

# **PERANCANGAN APLIKASI ANDROID BERBASIS KOTLIN DAN XML (MyQuran)**

*Hasil Laporan Ini Disusun Untuk Memenuhi Tugas 2 Mata Kuliah Sistem Informasi  
Manajemen*

**Dosen Pengampu:**

DR. Syopiansyah Jaya Putra, M.Sis



Universitas Islam Negeri  
**SYARIF HIDAYATULLAH JAKARTA**

**Disusun Oleh:**

Muhammad Thoriq Nabasa : 12409031050138

Muhammad Muhammad Faiqul Umam Dzunnuroeni : 12409031050122

Muhammad Aulia Addinul Haq : 12409031050131

**PROGRAM STUDI SITEM INFORMASI**

**FAKULTAS SSAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA**

**1445 H / 2025 M**

## DAFTAR ISI

# BAB 1

## PENDAHULUAN

### I. Latar Belakang

Indonesia merupakan negara dengan populasi Muslim terbesar di dunia, sehingga kebutuhan akan teknologi yang dapat menunjang aktivitas keagamaan menjadi sangat signifikan. Seiring berkembangnya era digital, berbagai aplikasi keislaman mulai bermunculan, menawarkan fitur-fitur seperti pengingat waktu salat, penunjuk arah kiblat, hingga pembacaan Al-Qur'an secara digital. Meskipun demikian, berdasarkan observasi awal yang dilakukan oleh tim kami, sebagian besar aplikasi tersebut masih terbatas pada fungsi-fungsi dasar dan belum sepenuhnya mengakomodasi kebutuhan interaktif dan sosial pengguna Muslim.

Berdasarkan permasalahan tersebut, kelompok kami merancang sebuah aplikasi Android bernama **MyQuran** yang dikembangkan menggunakan bahasa pemrograman **Kotlin** dan bahasa markup **XML** untuk antarmuka pengguna. MyQuran tidak hanya menyediakan fitur-fitur dasar seperti waktu salat, arah kiblat, dan pembacaan Al-Qur'an, tetapi juga menghadirkan fitur unggulan yaitu "**Community**", yang memungkinkan pengguna untuk berdiskusi, berbagi pengetahuan, serta membentuk jejaring sosial Islami secara online.

Pemilihan **Kotlin** sebagai bahasa pengembangan utama didasarkan pada interoperabilitasnya yang tinggi dengan Java, sintaks yang lebih ringkas dan aman, serta dukungan resmi dari Google sebagai bahasa utama dalam pengembangan aplikasi Android modern (Joorabchi et al., 2019; Sharma & Chauhan, 2022). Sementara itu, **XML** digunakan untuk mendefinisikan struktur antarmuka pengguna (UI), memberikan fleksibilitas dalam mendesain tampilan yang intuitif dan responsif (Sarker et al., 2021).

Proyek ini dirancang dengan pendekatan **project-based learning** yang menekankan pada pembelajaran melalui implementasi langsung dalam bentuk proyek nyata. Pendekatan ini terbukti meningkatkan keterampilan teknis sekaligus mendorong pemahaman konseptual yang lebih mendalam (Guzdial, 2019; Helle et al., 2006). Laporan ini turut dilengkapi dengan studi kasus dan cuplikan kode untuk memfasilitasi pemahaman teknis pengembangan aplikasi, serta mengintegrasikan prinsip-prinsip **user-centered design** dalam proses perancangannya agar sesuai dengan kebutuhan nyata pengguna Muslim di Indonesia.

### II. Rumusan Masalah

1. Bagaimana merancang aplikasi MyQuran yang interaktif, responsif, dan mudah digunakan pada platform Android dengan menggunakan bahasa pemrograman Kotlin?
2. Apa saja teknologi dan pustaka pendukung yang tepat untuk mengembangkan fitur-fitur penting seperti pembacaan ayat, terjemahan, audio murottal, dan bookmark ayat dengan menggunakan Kotlin di Android Studio?
3. Bagaimana merancang antarmuka pengguna (UI) yang sederhana, menarik, dan memudahkan pengguna dalam menavigasi konten Al-Qur'an secara efisien?

### **III. Tujuan Proyek**

1. Mengembangkan aplikasi Al-Qur'an digital bernama MyQuran yang dapat digunakan pada perangkat Android
2. Mengimplementasikan fitur-fitur utama seperti tampilan ayat dan terjemahan, pencarian surat atau ayat, audio murottal, serta fitur bookmark untuk menandai ayat pilihan.
3. Merancang antarmuka pengguna (UI) yang responsif, sederhana, dan sesuai dengan prinsip kenyamanan pengguna (user-friendly).

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

## BAB 3

### METODOLOGI

#### 3.1 Pendekatan Pengembangan

Perancangan Aplikasi Berbasis Kotlin Dan XML (MyQuran) dilakukan dengan pendekatan **Analisis dan Perancangan Sistem Informasi (APSI)** yang terstruktur. Pendekatan ini dipadukan dengan metode **User-Centered Design (UCD)** guna memastikan bahwa aplikasi benar-benar menjawab kebutuhan pengguna utama, yaitu umat Muslim dari berbagai latar belakang usia dan tingkat literasi teknologi. Proses pengembangan dilakukan secara iteratif dengan siklus analisis, desain, implementasi, pengujian, dan evaluasi berkelanjutan. Pendekatan ini tidak hanya menitikberatkan pada aspek teknis, tetapi juga pada kenyamanan, kegunaan, dan pengalaman pengguna secara menyeluruh.

##### A. Analisa Kebutuhan

Analisis kebutuhan dimulai dengan observasi langsung terhadap target pengguna untuk mengidentifikasi kebutuhan utama mereka. Hasil analisis menunjukkan bahwa pengguna membutuhkan fitur-fitur seperti pembacaan Al-Qur'an lengkap dengan transliterasi dan terjemahan, pengingat waktu salat berbasis lokasi, Lokasi masjid terdekat, Community sesama muslim serta penunjuk arah kiblat menggunakan kompas digital. Untuk memperdalam pemahaman tentang konteks pengguna, dikembangkan pula **User Interface** dan **journey map** guna memvisualisasikan perilaku dan alur interaksi pengguna saat menggunakan aplikasi

##### B. Perancangan Sistem

Tahap perancangan sistem mencakup aspek arsitektur aplikasi, integrasi API, serta struktur data. Aplikasi yang dirancang mengadopsi arsitektur berbasis Activity (Activity-based architecture) yang modular dan terpisah antar fitur, sehingga memudahkan proses pengembangan dan pemeliharaan.

Struktur Activity meliputi:

- **MainActivity**: Tampilan splash screen atau halaman pembuka.
- **MenuActivity**: Navigasi utama ke berbagai fitur.
- **ReadQuranActivity**: Menampilkan daftar surah dan ayat Al-Qur'an.
- **AdhanActivity**: Menyediakan jadwal salat yang disesuaikan dengan lokasi pengguna.
- **QiblaActivity**: Menampilkan arah kiblat dengan antarmuka kompas digital
- **SettingActivity**: Navigasi untuk mengatur aplikasi

Aplikasi ini mengintegrasikan dua API utama:

- <https://api.alquran.cloud> untuk teks, transliterasi, dan audio ayat-ayat Al-Qur'an.
- <https://api.myquran.com> untuk data lokasi dan waktu salat secara real-time

### C. Desain Antarmuka UI/UX

Desain antarmuka mengikuti prinsip UI/UX modern yang berfokus pada kesederhanaan, konsistensi, dan aksesibilitas. Antarmuka pengguna dirancang menggunakan **XML** dengan komponen **ConstraintLayout** dan **CardView** yang memungkinkan pemosisian elemen secara fleksibel dan responsif di berbagai ukuran layar. Ukuran font disesuaikan agar tetap terbaca jelas di perangkat berlayar kecil maupun besar, serta disiapkan untuk mendukung aksesibilitas visual.

### D. Implementasi Dengan Kotlin

Pengembangan aplikasi dilakukan dengan Kotlin sebagai bahasa utama karena dukungan resminya dari Google, serta kemampuannya dalam menulis kode yang ringkas dan bebas dari banyak bug umum seperti `NullPointerException`

Beberapa implementasi teknis penting antara lain:

- **Navigasi antar Activity** menggunakan *Intent* untuk berpindah antar fitur
- **Pengelolaan data** dilakukan melalui pemrosesan **JSON** dari API menggunakan library `OkHttp`, dan untuk penyimpanan lokal menggunakan `SharedPreferences`
- **Fitur interaktif** mencakup:
  - Audio murottal menggunakan **MediaPlayer**.
  - Arah kiblat ditampilkan dengan dukungan **SensorManager**, memanfaatkan sensor magnetometer dan akselerometer pada perangkat.

## 3.2 Teknologi Dan Tools

Dalam Perancangan Aplikasi Berbasis Kotlin Dan XML (MyQuran), pemilihan teknologi dan tools didasarkan pada kebutuhan untuk menciptakan aplikasi Android yang modern, responsif, mudah diintegrasikan, serta mampu memberikan pengalaman pengguna yang optimal. Berikut adalah rincian teknologi dan perangkat yang digunakan selama proses pengembangan.

### A. Bahasa Pemrograman Kotlin

Kotlin menawarkan interoperabilitas penuh dengan Java, sehingga memungkinkan integrasi dengan pustaka dan kode Java yang sudah ada tanpa kendala kompatibilitas. Selain itu, sintaks Kotlin lebih ringkas dan ekspresif dibandingkan Java, sehingga mempercepat proses pengembangan dan mengurangi risiko bug, khususnya *null pointer exception*. Kotlin juga mendukung fitur-fitur modern seperti:

- **Coroutines** untuk pemrograman asynchronous.
- **Extension functions** untuk menambahkan fungsi baru ke kelas lama.
- **Smart casting** dan *type inference* yang menyederhanakan struktur kode.

### B. Desain Antarmuka: XML

Untuk perancangan tampilan (User Interface), digunakan **XML** sebagai format utama. XML memungkinkan pengaturan struktur antarmuka pengguna secara terstruktur dan fleksibel. Tata letak dirancang menggunakan komponen **ConstraintLayout**, **CardView**, dan **RecyclerView**. Komponen *RecyclerView* secara khusus digunakan dalam fitur pembacaan Al-Qur'an untuk menampilkan daftar surah

dan ayat secara dinamis dan efisien. Desain UI juga disesuaikan dengan prinsip aksesibilitas dan responsivitas agar tetap optimal di berbagai ukuran dan resolusi layar.

### C. API Eksternal

Integrasi data dilakukan dengan memanfaatkan dua API eksternal yang kredibel dan terpercaya:

- API Al-Qur'an: <https://api.alquran.cloud/v1/surah> digunakan untuk mengambil daftar surah dan ayat, lengkap dengan transliterasi, terjemahan, dan audio murottal.
- API Jadwal Sholat: <https://api.myquran.com/v2/sholat> digunakan untuk mendapatkan data waktu salat harian berdasarkan lokasi pengguna, serta daftar kota di Indonesia.

Data dari API ini diproses dalam format JSON menggunakan *JSON parser* Kotlin (seperti `JSONObject` dan `JSONArray`), dan sebagian data disimpan secara lokal untuk mendukung mode offline.

### D. Tools Pengembangan: Android Studio

Pengembangan aplikasi dilakukan sepenuhnya melalui Android Studio, Integrated Development Environment (IDE) resmi dari Google untuk Android. Android Studio menyediakan berbagai fitur canggih seperti:

- **Real-time debugging.**
- **Layout Editor** untuk desain antarmuka secara visual.
- **Emulator** untuk menguji aplikasi di berbagai versi Android dan ukuran perangkat.
- **Profiling tools** untuk memantau performa aplikasi selama runtime

Dengan Android Studio, dapat menguji dan mengoptimalkan performa aplikasi dalam berbagai skenario, mulai dari perangkat berperforma tinggi hingga perangkat kelas menengah dengan keterbatasan memori.



## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil Pengembangan Perancangan Sistem Aplikasi dengan Kotlin dan XML

Di sini ini kelompok kami menyajikan hasil dari perancangan dan pengembangan aplikasi "MyQuran" serta pembahasan mendalam mengenai setiap aspeknya. Hasil yang disajikan mencakup wujud antarmuka pengguna (UI) dan pengalaman pengguna (UX) dari prototipe yang telah dibuat, serta detail implementasi teknisnya. Pembahasan akan mengupas relevansi hasil pengembangan dengan teori dan prinsip desain yang telah ditetapkan, arsitektur sistem yang dipilih, serta tantangan dan solusi yang ditemukan selama proses pengembangan.

##### 4.1.1 Arsitektur dan Fungsionalitas Aplikasi

Aplikasi MyQuran berhasil diimplementasikan dengan arsitektur modular berbasis Activity, di mana setiap fitur utama dikelola oleh komponen yang terpisah untuk memastikan kode yang bersih dan mudah dikelola. Alur navigasi antar-fitur diimplementasikan menggunakan Intent untuk perpindahan antar layar.

Fungsionalitas utama yang berhasil diimplementasikan adalah sebagai berikut:

1. **Read Quran:** Pengguna dapat menelusuri daftar 114 surah, melihat ayat per ayat dalam teks Arab, lengkap dengan transliterasi dan terjemahan. Fitur ini juga dilengkapi dengan pemutaran audio (audio playback) untuk setiap ayat.
2. **Adhan Schedule:** Aplikasi dapat menampilkan jadwal sholat harian (Subuh, Dzuhur, Ashar, Maghrib, Isya) yang akurat berdasarkan lokasi yang dipilih pengguna. Data diambil dari API eksternal dan dapat diperbarui.
3. **Qibla Direction:** Fitur ini menyediakan kompas digital yang menunjukkan arah kiblat secara *real-time* dengan memanfaatkan sensor accelerometer dan magnetometer pada perangkat pengguna.
4. **Last Read:** Aplikasi secara otomatis menyimpan progres bacaan terakhir pengguna (surah dan ayat) menggunakan SharedPreferences, sehingga pengguna dapat melanjutkannya dengan mudah saat membuka kembali aplikasi.
5. **Pencarian:** Fitur pencarian pada halaman daftar surah memungkinkan pengguna untuk memfilter dan menemukan surah yang diinginkan dengan cepat menggunakan EditText dan TextWatcher.

##### 4.1.2 Hasil Desain Antarmuka Pengguna (UI/UX)

Hasil perancangan antarmuka aplikasi "MyQuran" diwujudkan dalam bentuk prototipe *high-fidelity* yang dikembangkan menggunakan perangkat lunak Figma. Prototipe ini mencerminkan penerapan prinsip-prinsip UI/UX yang berfokus pada kesederhanaan,

konsistensi, dan aksesibilitas. Berikut adalah rincian hasil visual dari setiap fitur utama: Hasil Perancangan Antarmuka (UI/UX)

- [Gambar Halaman Sambutan Aplikasi MyQuran]
- **Halaman Menu Utama (menu\_activity.xml):** Berfungsi sebagai pusat navigasi. Tombol-tombol fitur utama disajikan menggunakan CardView untuk memberikan efek bayangan (elevation) dan tampilan modern. Tombol disusun secara vertikal di tengah layar untuk kemudahan jangkauan. Latar belakang menggunakan drawable gradien hijau yang menenangkan untuk menciptakan suasana yang sesuai. [Gambar Halaman Menu Utama Aplikasi MyQuran]
- **Halaman Baca Quran (read\_quran\_activity.xml):** Dirancang menggunakan ConstraintLayout untuk performa dan fleksibilitas. Halaman ini berisi kolom pencarian, kartu "Last Read", dan daftar surah yang ditampilkan dalam sebuah RecyclerView untuk memastikan efisiensi memori dan kelancaran saat menggulir daftar yang panjang. [Gambar Halaman Daftar Surah Aplikasi MyQuran]
- **Halaman Detail Ayat (ayat\_activity.xml):** Menampilkan detail ayat dengan jelas, memisahkan teks Arab, transliterasi, dan terjemahan dalam tata letak yang mudah dibaca. Terdapat tombol kontrol untuk audio (play/pause). [Gambar Halaman Detail Ayat dan Audio]
- **Halaman Jadwal Adzan (adhan\_schedule\_activity.xml):** Menggunakan LinearLayout vertikal untuk menyajikan informasi tanggal (Hijriyah dan Masehi) dan daftar waktu sholat secara terstruktur dan berurutan dari atas ke bawah. [Gambar Halaman Jadwal Adzan]
- **Halaman Arah Kiblat (compas\_main.xml):** Menggunakan ConstraintLayout untuk memastikan ImageView kompas dan informasi derajat tetap berada di tengah layar secara presisi pada berbagai ukuran perangkat. [Gambar Halaman Penunjuk Arah Kiblat]
- **Halaman Settings :** Fitur Settings memungkinkan pengguna untuk melakukan personalisasi aplikasi sesuai preferensi. Pengguna dapat mengubah bahasa terjemahan ayat (antara Indonesia, Inggris, dan Melayu), mengatur ukuran font terjemahan, serta mengaktifkan atau menonaktifkan tampilan transliterasi. Semua perubahan pengaturan ini disimpan menggunakan SharedPreferences, sehingga preferensi pengguna tetap tersimpan bahkan setelah aplikasi ditutup atau perangkat di-restart.

## 4.2 Pembahasan

Bagian ini menganalisis secara mendalam hasil implementasi aplikasi MyQuran dengan mengacu pada tujuan awal, metodologi, serta tantangan yang dihadapi selama proses pengembangan.

### 4.2.1 Kesesuaian Hasil dengan Analisis Kebutuhan

Produk aplikasi yang dikembangkan telah berhasil memenuhi kebutuhan fungsional dan non-fungsional yang diidentifikasi pada tahap analisis sistem, sejalan dengan pendekatan Analisa dan Perancangan Sistem Informasi (APSI).

- **Pemenuhan Kebutuhan Fungsional:** Seluruh fitur utama yang direncanakan—membaca Al-Quran, jadwal sholat, arah kiblat, dan penanda bacaan terakhir—telah terimplementasi dengan baik. Fitur-fitur ini secara langsung menjawab kebutuhan utama target pengguna, yaitu umat Muslim yang memerlukan alat bantu digital untuk aktivitas ibadah harian.
- **Pemenuhan Kebutuhan Non-Fungsional:**
  - **Performa:** Penggunaan RecyclerView untuk menampilkan daftar panjang (seperti daftar surah) dan ConstraintLayout untuk mengurangi kedalaman hierarki layout (nested layouts) berhasil menjaga aplikasi tetap responsif dan memiliki waktu muat yang cepat.
  - **Keandalan (Mode Offline):** Dengan menyimpan data dari API (seperti daftar kota dan surah) ke dalam penyimpanan internal setelah pengunduhan pertama, aplikasi dapat diandalkan bahkan saat koneksi internet tidak stabil, yang merupakan salah satu kebutuhan krusial bagi pengguna di Indonesia.
  - **Kegunaan (Usability):** Desain yang sederhana, navigasi yang intuitif, dan penggunaan ikon yang universal memastikan aplikasi mudah digunakan oleh berbagai segmen pengguna, sesuai dengan prinsip *User-Centered Design* (UCD). Ukuran font minimal 16sp dan kontras warna yang baik (misalnya, teks putih di atas latar hijau tua) juga mendukung aspek aksesibilitas bagi pengguna dengan keterbatasan visual.

#### 4.2.2 Analisis Pendekatan Desain dan Implementasi

Pemilihan teknologi dan pendekatan pengembangan memainkan peran krusial dalam keberhasilan proyek ini.

- **Kotlin dan XML:** Penggunaan Kotlin sebagai bahasa pemrograman utama terbukti sangat efektif. Sintaksisnya yang ringkas dan modern mempercepat proses pengembangan, sementara fitur keamanannya seperti *null safety* secara signifikan mengurangi risiko *crash* akibat NullPointerException, yang merupakan salah satu *bug* paling umum di platform Java. Di sisi lain, penggunaan XML untuk mendefinisikan UI menegaskan prinsip *separation of concerns*, yang memisahkan logika aplikasi (Kotlin) dari representasi tampilan (XML). Hal ini membuat struktur proyek lebih rapi, mudah dipelihara, dan memungkinkan kolaborasi yang efisien antara desainer UI dan pengembang.
- **Desain UI/UX:** Pendekatan desain yang berfokus pada kesederhanaan visual (minimalis) dan konsistensi (penggunaan skema warna, tipografi, dan tata letak yang seragam) berhasil menciptakan pengalaman pengguna yang kohesif dan tidak membingungkan. Penggunaan CardView tidak hanya berfungsi sebagai elemen estetika, tetapi juga sebagai penanda visual yang jelas untuk elemen-elemen interaktif, menciptakan hierarki visual yang membantu pengguna fokus pada tindakan utama.
- **Integrasi API:** Pemanfaatan API eksternal yang kredibel (api.alquran.cloud dan api.myquran.com) memungkinkan aplikasi untuk menyajikan konten yang dinamis dan akurat tanpa perlu membangun dan mengelola database konten yang kompleks dari awal. Hal ini membuat proses pengembangan menjadi jauh lebih efisien dan fokus pada pengalaman pengguna.

### 4.2.3 Pembahasan Tantangan Teknis dan Solusinya

Selama proses pengembangan, beberapa tantangan teknis ditemukan dan solusi efektif berhasil diterapkan untuk mengatasinya.

- **Tantangan 1:** Ketergantungan pada Koneksi Internet  
Akses data yang sepenuhnya bergantung pada API secara real-time dapat menyebabkan aplikasi tidak berfungsi atau lambat di area dengan koneksi internet yang buruk, yang dapat menurunkan tingkat keandalan aplikasi.  
**Solusi:** Menerapkan mekanisme penyimpanan data secara lokal atau *caching*. Data yang jarang berubah, seperti daftar surah dan daftar kota, diambil dari API sekali saat pertama kali diakses, lalu disimpan dalam format file JSON di penyimpanan internal perangkat. Untuk penggunaan selanjutnya, aplikasi akan memuat data dari penyimpanan lokal terlebih dahulu, sehingga fitur-fitur tersebut tetap dapat diakses secara instan dan *offline*.
- **Tantangan 2:** Gerakan Kompas Arah Kiblat yang Tidak Mulus  
Pembacaan data langsung dari sensor magnetometer dan accelerometer perangkat menghasilkan fluktuasi nilai yang tinggi, yang membuat animasi rotasi kompas terlihat patah-patah (jumpy atau jittery). Hal ini mengurangi kenyamanan dan kepercayaan pengguna terhadap fitur penunjuk arah kiblat.  
**Solusi:** Untuk menghaluskan gerakan, *RotateAnimation* diterapkan pada *ImageView* yang berfungsi sebagai kompas. Animasi ini membuat transisi visual dari satu sudut ke sudut berikutnya berjalan lebih lembut dan alami. Sebagai penyempurnaan lebih lanjut, sebuah filter *low-pass* dapat diaplikasikan pada data mentah dari sensor untuk mengurangi "noise" sebelum data tersebut digunakan untuk memperbarui rotasi kompas. Solusi ini secara signifikan meningkatkan kualitas pengalaman pengguna saat menggunakan fitur tersebut.
- **Tantangan 3:** Fitur "Last Read" yang hanya mampu menyimpan satu posisi terakhir tidak cukup untuk mendukung kebutuhan pengguna yang ingin mempelajari atau menghafal Al-Qur'an secara mendalam. Pengguna sering kali perlu menandai banyak ayat penting, ayat yang sedang dihafal, atau ayat yang berkaitan dengan tema tertentu untuk referensi di kemudian hari. Keterbatasan ini mengurangi utilitas aplikasi sebagai alat bantu studi.  
**Solusi:** Mengembangkan **sistem bookmark multi-fungsi**. Secara teknis, ini dapat dicapai dengan membuat tabel baru (bookmarks) dalam database SQLite lokal. Tabel ini akan menyimpan informasi seperti *surah\_id*, *ayat\_id*, catatan pengguna (opsional untuk memberi label atau tag), dan timestamp. Pada antarmuka *AyatActivity*, sebuah ikon *bookmark* akan ditambahkan di setiap baris ayat. Selain itu, perlu dibuat halaman Daftar Bookmark khusus yang menampilkan semua ayat yang telah ditandai, di mana pengguna dapat mengelola (menghapus, mengedit catatan) dan melompat langsung ke ayat-ayat tersebut.

### 4.3 Uji Coba Sistem

Pengujian aplikasi dilakukan pada beberapa perangkat Android dengan versi sistem operasi yang berbeda. Hasilnya menunjukkan bahwa seluruh fitur utama dapat berjalan dengan baik. Proses pengambilan dan penampilan data dari API, navigasi antar-halaman, fungsi pencarian, penyimpanan pengaturan, dan fitur berbasis sensor berjalan sesuai dengan yang diharapkan tanpa ditemukan *bug* mayor yang menghalangi fungsionalitas.

**Kesimpulan Uji Coba:** Berdasarkan hasil implementasi dan pengujian, aplikasi MyQuran telah berhasil memenuhi seluruh kebutuhan fungsional yang telah dirancang. Sistem ini stabil, responsif, dan memberikan pengalaman pengguna yang baik. Meskipun demikian, pengembangan lebih lanjut masih dapat dilakukan untuk mengatasi keterbatasan yang ada, terutama penambahan fitur *offline* dan sistem *bookmark* yang lebih canggih, agar aplikasi ini menjadi semakin lengkap dan bermanfaat bagi pengguna.

## **BAB 5 KESIMPULAN**

## **DAFTAR PUSTAKA**