadap prototipe awal dilakukan secara iteratif untuk memastikan bahwa solusi yang dikembangkan sesuai dengan konteks penggunaan sebenarnya. Dalam konteks penelitian ini, *Design Thinking* dipilih karena:

1. Membantu penelitian untuk memahami akar masalah dari perspektif pengguna akhir.
2. Mendorong solusi yang tidak hanya fungsional, tetapi juga *usable* dan sesuai dengan kondisi lapangan (misalnya keterbatasan media belajar konvensional).
3. Cocok digunakan dalam tahap perancangan awal produk digital pendidikan berbasis mobile.

Sementara itu, metode *Agile Software Development* digunakan karena iteratif dan fleksibel serta inkremental dalam pembangunan sistem. *Agile* sangat sesuai digunakan dalam proyek pengembangan aplikasi edukasi seperti ini karena:

1. Iterasi Cepat karena aplikasi dibangun dalam beberapa *sprint* yang masing-masing fokus pada satu atau dua fitur inti. Hal ini mempercepat waktu pengembangan dan memungkinkan pengujian awal secara langsung.
2. Responsif terhadap memungkinkan perubahan kebutuhan selama proses pengembangan, misalnya penyesuaian tampilan berdasarkan *feedback* dari pengguna TPQ.
3. Umpan balik langsung di setiap siklus pengembangan sehingga memungkinkan tim untuk mendapatkan umpan balik dari pengguna yang kemudian dijadikan dasar penyempurnaan fitur.
4. Fokus pada fitur prioritas dalam konteks aplikasi pembelajaran seperti Tilawati sehingga memudahkan peneliti untuk membangun fitur yang paling penting terlebih dahulu (misalnya tampilan materi dan kuis), sebelum menambahkan fitur lanjutan seperti akun guru atau fitur lainnya.

Dengan kombinasi kedua metode ini, penelitian ini tidak hanya fokus pada implementasi teknis, tetapi juga menekankan pendekatan humanistik dalam pengembangan aplikasi pendidikan. Hasil akhirnya diharapkan tidak hanya berupa sistem yang berjalan dengan baik, tetapi juga benar-benar membantu siswa dan guru dalam proses belajar mengajar Al Quran secara digital.

Metode penelitian yang ada yang akan dilakukan di dalam penelitian ini menggunakan metode *design thinking* pada proses analisis kebutuhan dan perancangan serta menggunakan metode *Agile* pada seluruh proses pengembangan aplikasi. Dengan metode *Agile* menjadikan proses dari pengembangan aplikasi menjadi lebih cepat dalam jangka waktu yang relatif pendek dan dapat melakukan pengerjaan yang harus diperbaiki secara berulang-ulang.

Penelitian yang akan dilakukan ini adalah penelitian tentang proses pengembangan perangkat lunak. Penelitian dimulai dari analisis kebutuhan dari

masalah yang ada di latar belakang serta mencari informasi dari narasumber, kemudian memetakan proses pengembangan yang akan dilakukan, mengumpulkan data dari sumber yang kredibel untuk bahan materi pembelajaran, dan setelah itu melakukan perancangan dan pengembangan aplikasi belajar Tilawati.

3.1 BAHAN DAN ALAT PENELITIAN

3.1.1 Bahan Penelitian

Bahan penelitian yang digunakan dalam pengembangan sistem aplikasi Tilawati ini diperoleh secara langsung dari modul cetak buku Tilawati yang umum digunakan dalam lembaga pendidikan Al Qur'an, seperti TPQ (Taman Pendidikan Al Quran) dan Madrasah Diniyah. Modul ini memuat materi pembelajaran membaca Al-Qur'an secara bertahap yang meliputi latihan pengenalan huruf hijaiyyah, penyusunan kata, hingga latihan membaca ayat-ayat pendek dengan berbagai hukum tajwid dasar.

Berbeda dengan teks Al-Qur'an seperti Juz 'Amma yang telah banyak tersedia dalam bentuk digital (*dataset* terbuka), hingga saat penelitian ini dilakukan, tidak tersedia *dataset* resmi atau terbuka untuk konten pembelajaran Tilawati dalam format digital. Oleh karena itu, pengumpulan data dilakukan secara mandiri dengan cara menyalin isi buku Tilawati menggunakan alat bantu pengenalan teks (OCR) melalui aplikasi Google Lens. Proses ini dilakukan secara sistematis, yaitu dengan memindai halaman demi halaman, kemudian hasil teks diekstraksi dan dikompilasi ke dalam format digital.

Setelah proses ekstraksi, teks yang dihasilkan kemudian melalui tahap verifikasi manual (*proofreading*) untuk memastikan keakuratan tulisan Arab, termasuk penulisan harakat dan tanda baca yang sangat krusial dalam pembelajaran Tilawati. Selanjutnya, data disusun dalam format *JavaScript Object Notation* (JSON) untuk dapat digunakan dalam pengembangan sistem aplikasi.

Struktur data JSON yang dibuat terdiri atas beberapa elemen penting, antara lain:

1. ID_Halaman yang menunjukkan halaman ke berapa dalam modul Tilawati.

2. Tingkatan Tilawati yang menunjukkan kategori atau jilid modul.
3. Daftar ayat atau baris latihan: berupa kumpulan teks Arab dalam satu halaman yang digunakan untuk latihan membaca.
4. Pertanyaan kuis: berisi soal-soal pilihan ganda yang disusun berdasarkan teks yang diambil dari materi.

Data ini digunakan untuk mendukung beberapa fitur utama dalam sistem aplikasi, yaitu:

1. Tampilan materi Tilawati per halaman.
2. Navigasi antar halaman berdasarkan jilid Tilawati.
3. Fitur kuis interaktif yang bertujuan menguji pemahaman pengguna terhadap materi yang telah dibaca.

Dengan demikian, bahan penelitian ini diperoleh secara langsung dari sumber asli dan diolah sendiri oleh peneliti untuk dijadikan data set internal, yang tidak hanya meningkatkan orisinalitas penelitian tetapi juga menunjukkan kontribusi terhadap digitalisasi materi pembelajaran Tilawati yang sebelumnya belum tersedia dalam bentuk data digital terstruktur.

3.1.2 Alat Penelitian

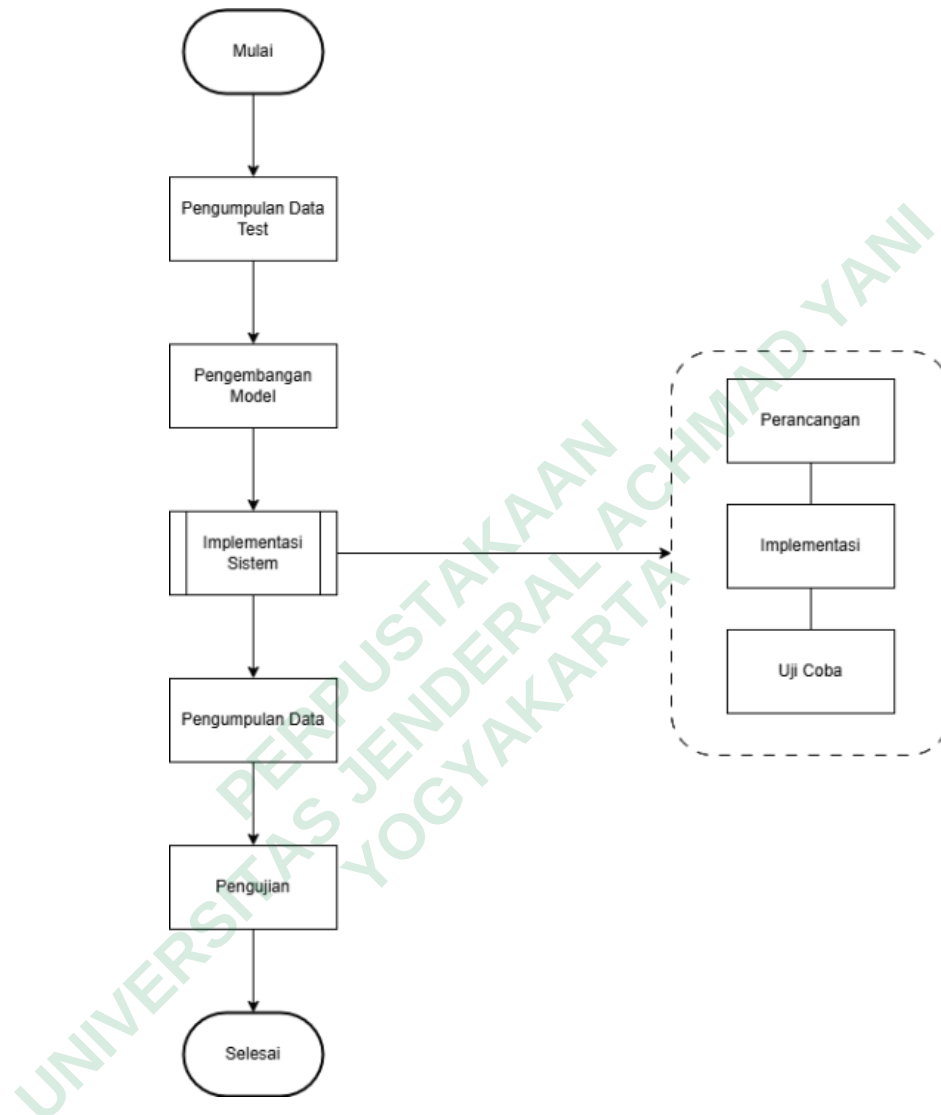
Alat yang digunakan dalam penelitian ini adalah sebuah perangkat laptop dengan spesifikasi cukup untuk menjalankan sistem operasi dan perangkat lunak pengembangan serta koneksitas internet.

Sistem Operasi dan program-program aplikasi yang dipergunakan dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi : Windows 11 atau lebih baru, dan Linux Ubuntu versi 23.04 atau lebih baru.
2. Aplikasi IDE : Visual Studio Code versi 1.90 dan Android Studio versi 2024.1.1 atau versi terbaru.
3. Bahasa pemrograman : Bahasa pemrograman Dart menggunakan *framework Flutter* versi 3.22.2 atau versi terbaru.
4. *Database engine* : *Supabase database* untuk data *Online* dan JSON untuk data lokal

3.2 JALAN PENELITIAN

Berikut merupakan alur pengembangan sistem yang akan diimplementasikan pada penelitian yang ditampilkan pada gambar berikut.



Gambar 3.1 Alur Penelitian

3.2.1 Pengumpulan Data *Test*

Untuk pengembangan sistem aplikasi Tilawati berbasis digital, diperlukan serangkaian data awal yang akan digunakan sebagai bahan uji coba dalam proses pembuatan dan pengujian fitur-fitur utama aplikasi. Salah satu langkah awal yang dilakukan adalah pengumpulan data *test*, yaitu pengambilan sebagian data dari keseluruhan materi Tilawati untuk dijadikan sampel pengujian.

Tujuan utama dari kegiatan ini adalah untuk:

1. Menyediakan data representatif yang mencakup struktur konten Tilawati seperti ID_Halaman, nomor Tilawati, dan daftar ayat per halaman.
2. Memastikan format dan struktur data sesuai dengan kebutuhan sistem, terutama dalam pengolahan data JSON dan proses navigasi per halaman.
3. Mendukung proses perancangan dan validasi fitur seperti: tampilan daftar Tilawati, detail halaman, navigasi antar halaman, serta pencatatan progres pengguna.
4. Menghindari penggunaan data penuh yang berisiko membebani sistem saat fase awal pengembangan.

Data *test* ini dikumpulkan dari sumber asli modul Tilawati dan disusun ulang dalam bentuk digital, dengan menyesuaikan pada struktur JSON. Sebagai contoh, data *test* mencakup 5–10 halaman Tilawati dari beberapa level (misalnya Tilawati 1 hingga Tilawati 3), yang masing-masing terdiri dari ID_Halaman dan daftar ayat yang ditampilkan dalam format baris dan teks.

Dengan adanya data *test* ini, tim pengembang dapat melakukan:

1. Simulasi navigasi dan tampilan detail ayat
2. Uji coba performa pengambilan data lokal
3. Pengembangan fitur *mark* halaman, serta integrasi dengan *backend* (jika diperlukan)

Pengumpulan data *test* ini bersifat sementara dan terbatas, dan akan diperluas ke seluruh data Tilawati setelah sistem aplikasi memasuki tahap pengujian menyeluruh (*end-to-end testing*).

3.2.2 Pengembangan Model

Setelah data *test* diperoleh dan diolah, tahapan selanjutnya adalah pengembangan model sistem. Model ini mencakup:

1. Struktur data yang menyimpan informasi halaman, ayat, dan level Tilawati.
2. Desain alur interaksi pengguna, seperti bagaimana pengguna dapat menavigasi halaman, membaca materi, dan menjawab kuis.
3. Skema relasi antar entitas, terutama jika sistem melibatkan *backend* atau penyimpanan data pengguna.

Pada tahap ini, peneliti menyusun *class diagram*, *use case diagram*, dan *sequence diagram* untuk memvisualisasikan proses dan relasi antar komponen sistem. Pengembangan model menjadi fondasi penting dalam menentukan kebutuhan fitur serta perencanaan implementasi teknis.

3.2.3 Implementasi Sistem

Tahap implementasi dilakukan dengan mengembangkan aplikasi sesuai model yang telah dirancang. Sistem dibangun menggunakan *framework* yang relevan (misalnya Flutter untuk aplikasi *mobile*), serta memanfaatkan data JSON sebagai sumber konten. Pada fase ini, dikembangkan beberapa fitur utama, antara lain:

1. Halaman daftar Tilawati berdasarkan jilid.
2. Tampilan detail halaman yang berisi daftar ayat atau baris per halaman.
3. Navigasi antar halaman.
4. Fitur kuis interaktif yang mengambil soal dari data JSON.

Selain itu, dilakukan juga integrasi penyimpanan lokal atau *backend* (jika diperlukan) untuk mencatat progres pembelajaran pengguna.

3.2.4 Pengumpulan Data

Setelah sistem dasar berhasil dibangun dan diuji dengan data *test*, peneliti melanjutkan proses pengumpulan data dari seluruh isi buku Tilawati. Proses ini mengikuti metode yang sama dengan tahap awal, yaitu menggunakan Google Lens untuk ekstraksi, dilanjutkan dengan verifikasi teks secara manual. Hasil akhir disusun ke dalam struktur JSON yang lebih besar dan lengkap.

Data yang dikumpulkan meliputi:

1. Semua halaman dari setiap jilid Tilawati.

2. Baris-baris ayat latihan per halaman.
3. Pertanyaan kuis yang dikembangkan dari materi yang ada.
4. Materi yang ada dalam buku modul cetak.

Tahap ini dilakukan agar aplikasi dapat berjalan dengan cakupan materi utuh sebagaimana yang digunakan dalam pembelajaran Tilawati secara konvensional.

3.2.5 Pengujian

Setelah sistem lengkap dibangun dengan data yang lebih luas, dilakukan proses pengujian untuk mengevaluasi kualitas sistem. Pengujian difokuskan pada:

1. Fungsi sistem, seperti apakah semua halaman dan fitur kuis berjalan sebagaimana mestinya.
2. Kesesuaian tampilan, terutama dalam menampilkan teks Arab secara utuh dan benar.
3. Kinerja sistem, seperti kecepatan akses halaman, ketepatan penyimpanan data, serta ketahanan saat berpindah antar fitur.

Metode pengujian dapat melibatkan teknik *black-box testing*, serta uji coba terbatas kepada calon pengguna seperti guru TPQ atau siswa untuk mendapatkan umpan balik. Tujuannya adalah untuk mengidentifikasi kesalahan (*bugs*), kekurangan desain, atau kebutuhan tambahan dari pengguna.

Dan untuk mengetahui sejauh mana aplikasi dapat diterima dan digunakan secara efektif oleh pengguna akhir, disusun kuesioner yang terdiri dari delapan pernyataan. Pertanyaan dalam kuesioner dirancang menggunakan skala *Likert* (Setuju, Netral, dan Tidak Setuju) dan ditujukan untuk mengukur aspek kenyamanan, kemudahan, serta efektivitas fitur dalam aplikasi.

3.3 METODE PEMODELAN SISTEM DAN DATA

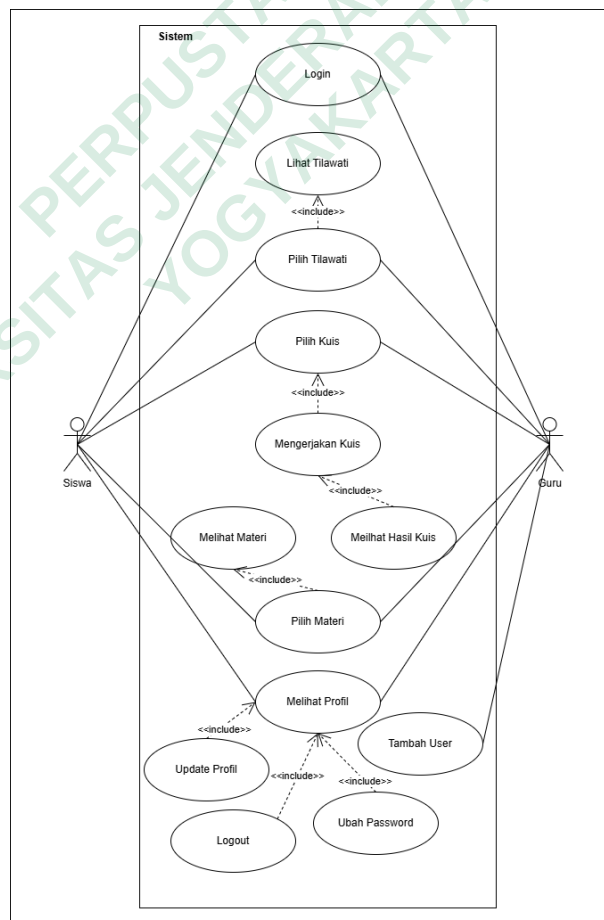
Untuk mendukung proses perancangan dan pengembangan aplikasi belajar Tilawati berbasis mobile, penelitian ini menggunakan pendekatan pemodelan sistem dan struktur data. Pemodelan ini bertujuan untuk memberikan gambaran menyeluruh mengenai interaksi pengguna, struktur internal aplikasi, serta format penyimpanan data yang digunakan.

Beberapa metode yang digunakan dalam pemodelan sistem dan data adalah sebagai berikut:

3.3.1 Use Case Diagram

Use Case Diagram menggambarkan interaksi antara aktor (pengguna) dengan sistem. Diagram ini mengilustrasikan fitur-fitur utama yang tersedia dalam aplikasi, seperti:

- Melihat daftar materi Tilawati per jilid.
- Mengakses halaman detail materi.
- Menjawab soal kuis.
- Melihat hasil kuis.
- Login/logout akun pengguna.
- Update profil dan kata sandi (khusus pengguna login).
- Admin dapat menambahkan user.



Gambar 3.2 Use Case Diagram.

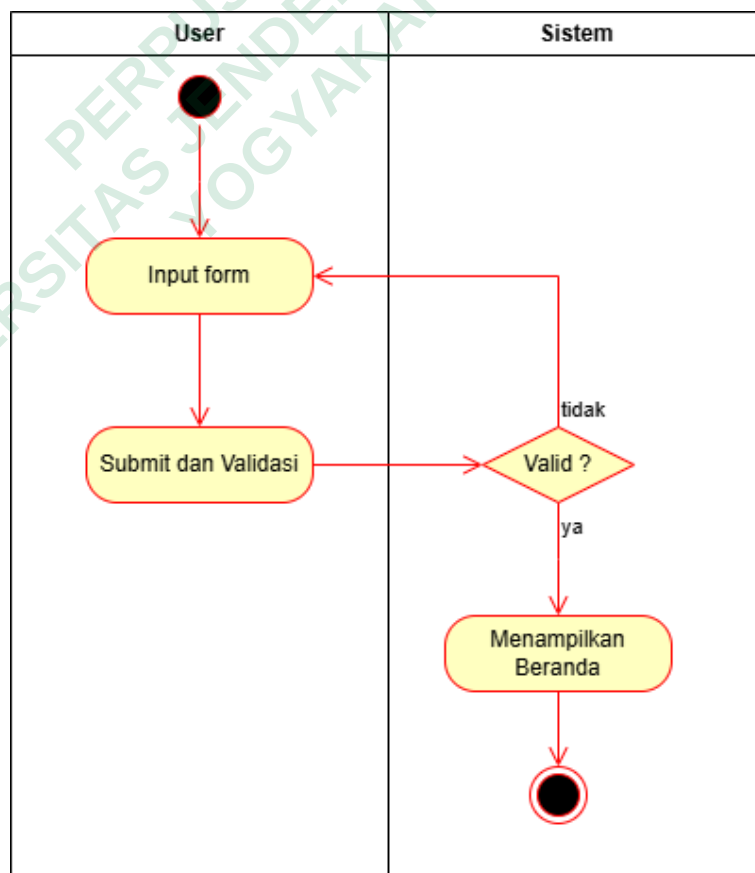
Use case diagram ini berfungsi sebagai dasar dalam mengidentifikasi kebutuhan fungsional dari sistem.

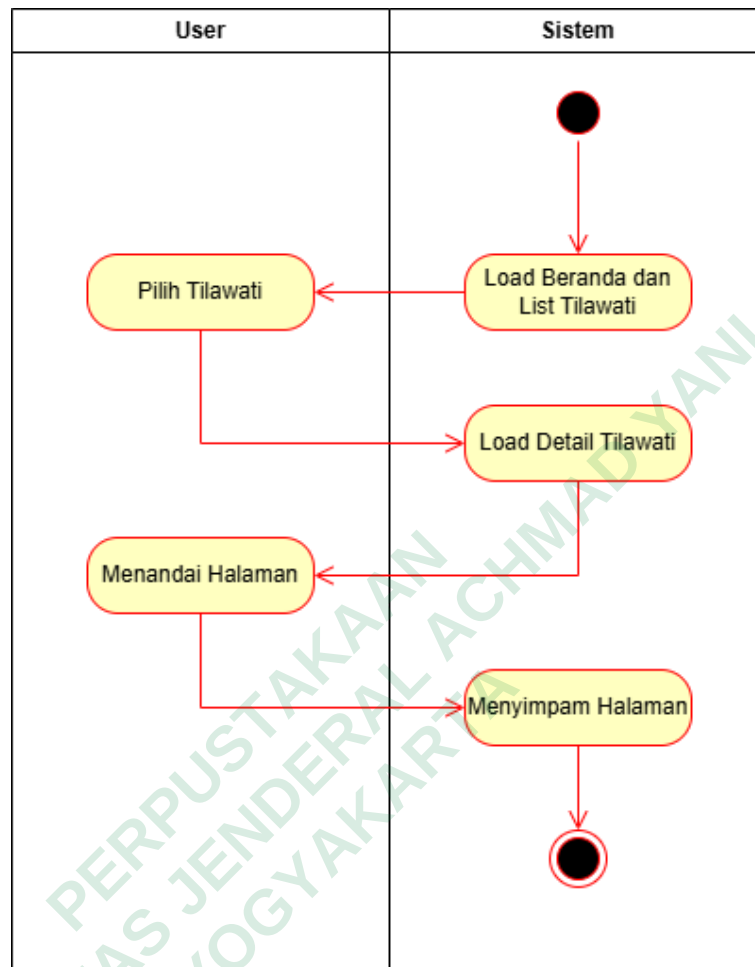
3.3.2 Activity Diagram

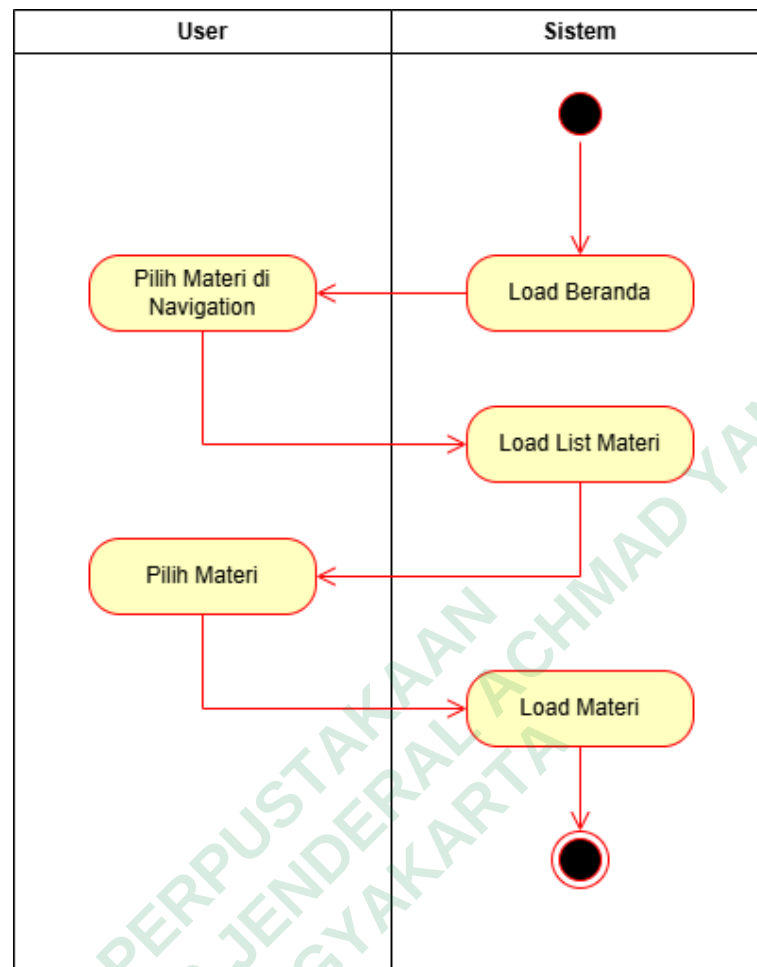
Activity diagram digunakan untuk menggambarkan alur logika dari proses-proses penting di aplikasi, antara lain:

- Alur login dan logout.
- Proses navigasi materi dan kuis.
- Proses pembuatan user baru oleh admin.
- Update data pengguna (profil dan password).

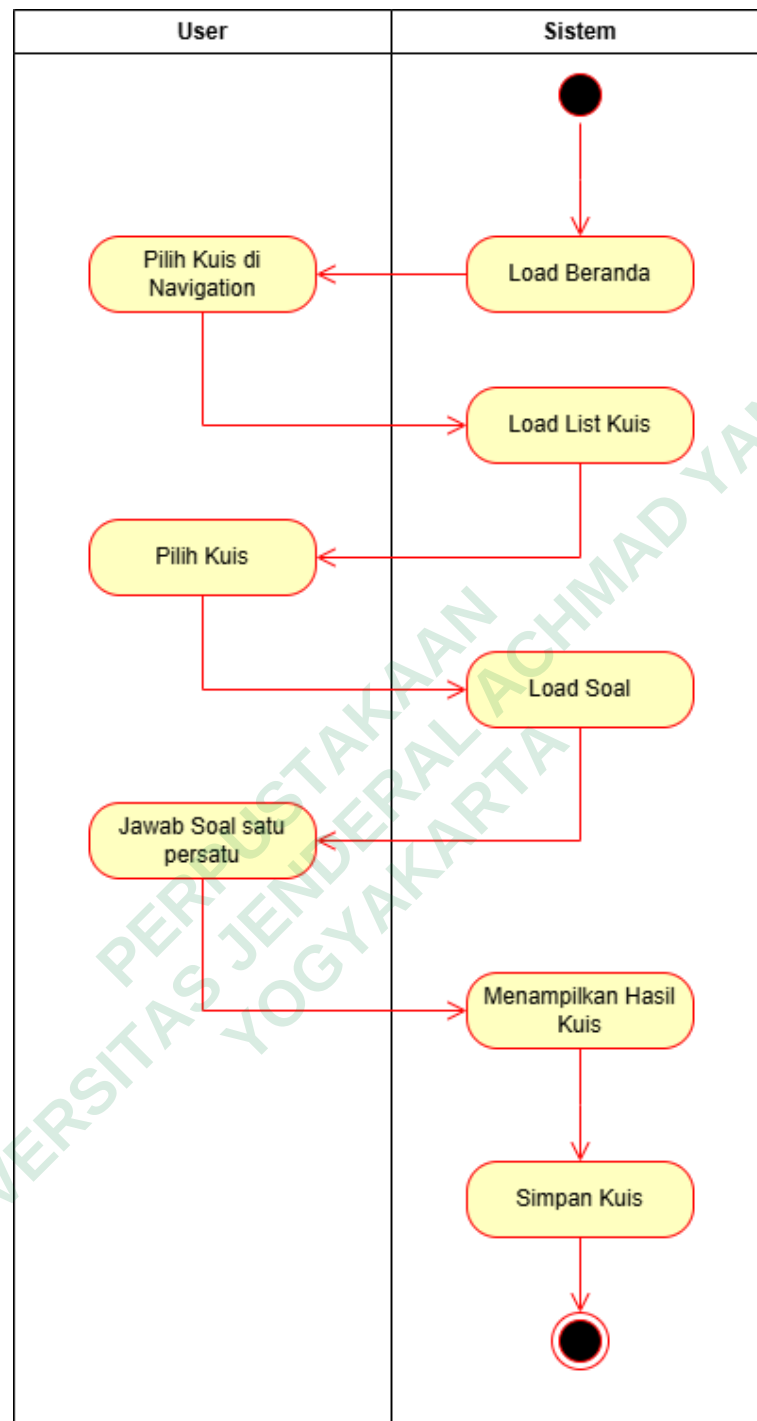
Diagram ini memudahkan dalam memahami logika sistem dari sudut pandang pengguna dan pengembang. Pemodelan *activity* diagram pada penelitian ini seperti gambar berikut:



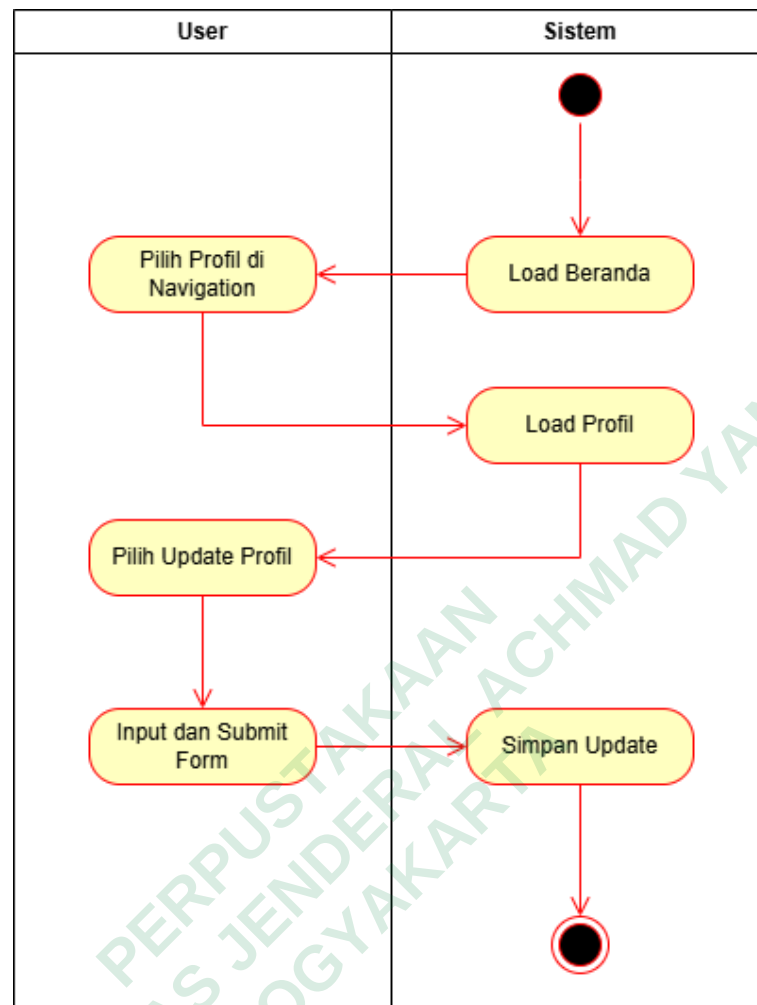
Gambar 3.3 Activity Diagram Login**Gambar 3.4** Activity Diagram Tilawati



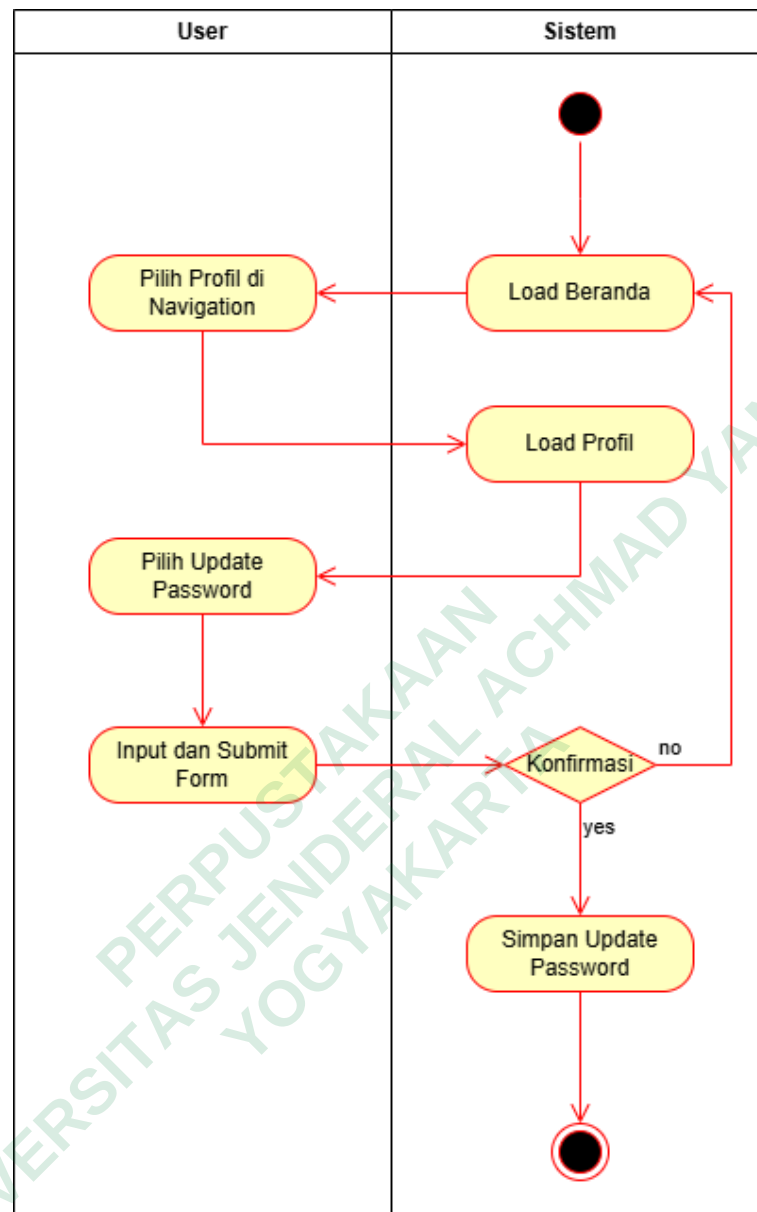
Gambar 3.5 Activity Diagram Materi



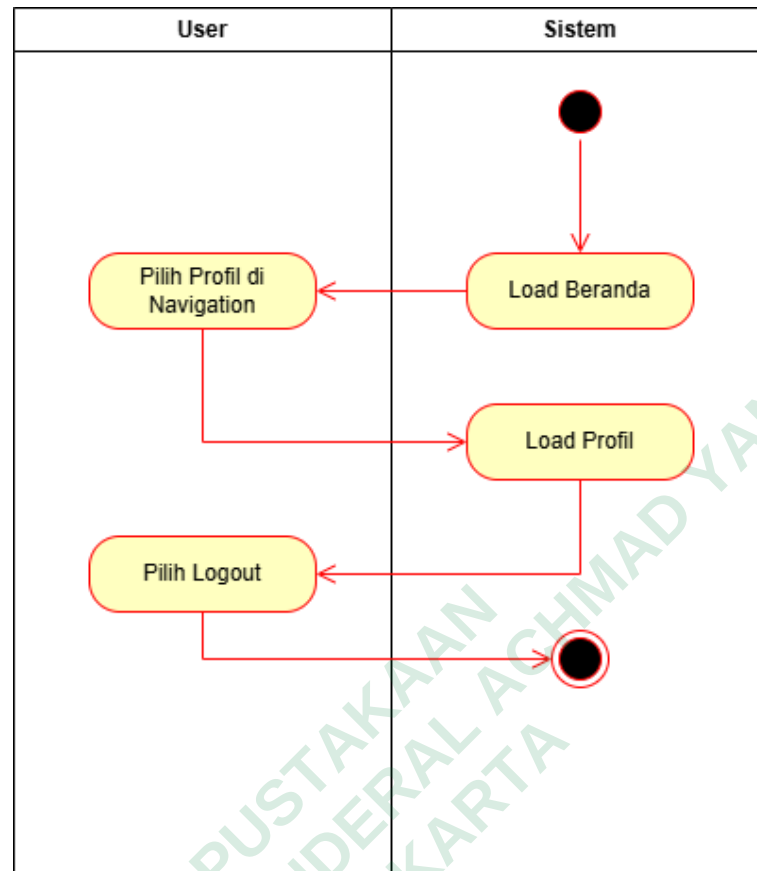
Gambar 3.6 Activity Diagram Kuis



Gambar 3.7 Activity Diagram Update Profil



Gambar 3.8 Activity Diagram Update Password



Gambar 3.9 Activity Diagram Logout

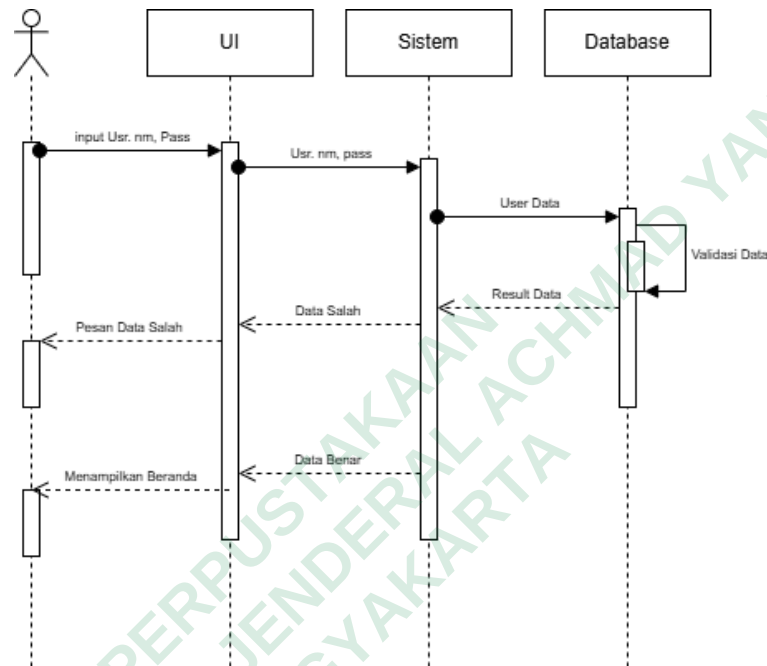
3.3.3 Sequence Diagram

Sequence diagram pada penelitian ini menjelaskan urutan proses dan interaksi antar komponen sistem saat pengguna menjalankan fitur tertentu. Beberapa sequence diagram yang dibuat:

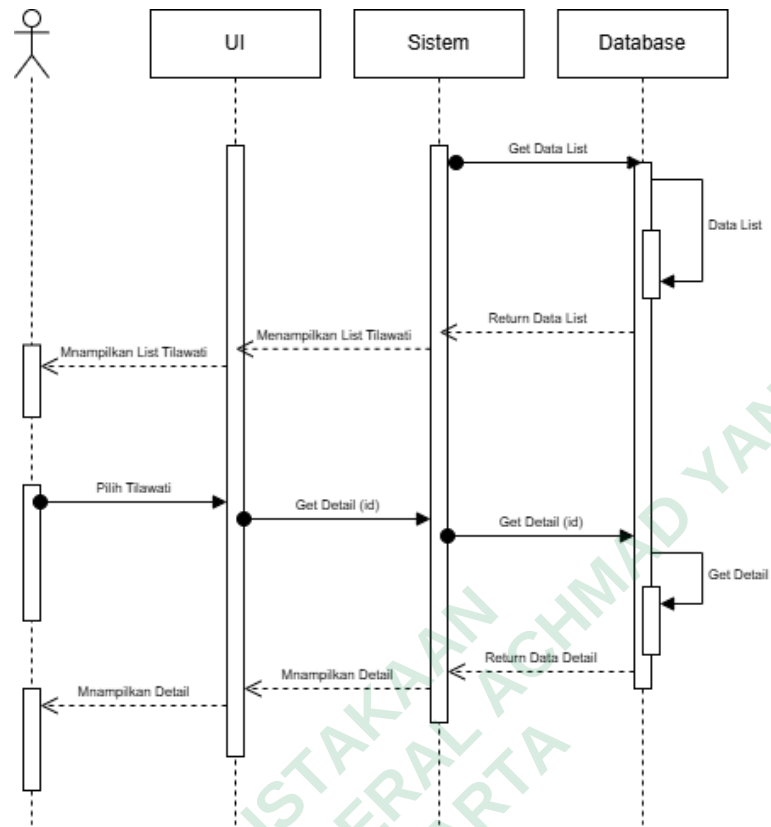
1. Login: Proses login dari input email/*password*, verifikasi ke Supabase, hingga *redirect* ke *dashboard*.
2. Materi Tilawati: Pengguna memilih materi, sistem mengambil data JSON, lalu menampilkan detail halaman.
3. Kuis: Sistem memuat soal dari *file* lokal atau *database*, menampilkan pilihan, menyimpan hasil.
4. *Update* Profil / *Password*: Sistem memverifikasi perubahan, lalu menyimpan ke *backend*.

5. Tambah User (admin): Admin menginput data, sistem memanggil Supabase Edge Function untuk membuat akun.

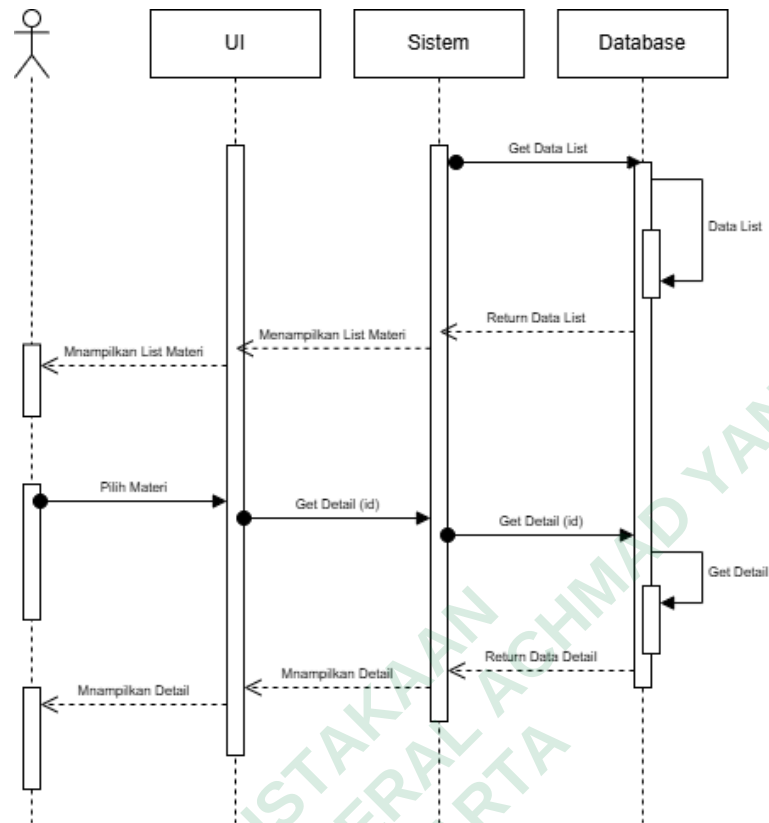
Diagram ini penting untuk memastikan logika alur sistem berjalan sesuai desain. Model *sequence* diagram penelitian ini seperti gambar berikut:



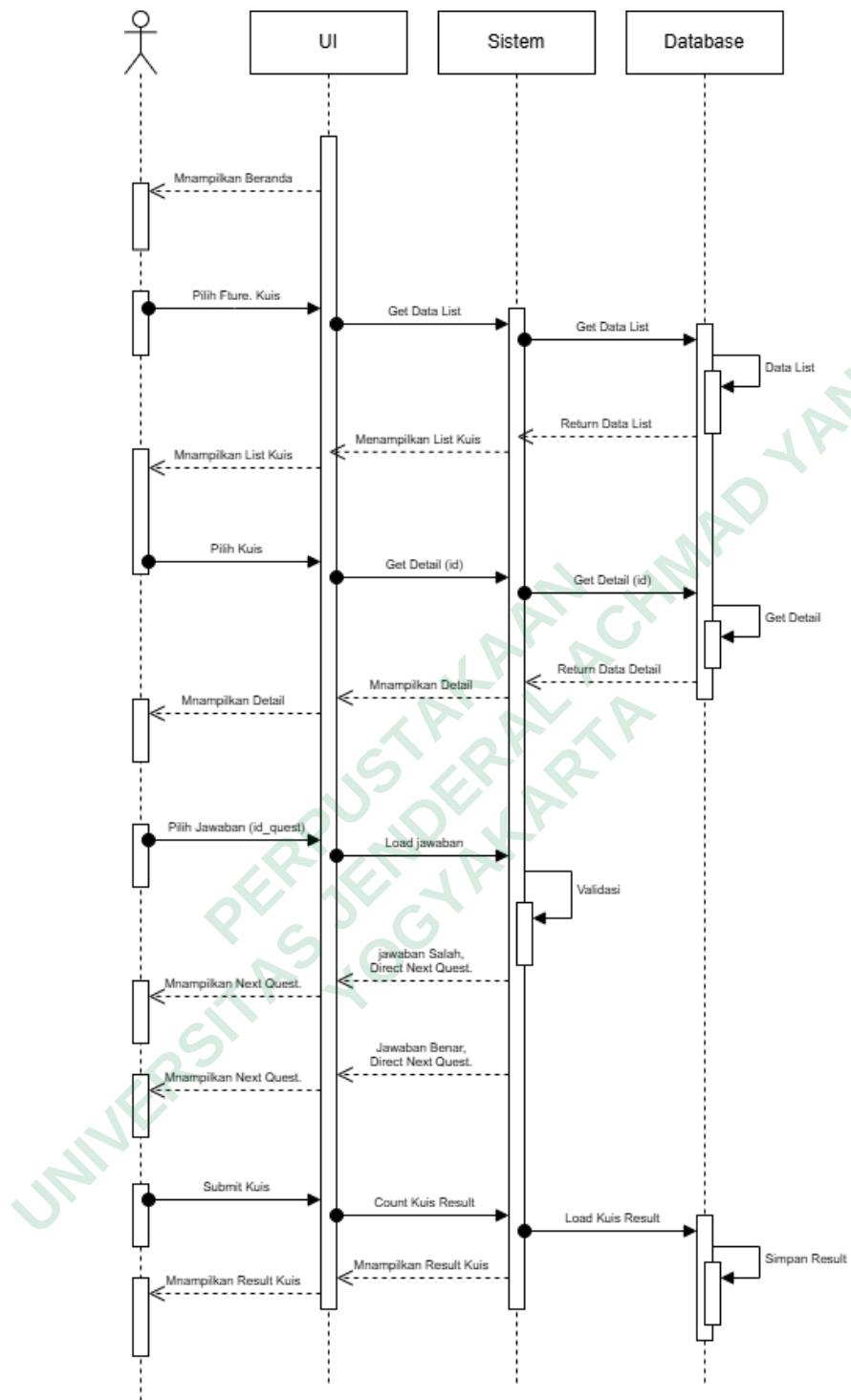
Gambar 3.10 Sequence Diagram Login



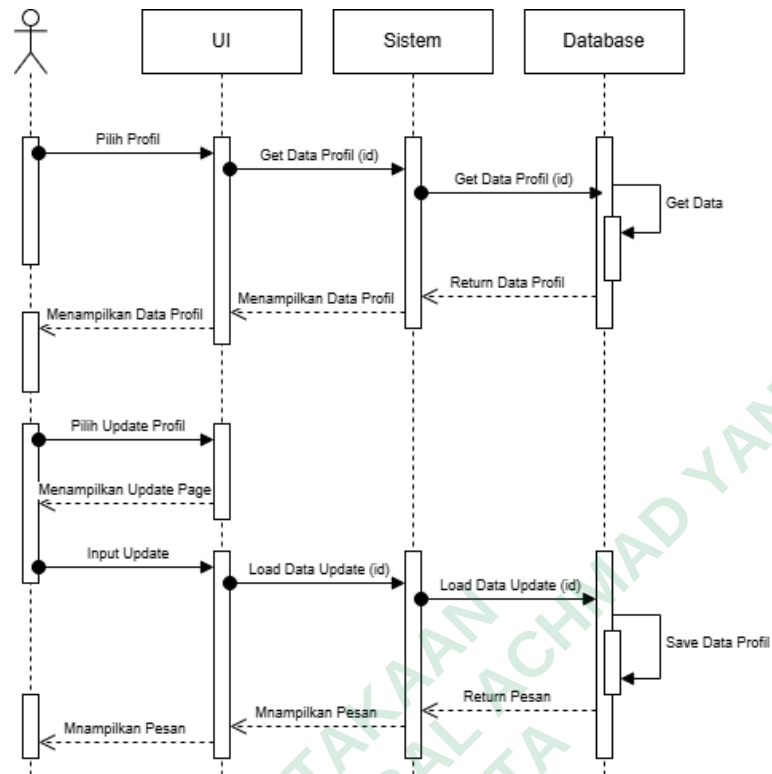
Gambar 3.11 Sequence Diagram Tilawati



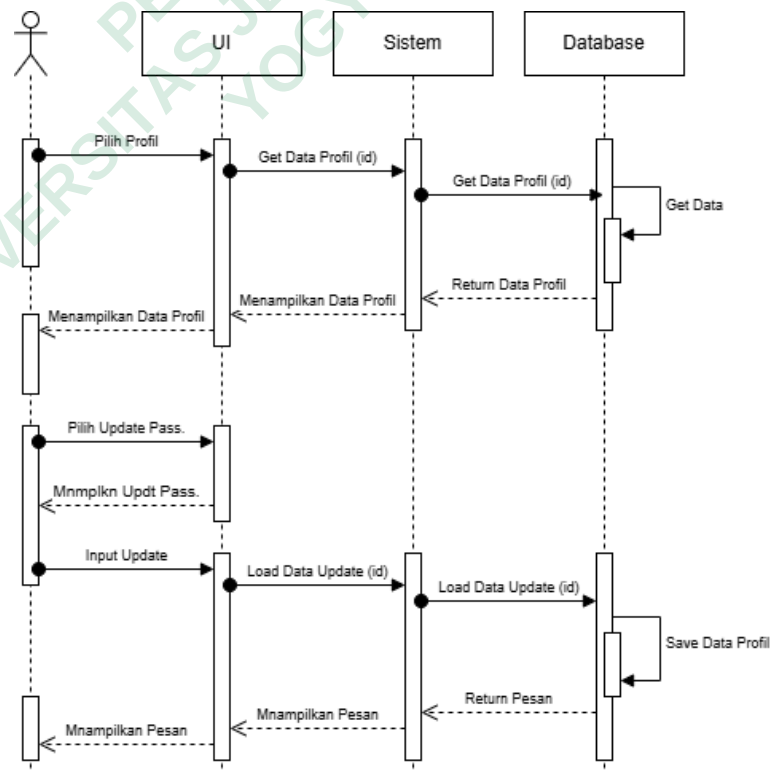
Gambar 3.12 Sequence Diagram Materi



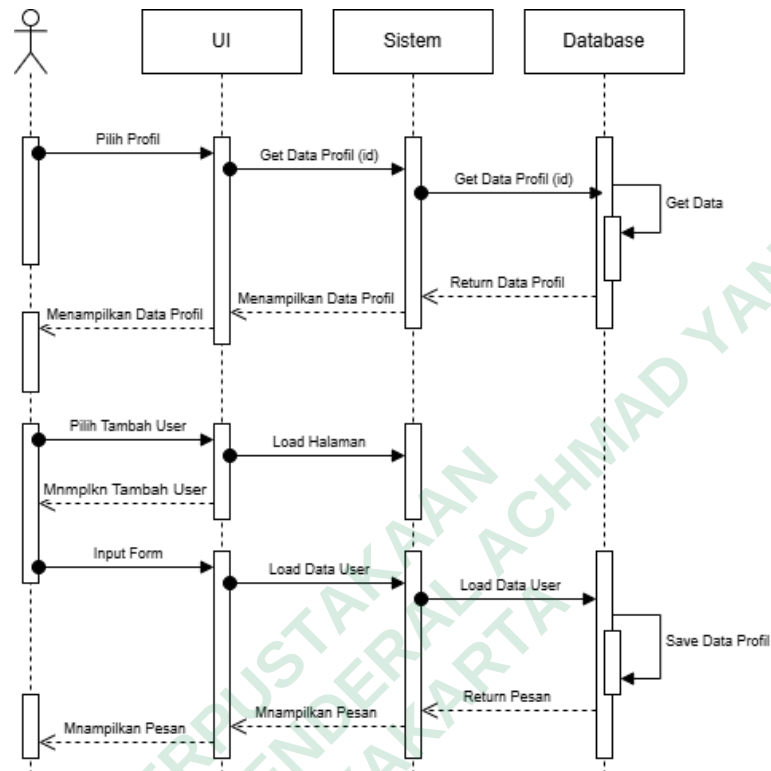
Gambar 3.13 Sequence Diagram Kuis



Gambar 3.14 Sequence Diagram Update Profil



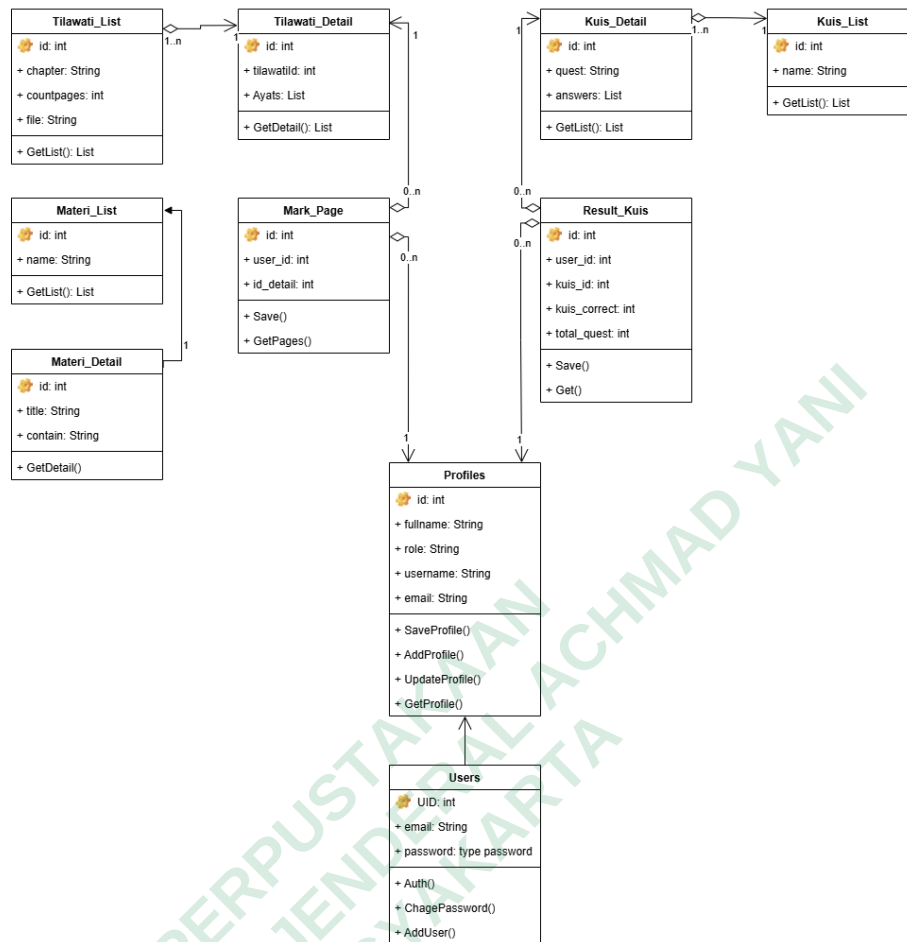
Gambar 3.15 Sequence Diagram Update Password



Gambar 3.16 Sequence Diagram Tambah User

3.3.4 Class Diagram

Class diagram digunakan untuk memodelkan struktur data dan hubungan antar objek. Diagram ini menggambarkan kelas-kelas utama seperti:



Gambar 3.17 Class Diagram

1. User memiliki atribut nama, email, role, password.
2. Materi berisi ID_Materi, title, dan konten.
3. Tilawati berisi ID_Tilawati, jilid, daftar halaman.
4. Kuis berisi pertanyaan, pilihan jawaban, dan penanda benar.
5. MarkPage menyimpan halaman terakhir yang dibaca.
6. HasilKuis menyimpan skor dan waktu pengerjaan.

Diagram ini menjadi acuan dalam mendesain model data di Flutter maupun model data di Supabase.

3.3.5 Struktur Data JSON

Dalam pengembangan aplikasi belajar Tilawati ini, digunakan format *JavaScript Object Notation* (JSON) sebagai format utama untuk menyimpan dan

memuat data secara lokal. Format JSON dipilih karena ringan, fleksibel, mudah diakses, dan sangat kompatibel dengan pengembangan aplikasi berbasis Flutter.

Seluruh data materi pembelajaran seperti Tilawati, Materi, dan Kuis disimpan dalam bentuk *file* JSON lokal yang dimuat langsung dari direktori *assets*. Pendekatan ini dipilih dengan pertimbangan utama bahwa:

1. Data pembelajaran bersifat statis, tidak berubah-ubah atau bergantung pada akun pengguna.
2. Data dapat langsung diakses tanpa koneksi internet, sehingga aplikasi tetap bisa digunakan secara *offline* oleh siswa di TPQ.
3. Proses pemuatan data lebih cepat, karena tidak memerlukan permintaan ke server eksternal seperti Supabase.
4. Meningkatkan performa dan efisiensi, terutama saat membuka halaman Tilawati atau mengerjakan kuis, karena seluruh data sudah tersedia di perangkat pengguna.

Berikut adalah struktur JSON yang digunakan dalam masing-masing fitur:

3.3.5.1 Data JSON Tilawati

Data JSON pada fitur Tilawati terbagi menjadi dua jenis, yaitu Tilawati *List* dan Tilawati Detail. Tilawati *List* digunakan untuk menyimpan informasi umum seperti ID Tilawati (*ID_Tilawati*) dan judul setiap jilid Tilawati, serta nama *file* JSON yang berisi detail halaman dari jilid tersebut. Data ini ditampilkan sebagai daftar pilihan bagi pengguna sebelum mengakses materi.

Tilawati Detail digunakan untuk menyimpan isi materi secara lengkap, termasuk ID halaman (*ID_Halaman*), *tilawati_id* sebagai penanda jilid, serta daftar ayat atau baris latihan dalam bentuk teks Arab. Data ini dimuat setelah pengguna memilih jilid tertentu dari daftar Tilawati *List*.

Pemisahan struktur ini bertujuan untuk mempercepat proses pemuatan data dan memudahkan navigasi, karena hanya data yang relevan yang dimuat saat dibutuhkan. Berikut struktur data JSON yang ada pada fitur Tilawati:

1. Tilawati *List*.

```
[
  {
    "id"      : 1,
    "chapter" : "Tilawati 1",
    "countpages": 44,
    "file"    : "detail_tilawati1.json"
  },
  {
    "id"      : 2,
    "chapter" : "Tilawati 2",
    "countpages": 44,
    "file"    : "detail_tilawati2.json"
  },
  ...
]
```

2. Tilawati Detail.

```
[
  {
    "id": 1,
    "tilawati_id": 1,
    "ayats": [
      { "baris": 1, "ayat": "بَ اَ", "type": "konten" },
      { "baris": 2, "ayat": "بَ اَ بَ اَ", "type": "konten" },
      { "baris": 3, "ayat": "اَ بَ اَ بَ", "type": "konten" },
      { "baris": 4, "ayat": "بَ بَ اَ اَ اَ", "type": "konten" },
      { "baris": 5, "ayat": "اَ بَ اَ اَ بَ", "type": "konten" },
      { "baris": 6, "ayat": "بَ اَ بَ بَ اَ", "type": "konten" },
      { "baris": 7, "ayat": "اَب اَب اَب اَب اَب اَب", "type": "keterangan" }
    ]
  },
  {
    ...
  }
]
```

```

1  {
2  {
3    "id"      : 1,
4    "chapter" : "Tilawati 1",
5    "countpages": 44,
6    "file"    : "detail_tilawati1.json"
7  },
8  },
9  > { ...
14 },
15 },
16 > { ...
21 },
22 },
23 > { ...
28 },
29 },
30 > { ...
35 },
36 },
37 > { ...
42 },
43 }

```

You, last month • change 6 Juni ...

Gambar 3.18 Struktur data JSON Tilawati *List*

```

1  {
2  {
3    "id": 1,
4    "tilawati_id": 1,
5    "ayats": [
6      { "baris": 1, "ayat": "بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ", "type": "konten" },
7      { "baris": 2, "ayat": "أَلَمْ يَجْعَلْ لَكُمْ سُبُلًا", "type": "konten" },
8      { "baris": 3, "ayat": "فَإِنْ كُنْتُمْ تُحِبُّونَ اللَّهَ", "type": "konten" },
9      { "baris": 4, "ayat": "فَتَقَبَّلْوهُنَّ عَلَى الْفُتُوحِ", "type": "konten" },
10     { "baris": 5, "ayat": "أَلَمْ يَجْعَلْ لَكُمْ سُبُلًا", "type": "konten" },
11     { "baris": 6, "ayat": "فَإِنْ كُنْتُمْ تُحِبُّونَ اللَّهَ", "type": "konten" },
12     { "baris": 7, "ayat": "فَتَقَبَّلْوهُنَّ عَلَى الْفُتُوحِ", "type": "keterangan" }
13   ]
14 },
15 > { ...
16 },
17 > { ...
18 },
19 > { ...
20 },
21 > { ...
22 },
23 > { ...
24 },
25 > { ...
26 },
27 > { ...
28 },
29 > { ...
30 },
31 > { ...
32 },
33 > { ...
34 },
35 > { ...
36 },
37 > { ...
38 },
39 > { ...
40 },
41 > { ...
42 },
43 > { ...
44 },
45 > { ...
46 },
47 > { ...
48 },
49 > { ...
50 },
51 > { ...
52 },
53 > { ...
54 },
55 > { ...
56 },
57 }

```

You, 4 weeks ago • change 11 Juni ...

Gambar 3.19 Struktur data JSON Tilawati Detail

3.3.5.2 Data JSON Kuis

Fitur kuis pada aplikasi belajar Tilawati dirancang untuk membantu pengguna dalam menguji pemahaman terhadap materi yang telah dipelajari. Untuk mendukung fitur ini, digunakan dua jenis struktur data JSON yang berbeda, yaitu data Kuis *List* dan data Kuis Detail. Pemisahan struktur ini dilakukan untuk mempermudah pengelolaan soal, serta mempercepat proses pemuatan data berdasarkan pilihan kuis yang diakses oleh pengguna.

1. Kuis *List*

Kuis *List* merupakan struktur data yang berisi daftar seluruh kumpulan soal yang tersedia dalam aplikasi. Setiap item dalam Kuis *List*

merepresentasikan satu paket soal kuis dan memuat informasi penting seperti ID kuis (ID_Kuislist), nama atau judul kuis, serta jumlah soal di dalamnya. Data ini digunakan untuk menampilkan pilihan kuis pada tampilan awal fitur kuis, sehingga pengguna dapat memilih paket soal tertentu sebelum masuk ke tampilan detail.

Berikut adalah contoh struktur data JSON untuk Kuis *List*:

```
[
  {
    "id": 1,
    "name_kuis": "Soal 1 - Tajwid Dasar",
    "jumlah_soal": 15
  },
  {
    "id": 2,
    "name_kuis": "Soal 2 - Tajwid Dasar",
    "jumlah_soal": 15
  },
  {
    ...
  }
]
```

2. Kuis Detail

Kuis Detail merupakan struktur data yang berisi daftar pertanyaan dan jawaban pilihan ganda untuk masing-masing kuis yang tersedia. Setiap entri terdiri dari ID Soal (ID_Soal), teks pertanyaan (kuis_*quest*), serta daftar jawaban (kuis_*answers*). Setiap jawaban memiliki atribut *answer* untuk teks jawaban dan *is_true* untuk menandai apakah jawaban tersebut benar atau tidak.

Contoh struktur data JSON untuk Kuis Detail adalah sebagai berikut:

```
[
  {
    "id": 1,
    "kuis_quest": "Huruf Mad berjumlah...",
    "kuis_answers": [
      {
        "is_true": false,
        "answer": "2 huruf"
      },
      {
        "is_true": true,
        "answer": "3 huruf"
      }
    ]
  }
]
```

```

    {
      "is_true": false,
      "answer": "4 huruf"
    },
    {
      "is_true": false,
      "answer": "5 huruf"
    }
  ]
},
{
  ...
}
]

```

Data ini digunakan untuk menampilkan soal dan pilihan jawaban secara dinamis di dalam aplikasi. Ketika pengguna memilih salah satu jawaban, sistem akan mengevaluasi kebenarannya berdasarkan nilai dari atribut *is_true*.

Dengan menggunakan struktur data ini, proses penambahan dan pengelolaan soal menjadi lebih mudah dan terstruktur. Sistem juga dapat memuat soal secara efisien karena hanya perlu mengambil data dari *file* JSON lokal berdasarkan kuis yang dipilih.

```

1  [
2    {
3      "id"      :1,
4      "name_kuis" : "Soal 1- Tajwid Dasar",
5      "jumlah_soal" : 15
6    },
7    {
8      "id"      :2,
9      "name_kuis" : "Soal 2 - Tajwid Dasar",
10     "jumlah_soal" : 15
11   },

```

Gambar 3.20 Struktur data JSON Kuis *List*

```

1  [
2  {
3    "id": 1,
4    "kuis_quest": "Huruf Mad berjumlah...",
5    "kuis_answers": [
6      {
7        "is_true": false,
8        "answer": "2 huruf"
9      },
10     {
11       "is_true": true,
12       "answer": "3 huruf"
13     },
14     {
15       "is_true": false,
16       "answer": "4 huruf"
17     },
18     {
19       "is_true": false,
20       "answer": "5 huruf"
21     }
22   ]
23 }

```

Gambar 3.21 Struktur data JSON Kuis Detail

3.3.5.3 Data JSON Materi

Fitur materi pada aplikasi belajar Tilawati berfungsi sebagai media penyampaian informasi tambahan berupa teori pembelajaran, hukum bacaan, dan konsep dasar dalam membaca Al Quran. Materi-materi ini mendukung pemahaman siswa terhadap topik yang tidak hanya dipelajari secara praktik, tetapi juga secara konseptual.

Untuk mengelola data materi secara efisien, struktur data JSON pada fitur ini dibagi menjadi dua jenis utama, yaitu Materi *List* dan Materi Detail.

1. Materi *List*

Materi *List* digunakan untuk menampilkan daftar topik materi secara ringkas pada tampilan awal fitur Materi. Setiap item materi dalam daftar ini memuat informasi identitas seperti *id*, judul (*title*), deskripsi singkat (*description*), serta nama *file* JSON yang merujuk pada isi lengkap dari materi tersebut.

Struktur ini memudahkan pengguna untuk memilih topik yang ingin dipelajari, sekaligus memberikan gambaran singkat mengenai isi materi sebelum dibuka.

Berikut adalah contoh struktur data JSON untuk Materi List:

```

[
  {
    "id": 1,
    "title": "Ahkamul Madd",
    "description": "Hukum Bacaan Panjang",
    "file": "materi_detail_1.json"
  }
]

```

```

    },
    {
      "id": 2,
      "title": "Ahkamul Huruf",
      "description": "Hukum Bacaan Huruf",
      "file": "materi_detail_2.json"
    },
    ...
  ]

```

Data ini digunakan dalam halaman daftar materi dan menjadi referensi untuk memuat isi lengkap materi dari *file* JSON eksternal yang sesuai.

2. Materi Detail

Materi Detail merupakan struktur yang menyimpan isi lengkap dari masing-masing topik materi. Setiap materi memuat *id*, *title* sebagai judul utama topik, dan *content* sebagai daftar bagian isi. Setiap item dalam *content* terdiri dari pasangan *subtitle* dan *body*, di mana *subtitle* digunakan untuk menandai sub judul materi, dan *body* berisi penjelasan dalam bentuk paragraf atau daftar.

Contoh struktur data JSON untuk Materi Detail adalah sebagai berikut:

```

[
  {
    "id": 1,
    "title": "Pengertian Madd",
    "content": [
      {
        "subtitle": "Pengertian",
        "body": "Madd ialah bacaan panjang dalam Al-Qur'an disebabkan adanya tanda madd. Tanda-tanda madd yaitu:"
      },
      {
        "subtitle": "",
        "body": "1. Fathah diikuti Alif (..-َ\n2. Kasrah diikuti Yaa Sukun (..-ِ\n3. Dhammah diikuti Waw Sukun (..-ُ\n4. Fathah berdiri atau Kasrah berdiri (..-ْ\n5. Dhammah terbalik (..ْ\n6. Tanda Madd panjang (..~\n"
      }
    ]
  },
  ...
]

```

Struktur ini dirancang agar dapat menampilkan materi dalam bentuk yang sistematis dan mudah dipahami oleh pengguna. Penggunaan *content* sebagai *array* memungkinkan materi disusun menjadi bagian-bagian kecil agar tidak membingungkan, serta memudahkan pemrosesan tampilan secara dinamis di aplikasi.

Seluruh data materi disimpan dalam format JSON lokal, sehingga dapat diakses tanpa koneksi internet dan dengan pembagian struktur menjadi Materi *List* dan Materi Detail, sistem menjadi lebih modular dan terstruktur. Pendekatan ini memudahkan pengembang untuk menambahkan, mengedit, atau menghapus materi tertentu tanpa memengaruhi struktur data secara keseluruhan.

```

1 {
2   {
3     "id": 1,
4     "title": "Ahkamul Madd",
5     "description": "Hukum Bacaan Panjang",
6     "file": "materi_detail_1.json"
7   },
8   {
9     "id": 2,
10    "title": "Ahkamul huruf",
11    "description": "Hukum Bacaan Huruf",
12    "file": "materi_detail_2.json"
13  },
14 }

```

Gambar 3.22 Struktur data JSON Materi *List*

```

1 {
2   {
3     "id": 1,
4     "title": "Pengertian Madd",
5     "content": [
6       {
7         "subtitle": "Pengertian",
8         "body": "Madd ialah bacaan panjang dalam Al-Qur'an disebabkan adanya tanda madd. Tanda-tanda madd yaitu:"
9       },
10      {
11        "subtitle": "",
12        "body": "1. Fathah diikuti Alif (..-.)\n2. Kasrah diikuti Yaa Sukun (..-.)\n3. Dhammah diikuti Waw Sukun (..-.)\n4. Fathah berdiri atau Kasroh be
13      ]
14    },
15  }
16 }

```

Gambar 3.23 Struktur data JSON Materi Detail