

PENGEMBANGAN APLIKASI AL-QUR'AN DIGITAL BERBASIS MOBILE DENGAN INTEGRASI CHATBOT TANYA JAWAB ISLAMI MENGUNAKAN METODOLOGI *EXTREME PROGRAMMING*

Muhammad Rafif Musyaffa, Bayu Prasetyo, Mutiara Andayani Komara

Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana Purwakarta
Jalan Raya Cikopak No. 53, Sadang 41151 Kabupaten Purwakarta, Jawa Barat
bayu.prasetyo@wastukencana.ac.id

ABSTRAK

Perkembangan teknologi *mobile* telah membuka peluang baru dalam penyediaan layanan keagamaan berbasis digital. Aplikasi AL-Quran yang ada umumnya menyediakan fitur dasar seperti membaca dan terjemahan, namun belum banyak yang menawarkan interaksi langsung untuk menjawab pertanyaan secara keislaman secara dinamis. Penelitian ini bertujuan untuk merancang dan membangun aplikasi AL-Quran digital berbasis *mobile* yang terintegrasi dengan fitur chatbot tanya jawab Islami. Pengembangan aplikasi dilakukan menggunakan metodologi *Extreme Programming* (XP), yang mendukung proses iteratif dan fleksibel. Aplikasi dikembangkan untuk platform Android dengan bahasa pemrograman Kotlin, serta memanfaatkan *Gemini API* sebagai integrasi *chatbot* yang memberikan jawaban dari sumber-sumber Islami yang relevan. Pengujian dilakukan dengan pendekatan *Blackbox Testing* untuk memastikan fungsionalitas sistem berjalan sesuai harapan. Aplikasi yang dihasilkan menyajikan fitur teks dan audio Al-Qur'an, pencarian, penanda (*bookmark*), serta interaksi *chatbot* untuk membantu pengguna memahami ajaran Islam secara lebih mudah dan praktis.

Kata kunci: Aplikasi Mobile, Al-Qur'an Digital, Chatbot, *Extreme Programming*, *Gemini API*

1. PENDAHULUAN

Perkembangan teknologi *mobile* telah membuka berbagai kemungkinan baru dalam pengembangan layanan berbasis digital, termasuk dalam bidang keagamaan. Salah satu wujudnya adalah hadirnya aplikasi Al-Qur'an digital yang mempermudah akses umat Islam terhadap kitab suci mereka melalui perangkat Android. Aplikasi semacam ini menjadi alternatif praktis bagi pengguna yang ingin membaca, memahami, dan mendalami isi Al-Qur'an di mana saja dan kapan saja.

Namun, sebagian besar aplikasi Al-Qur'an digital yang tersedia masih terbatas pada penyediaan teks, terjemahan, dan audio ayat. Fitur interaktif yang memungkinkan pengguna bertanya dan mendapatkan penjelasan langsung terkait ayat atau hukum Islam masih sangat jarang ditemukan. Padahal, kebutuhan akan pemahaman ajaran Islam secara lebih mendalam sering kali muncul dalam aktivitas membaca Al-Qur'an. Pengguna tidak hanya ingin membaca, tetapi juga ingin mendapatkan penjelasan yang relevan dan mudah dipahami, sesuai konteks.

Untuk menjawab kebutuhan tersebut, integrasi teknologi chatbot dalam aplikasi Al-Qur'an dapat menjadi solusi inovatif. *Chatbot* memungkinkan pengguna untuk berinteraksi langsung dengan sistem melalui antarmuka percakapan. Melalui teknologi ini, pengguna dapat mengajukan pertanyaan keislaman dan mendapatkan respons secara otomatis, cepat, dan tetap merujuk pada sumber yang valid.

Penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi Al-Qur'an digital berbasis *mobile* yang dilengkapi dengan fitur chatbot tanya jawab Islami. Pengembangan aplikasi dilakukan dengan pendekatan metodologi *Extreme*

Programming (XP), yang menekankan pada fleksibilitas, iterasi cepat, dan komunikasi yang intensif dalam proses pengembangan. Aplikasi dibangun untuk platform Android menggunakan bahasa pemrograman Kotlin. *Chatbot* diintegrasikan melalui layanan *Gemini API* yang memungkinkan pemberian jawaban dengan sumber Islami yang relevan. Pengujian sistem dilakukan menggunakan metode *Blackbox Testing* untuk memastikan seluruh fitur berfungsi dengan baik.

Melalui pengembangan ini, dihasilkan sebuah aplikasi Al-Qur'an digital yang tidak hanya menyajikan teks dan audio ayat, tetapi juga dilengkapi dengan fitur pencarian, penanda (*bookmark*), dan chatbot Islami yang mendukung interaksi dinamis. Aplikasi ini diharapkan dapat menjadi media pembelajaran Islam yang informatif, praktis, dan sesuai dengan kebutuhan pengguna di era digital.

2. TINJAUAN PUSTAKA

2.1. Aplikasi

Aplikasi merupakan sebuah perangkat lunak yang dirancang untuk menjalankan fungsi atau tugas tertentu di dalam komputer, smartphone, maupun perangkat digital lainnya. Aplikasi ini dibuat dengan tujuan untuk membantu pengguna dalam menyelesaikan pekerjaan, mengakses informasi, atau menikmati hiburan sesuai dengan kebutuhan yang telah ditentukan sebelumnya.[1]

2.2. Aplikasi Mobile

Aplikasi *mobile* merupakan gabungan dari dua istilah, yaitu *application* dan *mobile*. Kata *application* merujuk pada suatu program yang dirancang khusus untuk menjalankan fungsi tertentu, baik untuk

pengguna maupun aplikasi lain. Sementara itu, *mobile* berarti bergerak atau berpindah tempat, yang dalam konteks ini merujuk pada perangkat yang dapat dibawa ke mana saja.[2]

2.3. Android

Android dapat dipahami sebagai kumpulan perangkat lunak yang dirancang untuk perangkat *mobile*, yang mencakup sistem operasi, *middleware*, dan aplikasi inti yang juga dikembangkan oleh Google. Untuk pengembangan aplikasi di platform *Android*, tersedia SDK (*Software Development Kit*) yang menggunakan bahasa pemrograman *Java*. [3]

2.4. AL-Quran Digital

Al-Qur'an digital merupakan sebuah aplikasi yang memuat isi *Al-Qur'an* dan dapat diakses melalui perangkat elektronik seperti gadget atau CD. *Al-Qur'an* digital juga dapat dipahami sebagai alat berbasis teknologi yang dirancang untuk mempermudah proses belajar membaca *Al-Qur'an*, karena di dalamnya tersedia teks lengkap 30 juz, audio lantunan ayat, terjemahan, serta berbagai fitur pendukung lainnya.[4]

2.5. Chatbot

Chatbot merupakan salah satu bentuk penerapan kecerdasan buatan yang dirancang untuk meniru kemampuan berpikir manusia, terutama dalam hal mempelajari informasi baru. Teknologi ini memungkinkan sistem komputer untuk mengenali dan memahami bahasa manusia dalam bentuk aslinya, baik lisan maupun tulisan. Tujuan utama dari pengembangan *chatbot* adalah untuk menciptakan interaksi verbal yang alami antara manusia dan mesin. Dengan kemampuan memahami bahasa alami serta menarik kesimpulan secara otomatis sebagai respons terhadap pertanyaan yang diberikan, *chatbot* telah diprogram untuk menjadi alat komunikasi cerdas yang semakin mendekati percakapan manusia secara nyata.[5]

2.6. Gemini

Gemini AI adalah teknologi kecerdasan buatan (AI) yang dikembangkan oleh Google, dirancang untuk mendukung berbagai aplikasi dan proses pembelajaran. Model ini memiliki kemampuan untuk memproses dan menganalisis data dalam bentuk teks, gambar, dan *suara*, sehingga dapat memberikan solusi yang lebih komprehensif dalam berbagai konteks.[6] dalam konteks ini, *Gemini AI* dapat digunakan salah satunya untuk menjawab pertanyaan, memberikan rekomendasi.

Agar dapat menggunakan *Gemini* dalam aplikasi *al quran* ini, diperlukan akses ke API (*Application Programming Interface*) Key dari *Gemini*. Google telah menyediakan akses ke API *Gemini*, yang dapat dimanfaatkan oleh pengembang perangkat lunak serta pihak bisnis yang ingin mengintegrasikan kemampuan *Gemini* ke dalam aplikasi mereka.[7]

2.7. Android Studio

Android Studio adalah lingkungan pengembangan terpadu (IDE) yang dirancang khusus untuk membuat aplikasi *Android* dan dikembangkan oleh Google. IDE ini merupakan evolusi dari *Eclipse IDE* dan dibangun dengan dasar dari *IntelliJ IDEA*, sebuah IDE *Java* yang sangat populer.[8]

Android Studio tidak hanya berfungsi sebagai editor kode berbasis *IntelliJ* dan alat pengembang yang handal, tetapi juga menyediakan berbagai fitur yang dirancang untuk meningkatkan produktivitas Anda dalam membangun aplikasi *Android*. Dengan demikian, platform ini memberikan kemudahan dan efisiensi dalam proses pengembangan, selaras dengan pendekatan pragmatis yang telah diusung oleh *Kotlin*. [9]

2.8. Metode Extreme Programming

Extreme Programming (XP) adalah salah satu pendekatan dalam pengembangan perangkat lunak yang berasal dari metode *Agile*. Metodologi ini diperkenalkan oleh Kent Beck pada tahun 1996 saat ia menangani proyek sistem penggajian milik perusahaan *Chrysler*. Proyek tersebut sempat mengalami hambatan serius akibat kompleksitas sistem dan kegagalan dalam tahap pengujian, sehingga mendorong lahirnya XP sebagai solusi pengembangan yang lebih adaptif dan terstruktur.[10]

2.9. Unified Modeling Language

Unified Modeling Language adalah alat visual yang digunakan untuk memodelkan, merancang, dan mengembangkan perangkat lunak dengan pendekatan berbasis objek. Selain itu, UML berfungsi sebagai standar untuk mendokumentasikan desain, menyerupai cetak biru yang mencakup proses bisnis serta detail penulisan kelas-kelas dalam sebuah bahasa pemrograman.[11]

2.10. BlackBox Testing

Blackbox merupakan metode pengujian perangkat lunak yang berfokus pada aspek fungsional dari sebuah aplikasi. Dalam pendekatan ini, penguji tidak perlu mengetahui detail internal atau struktur kode, melainkan hanya mengevaluasi apakah fungsi-fungsi dalam aplikasi berjalan sesuai dengan yang ditentukan berdasarkan kebutuhan. Penguji akan merancang test case dan mengamati hasil keluaran dari input yang diberikan, untuk memastikan bahwa sistem memberikan respons yang sesuai dengan yang diharapkan.[12]

2.11. Penelitian Terdahulu

Beberapa penelitian terdahulu yang relevan dalam pengembangan aplikasi *Al-Qur'an* digital dan penerapan metodologi *Extreme Programming* (XP) dapat dijelaskan sebagai berikut:

Penelitian oleh [13] mengembangkan aplikasi *Al-Qur'an* berbasis *Flutter* dengan pendekatan *Clean Architecture*. Hasilnya menunjukkan tingkat

maintainability yang tinggi, dengan fitur-fitur seperti pembacaan ayat, tafsir, dan bookmark yang lengkap. Pendekatan modularitas turut meningkatkan efisiensi dan kejelasan struktur kode.

Selanjutnya, [14] mengembangkan aplikasi hafalan Al-Qur'an berbasis Android menggunakan Google Speech API. Aplikasi ini mampu mengubah suara menjadi teks dengan akurasi hingga 92%, serta mendapatkan skor usability tinggi berdasarkan metode System Usability Scale (SUS), membuktikan aplikasi mudah dipahami dan digunakan.

Sementara itu, dalam hal metodologi pengembangan perangkat lunak, [15] menggunakan metode Extreme Programming dalam merancang sistem pembelajaran Al-Qur'an untuk santri pemula. Aplikasi tersebut mencakup fitur tajwid dasar, rekaman bacaan, dan memungkinkan pembelajaran mandiri tanpa keterbatasan tempat dan waktu.

Penelitian oleh [16] juga menerapkan XP dalam pengembangan aplikasi pengenalan organisasi bagi mahasiswa baru. Metode XP terbukti memberikan fleksibilitas tinggi terhadap perubahan kebutuhan pengguna, serta mempercepat proses pengembangan melalui perencanaan iteratif dan komunikasi aktif antar tim.

3. METODE PENELITIAN

3.1. Metode Penelitian



Gambar 1. Metode Penelitian

Penelitian ini menggunakan metode *Extreme Programming* (XP) yang terdiri dari empat tahapan utama, yaitu *planning*, *designing*, *coding*, dan *testing*. Metode ini dipilih karena sifatnya yang fleksibel dan iteratif, memungkinkan pengembangan aplikasi menyesuaikan kebutuhan pengguna secara dinamis. Setiap tahapan dilakukan secara berulang dan berkelanjutan, sehingga mendukung pengujian sejak awal dan meminimalkan risiko kesalahan. Alur metode penelitian digambarkan pada Gambar 1.

3.2. Studi Pustaka

Studi pustaka dilakukan dengan menelaah referensi yang relevan terkait pengembangan aplikasi mobile, metode *Extreme Programming* (XP), penggunaan *chatbot* Islami, serta integrasi teknologi *Artificial Intelligence* melalui layanan seperti Gemini API. Sumber referensi meliputi jurnal ilmiah, prosiding, buku, artikel teknologi, dan dokumentasi resmi.

3.3. Metode Extreme Programming

Pada tahap ini dilakukan pengumpulan data dan analisis masalah yang dilakukan. Informasi yang dikumpulkan harus akurat dan relevan agar tetap sesuai dengan fokus topik yang dikaji.

3.4. Planning

Tahap ini diawali dengan identifikasi kebutuhan pengguna. Kebutuhan yang dikumpulkan antara lain fitur baca Al-Qur'an, pencarian ayat, pemutaran audio, bookmark, berbagi ayat, dan fitur *chatbot* Islami. Kebutuhan ini dituangkan dalam bentuk *user story* yang menjadi acuan pengembangan secara bertahap.

3.5. Design

Tahap desain meliputi perancangan antarmuka pengguna (*User Interface*) yang intuitif dan struktur sistem menggunakan pendekatan *Model-View-ViewModel* (MVVM). Selain itu, model sistem divisualisasikan melalui diagram UML seperti use case, activity diagram, dan class diagram untuk menggambarkan interaksi dan alur logika aplikasi.

3.6. Coding

Implementasi sistem dilakukan dengan menggunakan bahasa pemrograman Kotlin di Android Studio. Fitur dasar seperti daftar surah dan ayat diimplementasikan terlebih dahulu, kemudian dilanjutkan dengan fitur audio, bookmark, pencarian, dan integrasi *chatbot* menggunakan Gemini API.

3.7. Testing

Pengujian sistem dilakukan menggunakan metode *Blackbox Testing*, yaitu dengan menguji setiap fitur dari sisi pengguna tanpa melihat struktur internal program. Tujuan pengujian adalah memastikan seluruh fungsi berjalan dengan benar dan sesuai kebutuhan pengguna.

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Pada tahap ini, data dikumpulkan dan masalah yang ditemukan dianalisis. Agar tetap pada topik dan tidak menyimpang dari pembahasannya, Langkah-langkah berikut ditetapkan:

4.2. Studi Pustaka

Peneliti mengumpulkan berbagai informasi dengan mempelajari buku, jurnal, dan hasil penelitian terdahulu yang berkaitan dengan perancangan aplikasi

Al-Qur'an digital menggunakan metode pengembangan *Extreme Programming* (XP), agar pembahasan tetap sesuai dengan topik dan berdasarkan referensi yang mendukung sehingga menghasilkan sistem yang tepat guna.

4.3. Planning

Pada tahap perencanaan, dilakukan perumusan kebutuhan fungsional yang akan dimuat dalam aplikasi. Kebutuhan tersebut meliputi tampilan daftar surah, detail ayat, pencarian, pemutaran audio, bookmark, berbagi ayat, dan chatbot Islami. Selain itu, dilakukan juga analisis terhadap perangkat keras dan lunak yang dibutuhkan untuk menjalankan aplikasi.

a. Kebutuhan fungsional merupakan fitur utama yang harus dimiliki oleh sistem untuk menjalankan perannya dengan benar. Dalam penelitian ini, fitur yang dirancang meliputi:

- Menampilkan daftar surah dan ayat Al-Qur'an.
- Menyediakan terjemahan ayat dalam Bahasa Indonesia.
- Memutar audio setiap ayat.
- Menyimpan ayat ke dalam *bookmark*
- Mencari surah atau ayat berdasarkan kata kunci.
- Membagikan ayat melalui media social
- Chatbot Islami yang dapat menjawab pertanyaan pengguna

b. Kebutuhan Perangkat

Kebutuhan ini dibagi menjadi perangkat pengguna dan perangkat pengembang:

Perangkat pengguna:

- Smartphone Android minimal versi 9.
- RAM minimal 3 GB.
- Memori kosong minimal 100 MB.

Perangkat pengembang:

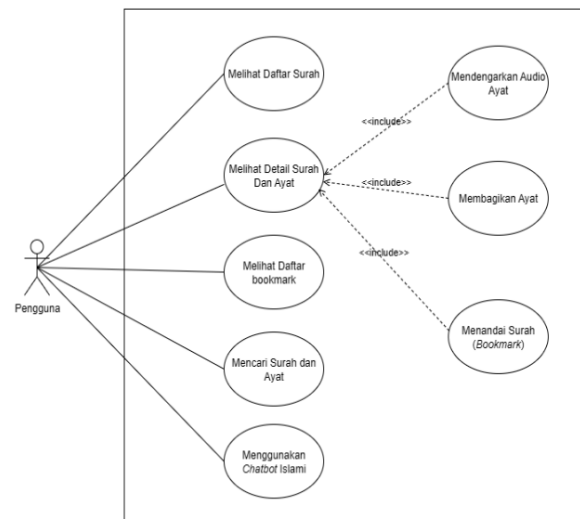
- Laptop/PC dengan RAM minimal 8 GB dan prosesor Intel Core i5.
- Android Studio sebagai IDE.
- Gemini API sebagai layanan chatbot.
- Emulator Android atau perangkat fisik untuk pengujian.

4.4. Desain

Tahap desain merupakan proses merancang struktur sistem dan antarmuka pengguna sebelum pengembangan dilakukan. Desain ini bertujuan untuk memastikan alur aplikasi berjalan sesuai kebutuhan, serta memudahkan proses implementasi dan pemeliharaan sistem.

4.5. Use Case Diagram

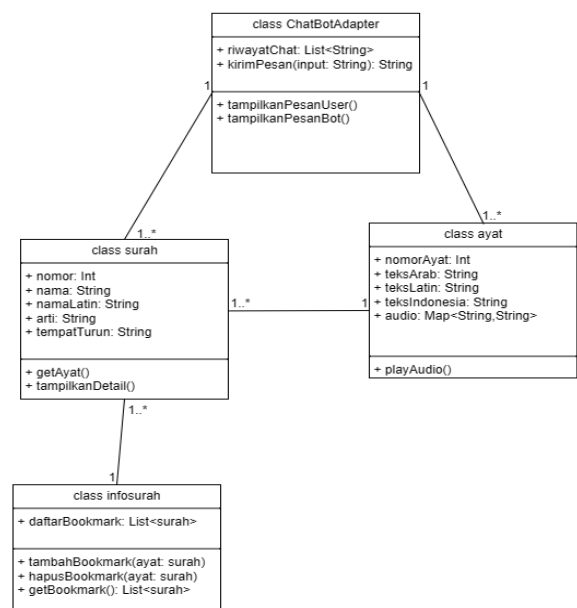
Diagram ini menggambarkan interaksi antara pengguna (aktor) dengan sistem. Dalam aplikasi ini, aktor utama adalah pengguna aplikasi yang dapat melakukan aktivitas seperti melihat daftar surah, membaca ayat, memutar audio, menandai ayat sebagai bookmark, mencari ayat tertentu, membagikan ayat, serta bertanya kepada chatbot.



Gambar 2. Use Case Diagram

4.6. Class Diagram

Desain sistem dilakukan untuk menggambarkan struktur dan hubungan antar komponen dalam aplikasi. Salah satu bentuk perancangan yang digunakan adalah *Class Diagram*, yang berfungsi untuk memodelkan objek-objek utama dalam sistem beserta atribut, metode, dan relasi antar kelas.



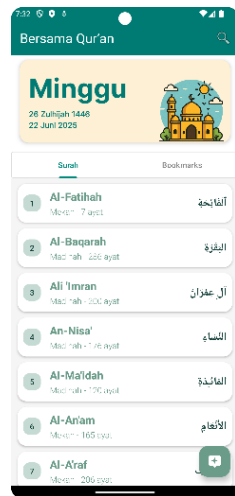
Gambar 3. Class Diagram

4.7. Coding

Tahap coding merupakan proses implementasi desain sistem ke dalam bentuk kode program menggunakan bahasa pemrograman Kotlin dan lingkungan pengembangan Android Studio. Proses ini dilakukan berdasarkan hasil dari tahap perencanaan dan perancangan sebelumnya, baik dari sisi fungsionalitas maupun struktur data.

Aplikasi dibangun menggunakan arsitektur *Model-View-ViewModel* (MVVM) untuk memisahkan antara logika program, tampilan, dan manajemen data.

Pendekatan ini dipilih agar kode lebih mudah dikelola dan dikembangkan secara berkelanjutan. Sebagai gambaran hasil implementasi, berikut ditampilkan salah satu tampilan antarmuka dari aplikasi Al-Qur'an digital yang telah dikembangkan.



Gambar 4. Tampilan Fitur Aplikasi

Dari Gambar 4 dapat dilihat bahwa antarmuka dirancang secara sederhana dan responsif, dengan menampilkan daftar surah yang dapat dipilih pengguna. Fitur pencarian dan navigasi utama juga ditampilkan secara intuitif untuk memudahkan akses ke fungsi-fungsi lainnya seperti audio, *bookmark*, dan chatbot Islami.

4.8. Testing

Tahap pengujian dilakukan untuk memastikan bahwa seluruh fungsi dalam aplikasi berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Pengujian dilakukan menggunakan metode Blackbox Testing, yaitu dengan memfokuskan pengujian pada keluaran sistem berdasarkan masukan yang diberikan, tanpa memperhatikan struktur internal dari kode program.

Pengujian dilakukan pada fitur-fitur utama aplikasi, dan hasilnya disajikan dalam tabel berikut:

Tabel 1. Pengujian

No	Pengujian Aplikasi	
1	Fitur yang Diuji	Melihat Daftar Surah
	Langkah Pengujian	Pengguna membuka halaman utama aplikasi
	Input dari Pengguna	Aplikasi dijalankan
	Hasil yang Diharapkan	Daftar seluruh surah Al-Qur'an ditampilkan sesuai urutan mushaf
	Hasil Aktual	Daftar surah tampil sesuai urutan mushaf
	Status	Berhasil
2	Fitur yang Diuji	Melihat Detail Surah dan Ayat
	Langkah Pengujian	Pengguna memilih salah satu surah
	Input dari Pengguna	Klik salah satu surah dari daftar
	Hasil yang Diharapkan	Ditampilkan ayat lengkap beserta terjemahan dan teks latin
	Hasil Aktual	Semua ayat tampil sesuai surah yang dipilih
	Status	Berhasil
3	Fitur yang Diuji	Mendengarkan Audio Ayat
	Langkah Pengujian	Pengguna menekan tombol play pada ayat tertentu
	Input dari Pengguna	Tekan tombol play pada ayat
	Hasil yang Diharapkan	Audio ayat yang dipilih diputar
	Hasil Aktual	Audio diputar dengan benar
	Status	Berhasil
4	Fitur yang Diuji	Membagikan Ayat
	Langkah Pengujian	Pengguna menekan tombol share pada ayat
	Input dari Pengguna	Tekan tombol share pada ayat
	Hasil yang Diharapkan	Menu berbagi terbuka dan ayat berhasil dibagikan
	Hasil Aktual	Ayat berhasil dibagikan ke aplikasi lain
	Status	Berhasil
5	Fitur yang Diuji	Menandai Surah (Bookmark)
	Langkah Pengujian	Pengguna menekan ikon bookmark pada ayat tertentu
	Input dari Pengguna	Tekan ikon bookmark pada ayat
	Hasil yang Diharapkan	Ayat disimpan ke dalam daftar bookmark
	Hasil Aktual	Bookmark berhasil ditambahkan
	Status	Berhasil
6	Fitur yang Diuji	Melihat Daftar Bookmark
	Langkah Pengujian	Pengguna membuka halaman bookmark
	Input dari Pengguna	Buka halaman bookmark
	Hasil yang Diharapkan	Daftar ayat yang ditandai sebelumnya ditampilkan
	Hasil Aktual	Bookmark ditampilkan dengan benar
	Status	Berhasil

No	Pengujian Aplikasi	
7	Fitur yang Diuji	Mencari Surah dan Ayat
	Langkah Pengujian	Ketik kata kunci dalam kolom pencarian
	Input dari Pengguna	Contoh: "Al-Ikhlash", "rezeki"
	Hasil yang Diharapkan	Sistem menampilkan hasil ayat atau surah yang relevan
	Hasil Aktual	Hasil pencarian sesuai dengan kata kunci
	Status	Berhasil
8	Fitur yang Diuji	Menggunakan Chatbot Islami
	Langkah Pengujian	Pengguna mengetik pertanyaan melalui fitur chatbot
	Input dari Pengguna	Contoh: "Apa itu zakat?"
	Hasil yang Diharapkan	Sistem menampilkan jawaban sesuai konteks dari chatbot
	Hasil Aktual	Jawaban tampil cepat dan sesuai konteks
	Status	Berhasil

5. KESIMPULAN DAN SARAN

Kesimpulan yang diperoleh dari penelitian ini adalah berhasil dikembangkan aplikasi Al-Qur'an digital berbasis mobile yang tidak hanya menyediakan fitur baca dan dengar ayat, tetapi juga dilengkapi dengan chatbot Islami interaktif berbasis Gemini API untuk menjawab pertanyaan keagamaan. Pengembangan menggunakan metode *Extreme Programming (XP)* memungkinkan proses iteratif dan fleksibel, sedangkan hasil pengujian *Black Box* menunjukkan seluruh fitur berjalan sesuai kebutuhan. Untuk pengembangan selanjutnya, disarankan agar chatbot ditingkatkan dengan referensi yang lebih luas dan spesifik, dilakukan pengujian langsung kepada pengguna (*user testing*), serta ditambahkan fitur edukatif seperti tafsir, pelafalan, dan pengelolaan data berbasis cloud untuk meningkatkan pengalaman pengguna secara menyeluruh.

DAFTAR PUSTAKA

- [1] N. K. Dewi, B. H. Irawan, E. Fitry, and A. S. Putra, "Konsep Aplikasi E-Dakwah Untuk Generasi Milenial Jakarta," *J. IKRA-ITH Inform.*, vol. 5, no. 2, pp. 26–33, 2021.
- [2] Pratama and M. Kamisutara, "Pengembangan Sistem Informasi Akademik Berbasis Mobile Menggunakan Flutter Di Universitas Narotama Surabaya," *Netw. Eng. Res. Oper.*, vol. 6, no. 2, p. 145, 2021, doi: 10.21107/nero.v6i2.238.
- [3] S. Eko, "Perancangan Aplikasi Pengenalan Budaya Nusantara Berbasis Android Dengan Metode Rad," *J. Ilmu Komput. JIK*, vol. 5, no. 01, pp. 30–39, 2022.
- [4] K. Nabilah, A. Mutiara, D. Resti, and Shela Amelia, "Pengaruh Aplikasi Al-Qur'an Digital Terhadap Motivasi Beribadah Mahasiswa Di Purwakarta," *Al Qodiri J. Pendidikan, Sos. dan Keagamaan*, vol. Volume 21, pp. 30–48, 2023.
- [5] B. Prasajo, M. Huda, I. N. Khasanah, and E. Wahyuningsih, "Aplikasi Chatbot Berbasis Telegram untuk Universitas Ma ' Arif," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 893–902, 2024, [Online]. Available: <https://journal.eng.unila.ac.id/index.php/jitet/article/view/4013>
- [6] N. Rachmat and D. P. Kesuma, "Implementasi Large Language Models Gemini Pada Pengembangan Aplikasi Chatbot Berbasis Android," *J. Ilmu Komput.*, vol. 4, no. 1, p. 2024, 2024, doi: 10.31314/juik.v4i1.2831.
- [7] A. Nazarius, F. Saputra, N. Noor Kamala sari, and V. Handrianus Pranatawijaya, "Penerapan Gemini Ai Dalam Pembuatan Deskripsi Produk E-Commerce," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3721–3725, 2024, doi: 10.36040/jati.v8i3.9670.
- [8] D. Rafa, E. Dyar Wahyuni, and A. Anjani Arifiyanti, "Rancang Bangun Aplikasi Donor Darah Darurat Donora Berbasis Android Dengan Konsep Gamifikasi Menggunakan Kotlin," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, pp. 3009–3020, 2024, doi: 10.23960/jitet.v12i3.5025.
- [9] R. Ditha, S. T. Faulina, and Wisnumurti, "Rancang Bangun Aplikasi Layanan Pengaduan Pada Dinas Pendidikan Kabupaten Oku Berbasis Android Menggunakan Android Studio," *Jik*, vol. 14Faulina, no. 2, pp. 25–35, 2023, [Online]. Available: <https://journal.unmaha.ac.id/index.php/jik/article/view/252/214>
- [10] L. S. Ambarsari, W. Puspitasari, and A. Syahrina, "Perancangan Modul Landing Page Dan Pembayaran Pada Website Pahamee Tentang Kesehatan Mental Menggunakan Metode Extreme Programming," *e-Proceeding Eng.*, vol. 8, no. 5, p. 9639, 2021.
- [11] I. Rohmah, A. Voutama, and A. Voutama, "Perancangan Sistem Informasi Laundry Berbasis Web Pada Perusahaan Rumah Laundry Menggunakan Uml," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 4, pp. 5691–5699, 2024, doi: 10.36040/jati.v8i4.9916.
- [12] B. B. Sasongko, F. Malik, F. Ardiansyah, A. F. Rahmawati, F. D. Adhinata, and D. P. Rakhmadani, "Pengujian Blackbox Menggunakan Teknik Equivalence Partitions pada Aplikasi Petgram Mobile," *J. ICTEE*, vol. 2, no. 1, p. 10, 2021, doi: 10.33365/jictee.v2i1.1012.
- [13] Azziqra and I. Nuryasin, "IMPLEMENTASI CLEAN ARCHITECTURE PADA APLIKASI MOBILE AL-QURAN BERBASIS FLUTTER," vol. 8, no. 2, pp. 369–380, 2024.

- [14] M. Assisi, A. Septiarini, A. H. Kridalaksana, and M. Wati, "Rancang Bangun Aplikasi Hafalan Al-Quran dengan Google Speech API Berbasis Android," *J. Rekayasa Teknol. Inf.*, vol. 6, no. 1, p. 26, 2022, doi: 10.30872/jurti.v6i1.8006.
- [15] 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.
- [16] M. F. F. Rahman, K. Darussalam, R. C. Saphira, and F. Purwani, "Implementasi Extreme Programming Dalam Pengembangan Aplikasi Mobile Pengenalan Organisasi Pada Masa Orientasi Mahasiswa," *Just IT J. Sist. Informasi, Teknol. Inform. dan Komput.*, vol. 14, no. 2, pp. 128–132, 2024.