

Perancangan Sistem Aplikasi Al-Qur'an Mushaf Gontor Juz 1 Berbasis Mobile

M. Ainul Qolbi Al-Hebroni¹, Dihin Muriyatmoko², Triana Harmini³

¹²³Teknik Informatika, ¹²³Universitas Darussalam Gontor

¹²³Ponorogo, ¹²³Jawa Timur - Indonesia

Email: ¹alhebroni14@mhs.unida.gontor.ac.id, ²dihin@unida.gontor.ac.id,
³triana@unida.gontor.ac.id

Abstract

Pondok Modern Darussalam Gontor possesses a valuable spiritual asset in the form of the Gontor Mushaf, written in a distinctive Naskhi script style by Al-Ustadz M. Nur, Lc. However, the current availability of this Mushaf solely in printed format poses preservation challenges due to physical damage risks and limited accessibility for students and the broader community. This study aims to design and build a mobile application system for the Gontor Mushaf (Juz 1) based on Android that operates fully independently (offline). The system development utilizes the Waterfall model, encompassing requirements analysis, system design, implementation, testing, and operation & maintenance phases. The system is built using the Kotlin programming language with an Embedded Assets architecture, where vector-based PDF files and JSON text databases are embedded directly into the application. The results show that the 228 MB application successfully renders sharp Mushaf visuals without pixelation when zoomed, and the verse search feature operates instantly without an internet connection. Black Box testing across various device specifications confirms system stability. The application is prepared for distribution via the Google Play Store to ensure broad public access.

Keywords: Al-Qur'an, Android, Mobile Application, Mushaf Gontor, Waterfall.

Abstraksi

Pondok Modern Darussalam Gontor memiliki aset spiritual bernilai tinggi berupa Mushaf Gontor yang ditulis dengan gaya Khat Naskhi khas karya Al-Ustadz M. Nur, Lc. Namun, keberadaan mushaf ini yang hanya tersedia dalam format cetak menimbulkan kendala preservasi akibat risiko kerusakan fisik serta keterbatasan aksesibilitas bagi santri dan masyarakat luas. Penelitian ini bertujuan untuk merancang bangun sistem aplikasi mobile Mushaf Gontor Juz 1 berbasis Android yang mampu beroperasi sepenuhnya secara mandiri (offline). Metode pengembangan sistem menggunakan model Waterfall yang mencakup tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta operasi dan pemeliharaan. Sistem dibangun menggunakan bahasa pemrograman Kotlin dengan arsitektur Embedded Assets, di mana file PDF berbasis vektor dan basis data teks JSON ditanamkan langsung ke dalam aplikasi. Hasil penelitian menunjukkan bahwa aplikasi berukuran 228 MB ini mampu merender tampilan mushaf dengan kualitas tajam tanpa pecah saat diperbesar (zoom) dan fitur pencarian ayat berjalan instan tanpa memerlukan koneksi internet. Pengujian Black Box pada berbagai spesifikasi perangkat mengonfirmasi bahwa sistem berjalan stabil. Aplikasi ini dipersiapkan untuk didistribusikan melalui Google Play Store guna menjamin kemudahan akses publik.

Kata Kunci: Al-Qur'an, Android, Aplikasi Mobile, Mushaf Gontor, Waterfall.

1. PENDAHULUAN

Perkembangan teknologi telah mengubah perilaku umat Islam dalam membaca Al-Qur'an, dari media cetak ke media digital berbasis smartphone. Aplikasi Al-Qur'an mobile kini menawarkan akses yang fleksibel, memungkinkan pengguna membaca Al-Qur'an kapan dan di mana saja tanpa membawa mushaf fisik[1]. Di era digital saat ini, aplikasi *mobile* telah menjadi medium utama untuk mengakses berbagai layanan, mulai dari edukasi hingga pemenuhan kebutuhan ibadah. Aplikasi Al-Qur'an digital, secara khusus, telah berkembang pesat dan menjadi salah satu kategori aplikasi paling populer, yang memudahkan umat Muslim untuk membaca, menghafal, dan memahami kitab suci mereka kapan saja dan di mana saja[2]. Al-Qur'an, sebagai kitab suci dan petunjuk hidup utama bagi umat Islam, wajib untuk dibaca, dipahami, dan diamalkan[3].

Dalam ekosistem pendidikan Islam di Indonesia, Pondok Modern Darussalam Gontor memiliki aset spiritual dan kultural yang bernilai tinggi berupa Mushaf Gontor. Mushaf ini digagas secara khusus sebagai monumen intelektual untuk menyambut satu abad usia pondok, yang merepresentasikan identitas dan filosofi lembaga. Keunikan Mushaf Gontor terletak pada karakteristik visualnya, yang menampilkan gaya Khat Naskhi hasil tulisan tangan Al-Ustadz M. Nur, Lc., serta ornamen iluminasi bingkai yang dirancang dengan estetika khas Gontor. Selain nilai artistik, mushaf ini mempertahankan standar "Mushaf Pojok" (15 baris per halaman) yang sangat membantu dalam metode penghafalan[4]. Namun, keberadaan mushaf ini saat ini masih terbatas pada format fisik (cetak). Keterbatasan ini memunculkan kendala signifikan dari sisi preservasi dan distribusi. Format cetak memiliki risiko kerusakan fisik akibat faktor lingkungan dan degradasi usia, serta menyulitkan akses bagi ribuan alumni dan santri yang tersebar di berbagai penjuru dunia. Lebih jauh lagi, format statis ini tidak mendukung fitur pencarian cepat (*searchability*), yang menjadi hambatan dalam efisiensi studi dan penelusuran referensi ayat bagi para pengkaji Al-Qur'an[5]. Hingga saat ini, Mushaf Gontor yang bernilai historis dan memiliki kekhasan tersebut baru tersedia dalam bentuk media cetak (*hard copy*). Keterbatasan ini memunculkan permasalahan utama: media cetak rentan terhadap kerusakan fisik atau kehilangan, serta kurang praktis untuk diakses oleh santri, alumni, dan masyarakat luas yang hidup di era modern dengan mobilitas tinggi [5]. Belum adanya versi digital resmi dari Mushaf Gontor menciptakan sebuah kesenjangan (*gap*) antara ketersediaan warisan intelektual pondok dengan kebutuhan akses digital yang cepat dan efisien.

Guna mengatasi kesenjangan aksesibilitas dan risiko preservasi tersebut, penelitian ini mengajukan pendekatan pengembangan Aplikasi Native Android sebagai solusi digitalisasi[6]. Pemilihan platform Android didasari oleh kebutuhan krusial akan kemampuan rendering dokumen digital berkualitas tinggi secara offline[7]. Berbeda dengan solusi berbasis web yang sering mengalami kendala latency saat memuat dokumen besar, aplikasi native memungkinkan implementasi teknologi rendering PDF

berbasis vektor (vector-based PDF rendering). Pendekatan ini dipilih untuk menjaga integritas estetika Khat Mushaf Gontor; format vektor memungkinkan pengguna memperbesar tampilan (zoom) tanpa mengalami penurunan kualitas visual atau pecah (pixelated), sebuah keunggulan yang sulit dicapai oleh format gambar raster biasa[7]. Selain itu, manajemen memori pada lingkungan Android mendukung navigasi perpindahan halaman PDF yang responsif, memberikan pengalaman membaca (User Experience) yang mendekati kenyamanan membaca mushaf fisik[6].

Secara teknis, perancangan sistem ini menerapkan strategi hibrida (hybrid strategy) dalam pengelolaan data. Untuk tampilan visual utama (Tampilan Mushaf), sistem memanfaatkan file PDF scan vector yang disimpan secara lokal (local assets), menjamin tampilan yang otentik sesuai versi cetak[8]. Sementara itu, untuk mendukung fungsionalitas Tampilan Per-Ayat dan fitur pencarian, sistem mengintegrasikan basis data teks Al-Qur'an (JSON) yang ditanam di dalam aplikasi. Data teks ini dipetakan secara logis terhadap halaman PDF, sehingga memungkinkan fitur pencarian bekerja secara presisi: pengguna mencari kata kunci pada database lokal, dan sistem akan mengarahkan tampilan langsung ke halaman PDF yang relevan. Berdasarkan arsitektur ini, penelitian bertujuan merancang bangun sistem aplikasi mobile Mushaf Gontor Juz 1 yang mandiri (standalone), guna melestarikan keaslian visual mushaf dalam kualitas terbaik sekaligus menghadirkan fungsionalitas digital yang praktis bagi komunitas pondok dan masyarakat luas.

2. TINJAUAN PUSTAKA

Pengembangan aplikasi Al-Qur'an digital menempati posisi strategis di persimpangan antara kemajuan teknologi *mobile* dan studi keislaman. Landasan teknologi penelitian ini bertumpu pada konsep aplikasi *mobile*, yakni perangkat lunak yang dirancang khusus untuk beroperasi pada perangkat portabel seperti *smartphone* dan tablet, yang menawarkan portabilitas tinggi bagi pengguna untuk mengakses layanan spiritual kapan saja dan di mana saja[9]. Dalam implementasinya, penelitian ini memilih platform Android berbasis sistem operasi Linux karena sifatnya yang terbuka (*open source*) dan fleksibilitas komponen utamanya, seperti *Activity* untuk antarmuka pengguna dan *Service* untuk proses latar belakang, yang sangat mendukung pengembangan aplikasi berkinerja tinggi[10]. Untuk memastikan proses pengembangan yang terstruktur dan sistematis, penelitian ini mengadopsi metode *Waterfall* yang telah terbukti efektivitasnya dalam meminimalkan kesalahan logika alur sistem pada pengembangan aplikasi Android[9].

Dalam memetakan posisi penelitian (*state of the art*), tinjauan terhadap studi rancang bangun terdahulu menjadi krusial untuk menonjolkan kebaruan penelitian. Ridhoni, Sholeh, dan Basuki (2023) mengembangkan aplikasi Al-Qur'an *online* dengan memanfaatkan REST API[11]. Meskipun berhasil menyajikan data ayat yang lengkap, pendekatan ini memiliki ketergantungan mutlak pada koneksi internet dan antarmuka yang disajikan berbasis teks digital standar, sehingga tidak menampilkan karakteristik

visual mushaf tertentu. Di sisi lain, Kasoni, Liesnaningsih, dan Afif (2024) melakukan perancangan sistem pembelajaran Al-Qur'an berbasis Android menggunakan metode *Extreme Programming*[12]. Fokus penelitian tersebut adalah pada fitur interaktif untuk pembelajaran tajwid bagi pemula, namun belum menyentuh aspek preservasi manuskrip fisik ke dalam bentuk digital.

Penelitian rancang bangun lainnya oleh Pratiwi, Dhakiroh, dan Saefudin (2023) berfokus pada aplikasi hafalan Al-Qur'an untuk pemula[13]. Aplikasi tersebut menggunakan aset audio dan teks standar untuk membantu hafalan[13]. Berbeda dengan pendekatan-pendekatan tersebut, penelitian ini menawarkan kebaruan (*novelty*) melalui pengembangan sistem yang beroperasi secara mandiri (*offline*) dengan mengintegrasikan aset lokal, bukan API *online*. Selain itu, jika penelitian digitalisasi umumnya menggunakan citra *raster* (JPEG/PNG) yang rentan pecah saat diperbesar, penelitian ini menerapkan teknologi *rendering* PDF berbasis vektor untuk menjaga kehalusan tulisan tangan (khat). Dengan demikian, penelitian ini mengisi celah kosong dengan berfokus pada rancang bangun teknis digitalisasi artefak spesifik, yakni Mushaf Gontor, menggunakan pendekatan hibrida aset lokal yang belum banyak dibahas dalam penelitian sejenis.

Objek utama dalam penelitian ini adalah Mushaf Gontor. Secara etimologi, mushaf bermakna kumpulan lembaran tertulis wahyu Al-Qur'an yang dikodifikasikan sejak masa Khalifah Utsman bin Affan[14]. Mushaf Gontor memiliki karakteristik unik yang membedakannya dari Mushaf Standar Indonesia (MSI) pada umumnya, meskipun sama-sama menggunakan standar 15 baris (mushaf pojok). Kekhasan tersebut terletak pada gaya penulisan Khat Naskhi karya Al-Ustadz M. Nur, Lc., serta ornamen bingkai iluminasi yang mencerminkan identitas Pondok Modern Darussalam Gontor. Preservasi karakteristik visual inilah yang menjadi fokus utama dalam perancangan sistem aplikasi ini.

3. METODE PENELITIAN

Metode Pengembangan Sistem Penelitian ini menerapkan metode System Development Life Cycle (SDLC) dengan model Waterfall. Model ini dipilih karena karakteristik kebutuhan sistem—yaitu digitalisasi Mushaf Gontor Juz 1—bersifat fixed requirements (telah terdefinisi dengan jelas di awal) dan tidak menuntut perubahan dinamis di tengah proses pengembangan. Tahapan penelitian dilakukan secara sekuensial meliputi: Analisis Kebutuhan, Perancangan Sistem, Implementasi, dan Pengujian Sistem dan Operasi dan Pemeliharaan.

3.1. Analisis Kebutuhan (Requirements Definition)

Tahap ini bertujuan untuk mengidentifikasi spesifikasi teknis dan fungsional yang harus dipenuhi oleh sistem agar dapat berjalan sesuai tujuan penelitian.

1. Identifikasi Aktor: Sistem dirancang untuk digunakan oleh satu jenis aktor, yaitu Pengguna Umum (User). Aktor ini memiliki hak akses penuh terhadap seluruh fitur aplikasi tanpa perlu melalui proses autentikasi (login) ke server, guna mendukung prinsip aksesibilitas 100% offline.

2. Kebutuhan Fungsional:

- Fitur Visualisasi Mushaf: Sistem harus mampu merender file PDF berbasis vektor dari penyimpanan lokal (assets) untuk menampilkan halaman Mushaf Gontor dengan kualitas tinggi yang tidak pecah saat diperbesar (zoom).
- Fitur Baca Per-Ayat: Sistem harus mampu melakukan parsing data teks Al-Qur'an dari format JSON lokal dan menampilkannya dalam struktur daftar (list view) yang responsif.
- Fitur Pencarian Lokal: Sistem harus menyediakan mekanisme pencarian kata kunci yang memindai basis data teks lokal dan mengarahkan pengguna ke ayat atau halaman yang relevan secara instan.
- Fitur Penanda (Bookmark): Sistem harus dapat menyimpan status halaman terakhir yang dibaca pengguna ke dalam penyimpanan lokal perangkat (Shared Preferences atau Local Database).

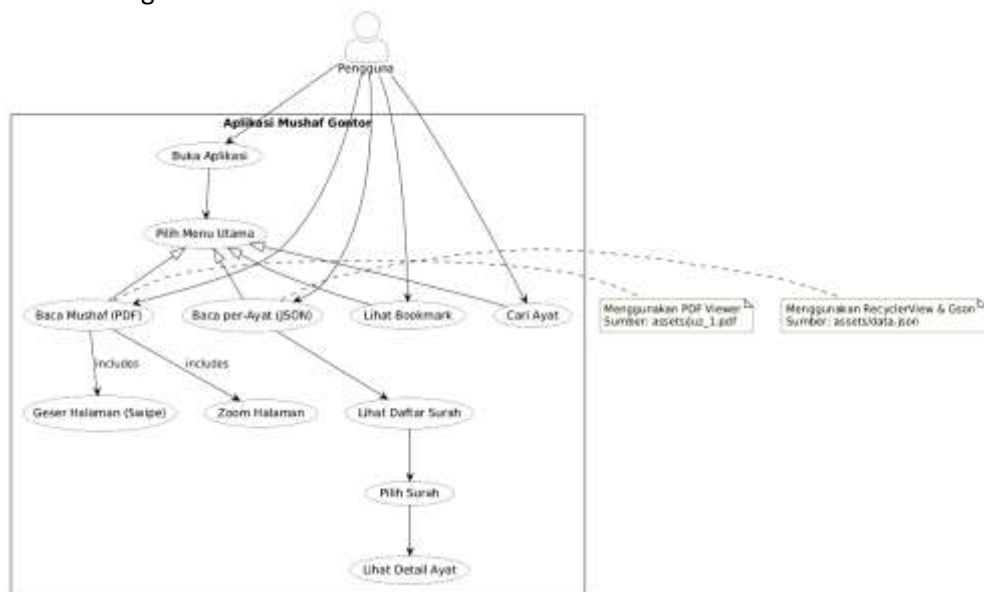
3. Kebutuhan Non-Fungsional:

- Aplikasi dapat berjalan pada perangkat bergerak dengan sistem operasi Android minimal versi 7.0 (Nougat).
- Aplikasi harus beroperasi sepenuhnya secara mandiri (standalone) tanpa membutuhkan koneksi internet atau API eksternal.
- Waktu respon perpindahan halaman mushaf tidak boleh melebihi 2 detik.

3.2. Perancangan Sistem (System and Software Design)

Pada tahap ini, hasil analisis kebutuhan diterjemahkan ke dalam rancangan arsitektur perangkat lunak menggunakan standar Unified Modeling Language (UML).

1. Use Case Diagram

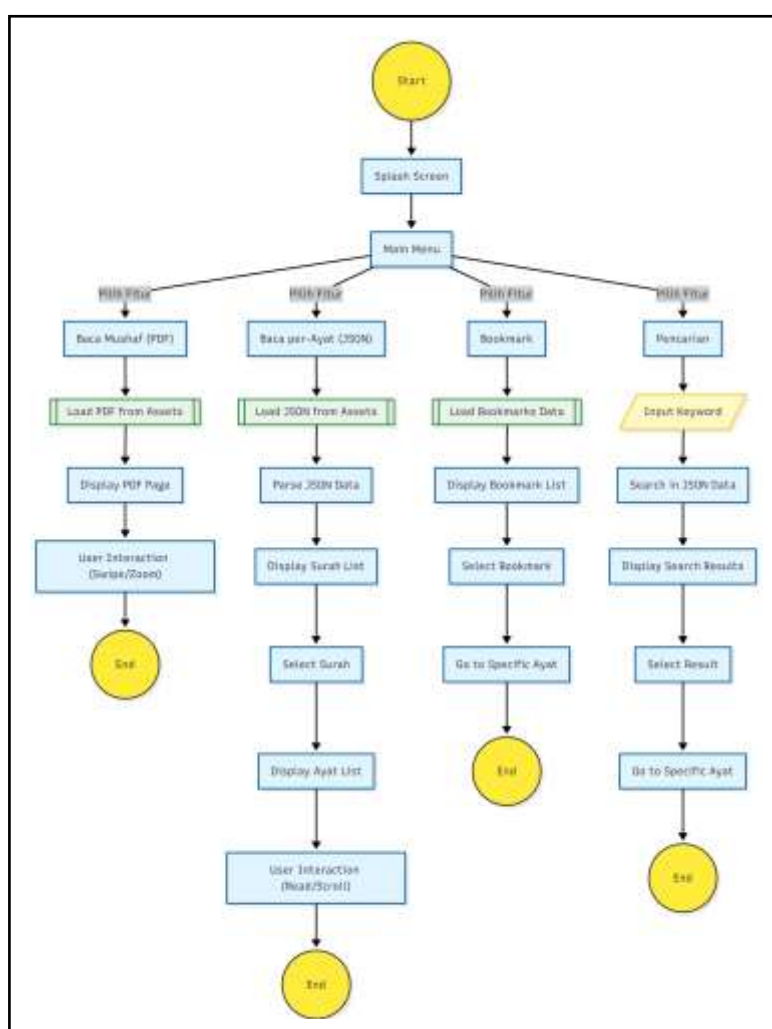


Gambar 1. Usecase Diagram

Use Case Diagram Use Case pada Gambar 1 memodelkan interaksi fungsional antara aktor Pengguna dengan sistem. Diagram ini menegaskan bahwa sistem dibangun berdasarkan fungsi utama, bukan sekadar struktur menu.

- a. Baca Mushaf (PDF): Merupakan fungsi inti untuk menampilkan digitalisasi fisik mushaf. Fungsi ini memiliki relasi include dengan fungsi Geser Halaman (Swipe) dan Zoom Halaman, yang menandakan bahwa fitur navigasi dan perbesaran adalah bagian tak terpisahkan dari pengalaman membaca mushaf.
- b. Baca Per-Ayat (JSON): Fungsi alternatif untuk membaca teks Arab beserta terjemahannya dalam format vertikal. Fungsi ini mencakup alur pemilihan Surat hingga tampilan detail ayat.
- c. Cari Ayat & Bookmark: Fungsi pendukung untuk mempercepat navigasi pengguna menuju konten spesifik.

2. Flowchart



Gambar 2. Flowchart Sistem Aplikasi

Perancangan Flowchart Sistem Alur logika sistem secara global digambarkan melalui Flowchart pada Gambar 2. Alur dimulai dari Splash Screen menuju Menu Utama, kemudian sistem membagi proses ke dalam empat percabangan logika utama:

- a. Logika Rendering PDF: Sistem memanggil Asset Manager untuk memuat file `juz_1.pdf` . Jika berhasil, halaman ditampilkan dengan fitur interaktif touch event (geser/cubit).
- b. Logika Parsing JSON: Sistem membaca file `data.json` , mengonversinya menjadi objek Java/Kotlin menggunakan pustaka Gson, lalu mendistribusikannya ke dalam adapter tampilan daftar (RecyclerView).
- c. Logika Pencarian: Sistem menerima input string dari pengguna, melakukan iterasi pencocokan pada data JSON lokal, dan menampilkan hasil. Klik pada hasil pencarian akan memicu navigasi (intent) ke posisi ayat terkait.
- d. Logika Bookmark: Sistem membaca/menulis data ID halaman ke memori internal perangkat.

3.3. Implementasi (Implementation and Unit Testing)

Tahap implementasi adalah proses penerjemahan desain ke dalam kode program. Aplikasi dibangun menggunakan lingkungan pengembangan Android Studio dengan bahasa pemrograman Kotlin. Arsitektur aplikasi menerapkan konsep Offline-First, di mana seluruh aset digital ditanamkan langsung (embedded) di dalam paket aplikasi (APK).

1. Manajemen Aset: File Mushaf disimpan dalam format PDF Vektor di direktori `assets/pdf/` , sedangkan data teks disimpan dalam format JSON di `assets/json/` .
2. Antarmuka Pengguna (UI): Disusun menggunakan XML Layouts yang menerapkan komponen Material Design untuk menjamin kenyamanan visual.
3. Logika Program: Menggunakan komponen PDFRenderer bawaan Android untuk visualisasi mushaf dan Kotlin Coroutines untuk menangani proses pembacaan data di latar belakang agar tidak membebani antarmuka utama.

3.4. Pengujian Sistem (Integration and System Testing)

Tahap pengujian dilakukan untuk memverifikasi kesesuaian sistem dengan kebutuhan awal menggunakan metode Black Box Testing. Fokus pengujian adalah validasi input-output pada fitur-fitur utama tanpa memeriksa struktur kode internal. Skenario pengujian meliputi: (1) Validasi akurasi hasil pencarian ayat, (2) Uji stabilitas rendering PDF saat di-zoom maksimal, (3) Uji fungsi penyimpanan bookmark setelah aplikasi ditutup dan dibuka kembali, dan (4) Uji navigasi antar halaman untuk memastikan tidak terjadi kegagalan aplikasi (force close).

3.5. Operasi dan Pemeliharaan (Operation and Maintenance)

Tahap akhir dari metode *Waterfall* ini mencakup proses distribusi dan pemeliharaan sistem pasca-pengembangan.

1. Operasi (Deployment): Aplikasi didistribusikan dalam format paket instalasi Android (.apk) yang dapat dipasang pada perangkat pengguna sasaran (santri/masyarakat).
2. Pemeliharaan (Maintenance): Dilakukan pemeliharaan korektif (*corrective maintenance*) apabila ditemukan *bug* atau kesalahan data setelah penggunaan jangka panjang, serta pemeliharaan adaptif untuk memastikan aplikasi tetap kompatibel dengan versi Android terbaru di masa depan. Mengingat penelitian ini berfokus pada

Juz 1, tahap ini juga mencakup rencana penambahan konten data untuk Juz 2 hingga Juz 30 pada iterasi pengembangan selanjutnya.

4. HASIL DAN PEMBAHASAN

Bagian ini memaparkan hasil implementasi dari perancangan yang telah dijelaskan pada bab metode penelitian. Pengembangan aplikasi dilakukan menggunakan Android Studio dengan bahasa Kotlin. Hasil pengembangan mencakup implementasi antarmuka pengguna (UI) dan implementasi fitur inti, serta temuan kendala teknis yang dihadapi selama proses pengembangan.

4.1. Hasil Implementasi Antarmuka (UI)

Implementasi antarmuka pengguna (UI) berhasil menerjemahkan desain *wireframe* yang dibuat menggunakan Figma menjadi *layout* XML yang fungsional di Android. Antarmuka aplikasi dirancang dengan pendekatan visual yang sederhana namun tetap merupakan realisasi desain visual yang mengadopsi identitas Pondok Modern Darussalam Gontor.

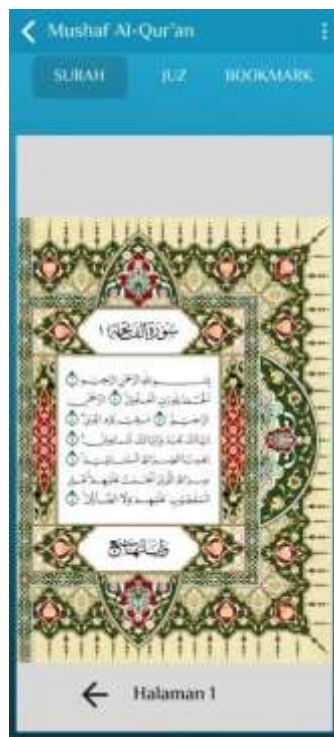
Hasil implementasi utama dapat dilihat pada Gambar 3. Saat aplikasi dijalankan, pengguna disambut dengan Halaman Utama (Gambar 3) yang menyajikan navigasi minimalis untuk memilih mode baca. Dari sana, pengguna dapat mengakses Daftar Surah (Gambar 4) yang memuat indeks 114 surat. Fitur inti aplikasi adalah Mode Mushaf (Gambar 5) yang menampilkan rendering PDF vektor berkualitas tinggi sehingga tulisan Khat Naskhi tetap tajam saat diperbesar. Selain itu, tersedia Mode Baca Per-Ayat (Gambar 6) yang memanfaatkan basis data JSON lokal untuk menampilkan teks Arab dan terjemahan dalam format daftar vertikal yang responsif.



Gambar 4. Implementasi Halaman Utama



Gambar 4. Implementasi Halaman Surah



Gambar 6. Implementasi Halaman Mushaf Al-Qur'an



Gambar 6. Implementasi Halaman Per-Ayat

4.2. Implementasi Logika Sistem

Di balik tampilan visual tersebut, terdapat arsitektur sistem yang bekerja untuk memastikan aplikasi berjalan optimal tanpa koneksi internet.

1. Strategi Manajemen Aset (Embedded Assets) Salah satu tantangan terbesar dalam pengembangan ini adalah menyeimbangkan kualitas visual dengan performa. Kami memutuskan untuk menanamkan seluruh file PDF kualitas tinggi dan basis data JSON langsung ke dalam paket aplikasi (APK). Konsekuensinya, ukuran akhir aplikasi membengkak menjadi **228 MB**. Meskipun angka ini terlihat besar untuk standar aplikasi teks biasa, ini adalah *trade-off* atau kompromi teknis yang disengaja. Dengan ukuran ini, pengguna hanya perlu mengunduh satu kali di awal, dan setelahnya mereka mendapatkan jaminan akses mushaf kualitas *High Definition* (HD) selamanya tanpa perlu membuang kuota internet lagi untuk memuat halaman.
2. Efisiensi Penyimpanan Bookmark Untuk fitur penanda halaman (*bookmark*), sistem tidak menggunakan database yang berat. Kami menerapkan mekanisme **SharedPreferences**. Karena data yang disimpan hanyalah informasi sederhana—yaitu nomor halaman terakhir yang dibuka—penggunaan *SharedPreferences* jauh lebih ringan dan cepat dieksekusi oleh prosesor ponsel dibandingkan harus membangun database SQLite yang kompleks. Ini membuat fitur *bookmark* terasa instan saat ditekan.

4.3. Hasil Pengujian Fungsional (Black Box Testing)

Pengujian Sistem Untuk memastikan aplikasi layak digunakan oleh santri dan masyarakat luas dengan berbagai tipe perangkat, kami melakukan pengujian fungsional (*Black Box Testing*). Pengujian ini dilakukan pada lingkungan perangkat fisik (*real device*) yang mewakili tiga segmen pasar berbeda, mulai dari spesifikasi rendah hingga tinggi.

Tabel 1. Lingkungan Perangkat Pengujian

Kategori Perangkat	Tipe Ponsel	Spesifikasi Utama
<i>High-End</i>	Samsung Galaxy S23	RAM 8 GB, Android 16
<i>Mid-Range</i>	Samsung Galaxy A33	RAM 8 GB, Android 15
<i>Entry-Level</i>	Samsung Galaxy A03	RAM 4 GB, Android 13

Pengujian dilakukan dengan menjalankan skenario penggunaan utama pada ketiga perangkat tersebut. Hasil pengujian dirangkum dalam tabel berikut:

Tabel 2. Hasil Pengujian Fungsional (Black Box)

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian Lapangan	Status
1	Kualitas Rendering: Melakukan <i>zoom-in</i> maksimal pada halaman mushaf PDF.	Tulisan tetap tajam, tidak buram/pecah, dan terbaca jelas.	Berhasil. Visual tajam di semua perangkat. Pada Samsung A03, rendering awal halaman membutuhkan waktu jeda sekitar 1-2 detik, namun setelah terbuka visual tetap lancar.	Valid
2	Akurasi Pencarian: Memasukkan kata kunci ayat pada kolom pencarian.	Sistem menampilkan daftar ayat yang mengandung kata tersebut dari database lokal.	Berhasil. Hasil pencarian muncul secara instan pada ketiga perangkat tanpa kendala koneksi.	Valid
3	Navigasi Halaman: Melakukan geser (<i>swipe</i>) cepat antar halaman mushaf.	Perpindahan halaman mulus (<i>smooth</i>) tanpa aplikasi tertutup sendiri (<i>force close</i>).	Berhasil. Navigasi berjalan lancar dan responsif.	Valid
4	Fungsi Bookmark: Menandai halaman, menutup aplikasi sepenuhnya, lalu membuka kembali.	Aplikasi mengingat halaman terakhir yang ditandai.	Berhasil. Data tersimpan akurat menggunakan mekanisme <i>SharedPreferences</i> .	Valid

4.4. Operasi dan Pemeliharaan

1. Operasi (Deployment): Aplikasi telah berhasil dikompilasi menjadi paket instalasi Android (APK) versi 1.0. Sesuai dengan target pengguna masyarakat umum, aplikasi ini dipersiapkan untuk didistribusikan secara luas melalui Google Play Store. Publikasi melalui toko aplikasi resmi ini menjamin keamanan instalasi bagi pengguna serta memungkinkan jangkauan akses tanpa batasan geografis, sehingga digitalisasi Mushaf Gontor dapat dimanfaatkan oleh umat Islam secara global.
2. Pemeliharaan (Maintenance): Strategi pemeliharaan saat ini berfokus pada Corrective Maintenance, yaitu perbaikan bug minor jika ditemukan pada perangkat spesifik. Selain itu, rencana Adaptive Maintenance telah disiapkan untuk menambah konten Juz 2 hingga Juz 30 pada pembaruan versi berikutnya tanpa mengubah arsitektur dasar sistem yang sudah teruji stabil.

Keberhasilan implementasi aplikasi Mushaf Gontor Juz 1 ini membuktikan bahwa pendekatan "Offline-First" adalah solusi yang paling relevan untuk menjawab masalah aksesibilitas yang dihadapi. Temuan menarik dari pengujian ini adalah mengenai performa pada perangkat *entry-level* (Samsung A03 dengan RAM 4GB). Meskipun terdapat sedikit jeda waktu saat memuat halaman PDF dibandingkan perangkat *flagship* (S23), aplikasi tetap berjalan stabil dan tidak mengalami *crash*. Hal ini menunjukkan bahwa keputusan menggunakan **PDF Vektor** dan penyimpanan **SharedPreferences**

sangat efektif dalam menjaga efisiensi memori, sehingga aplikasi ini inklusif dan dapat diakses oleh santri yang mungkin tidak memiliki ponsel spesifikasi tinggi.

Selain itu, ukuran aplikasi sebesar 228 MB yang awalnya dikhawatirkan akan membebani, ternyata memberikan nilai tambah berupa kemandirian akses. Pengguna tidak lagi bergantung pada sinyal internet yang seringkali tidak stabil di area pondok atau pedesaan. Dengan demikian, tujuan digitalisasi untuk melestarikan Mushaf Gontor sekaligus mempermudah aksesnya telah tercapai melalui arsitektur sistem ini.

5. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan mengenai "Rancang Bangun Aplikasi Mobile Mushaf Gontor Juz 1 Berbasis Android", dapat ditarik beberapa kesimpulan sebagai berikut:

1. **Kemandirian Sistem (Offline-Capability):** Penelitian berhasil merancang aplikasi dengan arsitektur *embedded assets* yang beroperasi 100% tanpa koneksi internet, menjamin aksesibilitas mushaf di area minim sinyal meskipun berdampak pada ukuran instalasi sebesar 228 MB.
2. **Preservasi Visual:** Penerapan teknologi rendering PDF berbasis vektor terbukti efektif mendigitalisasi Mushaf Gontor, menjaga ketajaman detail Khat Naskhi tetap tinggi dan tidak pecah (pixelated) saat fitur perbesaran (zoom) digunakan.
3. **Kompatibilitas Perangkat:** Hasil pengujian Black Box pada perangkat entry-level (Samsung A03) hingga high-end (Samsung S23) menunjukkan bahwa sistem berjalan stabil dan inklusif berkat efisiensi penggunaan memori dan mekanisme penyimpanan ringan SharedPreferences.

DAFTAR PUSTAKA

- [1] S. Ariyadi, "THE QUR'AN, DIGITAL TOOLS AND THE TRANSFORMATION OF," vol. 2, no. 3, pp. 2793–2800, 2025.
- [2] Nurkholis and A. B. Susanto, "Penggunaan Aplikasi Al Qur'an Digital Dalam Pembelajaran Pada Mata Pelajaran Qur'an Hadist," *Islam EDU J. Ilm. Pendidik. Agama Islam*, vol. 2, no. 1, pp. 14–26, 2024.
- [3] M. Jaedi, "Pentingnya Memahami Al-Qur'an dan Ilmu Pengetahuan," *J. Pendidik. dan Stud. Islam*, vol. 5, no. 1, pp. 62–70, 2019, doi: 10.5281/zenodo.2618950.
- [4] A. A. Rokhim and M. Munawir, "Penilaian Hasil Belajar Tahfidzul Qur'an Usia Dewasa Di Yayasan Griya Al Qur'an Surabaya," *Tawazun J. Pendidik. Islam*, vol. 14, no. 3, p. 285, 2021, doi: 10.32832/tawazun.v14i3.5710.
- [5] D. lin Sriwati, "Gambaran Manuskrip Al- Qur' an Hj . Rippun di Desa Pambusuang Kabupaten Polewali Mandar (Studi Kodikologi dan Filologi)," *El-Adabi J. Stud. Islam*, vol. 02, no. 01, pp. 89–111, 2023, [Online]. Available: <https://jurnal.stainidaeladabi.ac.id/index.php/eladabi/article/view/40/50>
- [6] A. F. Aini, A. Prajawinanti, A. Sulton, and A. Z. Alifa, "Digital Preservation of Qur'anic Manuscript," *Khazanah al-Hikmah J. Ilmu Perpustakaan, Informasi, dan Kearsipan*, vol. 13, no. 1, pp. 37–53, 2025, doi: 10.24252/v13i1a4.
- [7] H. Zhu, J. Cao, Y. Xiao, Z. Chen, and Y. J. Zhang, "Supplementary Material for : TCB-

- spline-based Image Vectorization,” vol. 41, no. 3, 2022.
- [8] F. Feronika, M. Masrizal, and I. R. Munthe, “Evaluation of Machine Learning Using The K-NN Algorithm to determine The Quality of Meat before consumption,” *J. Ris. Inform.*, vol. 5, no. 2, pp. 171–176, 2023, doi: 10.34288/jri.v5i2.467.
- [9] N. Hamzah, D. N. A. Halim, M. H. Hassan, and A. Ariffin, “Android Application for Children to Learn Basic Solat The Approach of Behaviourism Learning Theory,” *iJIM*, vol. 13, no. 7, pp. 69–79, 2019.
- [10] . S., S. Aisyah, and D. Lase, “Development of Assistant Quran Application With Real-time Voice Commands,” *KnE Eng.*, vol. 2024, pp. 429–433, 2024, doi: 10.18502/keg.v6i1.15412.
- [11] I. W. Ridhoni and M. Sholeh, “Application of Augmented Reality-Based Solar System Recognition Using Metaverse Studio,” *Int. J. Appl. Sci. Smart Technol.*, vol. 5, no. 1, pp. 55–66, 2023, doi: 10.24071/ijasst.v5i1.5447.
- [12] D. Kasoni, L. Liesnaningsih, and F. F. Afif, “Perancangan Sistem Pembelajaran Al-Quran Berbasis Android Dengan Metode Extreme Programming,” *JIKA (Jurnal Inform.)*, vol. 8, no. 1, p. 89, 2024, doi: 10.31000/jika.v8i1.10270.
- [13] A. P. Pratiwi and M. Taufiq, “Kompetensi Literasi Perspektif Al-Qur'an (Studi Komparasi Mufasir Minangkabau),” *Tsaqofah*, vol. 3, no. 5, pp. 948–957, 2023, doi: 10.58578/tsaqofah.v3i5.1753.
- [14] J. M. Arib and S. Mokodenseho, “Mushaf Bone: Telaah Aspek Kodikologi, Tulisan, Teks dan Visual Al-Qur'an,” *OSF Prepr.*, pp. 1–17, 2020.