

Implementasi *Automatic Speech Recognition* Pada Penilaian Hafalan Al-Quran Dengan Metode Muroja'ah Berbasis *Android*

Alun Sujjada¹, Gina Prnama Insany², Muhamad Fajar Nugraha^{*3}

^{1,2,3}Program Studi Teknik Informatika, Fakultas Teknik Komputer dan Desain, Universitas Nusa Putra

Email : ¹alun.sujjada@nusaputra.ac.id, ²gina.purnama@nusaputra.ac.id,
^{*3}fajar.nugraha_ti20@nusaputra.ac.id

Abstrak

Penelitian ini mengimplementasikan teknologi Pengenalan Ucapan Otomatis (*Automatic Speech Recognition/ASR*) dalam proses penilaian hafalan Al-Qur'an untuk mengatasi masalah meningkatnya jumlah orang yang tidak lancar membaca Al-Qur'an. Dengan ASR, komputer dapat mengenali dan menuliskan ucapan manusia secara akurat, sehingga dapat meningkatkan kualitas dan ketepatan pelafalan serta memberikan umpan balik yang konstruktif kepada para pembelajar. Aplikasi berbasis *Android* yang menggunakan layanan ASR dari *Google Speech API* membuat pengguna dapat membacakan ayat-ayat Al-Qur'an dan secara otomatis mengenali serta mengevaluasi apa yang telah mereka hafal. Harapannya, penggunaan teknologi ASR dapat meningkatkan kemampuan menghafal Al-Qur'an secara lebih efektif dan fleksibel tanpa memerlukan bantuan pengajar langsung. Sistem ini dikembangkan menggunakan metode *Rapid Application Development (RAD)* dengan pengujian *Blackbox Testing*.

Kata Kunci—*Speech Recognition, Al-qur'an, Android, Rapid Application Development*

Abstract

This research implements *Automatic Speech Recognition (ASR)* technology in the process of assessing Al-Qur'an memorization to overcome the problem of the increasing number of people who cannot read the Al-Qur'an fluently. With ASR, computers can recognize and transcribe human speech accurately, thereby improving the quality and accuracy of pronunciation and providing constructive feedback to learners. An *Android*-based application that uses the ASR service from the *Google Speech API* allows users to recite verses from the Koran and automatically recognize and evaluate what they have memorized. The hope is that the use of ASR technology can improve the ability to memorize the Al-Qur'an more effectively and flexibly without requiring direct teacher assistance. This system was developed using the *Rapid Application Development (RAD)* method with *Blackbox Testing*.

Keywords—*Automatic Speech Recognition, Al-qur'an, Android, Rapid Application Development*

1. PENDAHULUAN

Al-Qur'an adalah firman Allah SWT yang diturunkan kepada Nabi Muhammad SAW untuk menjadi pedoman hidup bagi umat Islam. Bacaan Al-Qur'an adalah bagian paling penting dalam penanaman nilai agama dan moral agar jiwa umat Islam tumbuh diatas fitrah.(Rohimah and Ngulwiyah 2023)

Pendidikan nilai agama dan moral menjadi pondasi dan harus ditanamkan kepada anak usia dini agar tetap tertanam di dalam benak pikiran dan jiwa anak. Pembelajaran Baca Tulis Al-

Qur'an (BTAQ) merupakan pelajaran sebagai proses pembelajaran untuk mengenal dan mempelajari bacaan yang terkandung di dalam ayat-ayat Al-Qur'an. Dalam membaca Al-Qur'an, umat Islam dituntut untuk membaca secara tartil dan lafadz yang ada di Al-Qur'an harus benar-benar terbaca sebagaimana Allah berfirman dalam Q.S Al-Muzamil Ayat 4: "Dan Bacalah Al-Qur'an dengan tartil".(Haikal 2023)

Tantangan internal saat ini adalah meningkatnya angka buta mengaji Al-Qur'an, hal ini disebabkan melemahnya sistem agama pada jalur pendidikan formal, kurang perhatiannya orang tua dalam membimbing anaknya pada pengajaran Al-Qur'an. Dengan adanya tantangan tersebut perlu kiranya dihadirkan sebuah teknologi pembelajaran cara baca Al-Qur'an sebagai *complementary learning* untuk dapat mendukung ketersediaan tata cara baca Al-Qur'an yang dapat digunakan secara mandiri oleh pembelajar. Salah satu teknologi pembelajaran yang dapat digunakan untuk mendeteksi hafalan cara baca Al-Qur'an adalah dengan menggunakan teknologi Pengenalan Ucapan Otomatis (*Automatic Speech Recognition/ASR*).

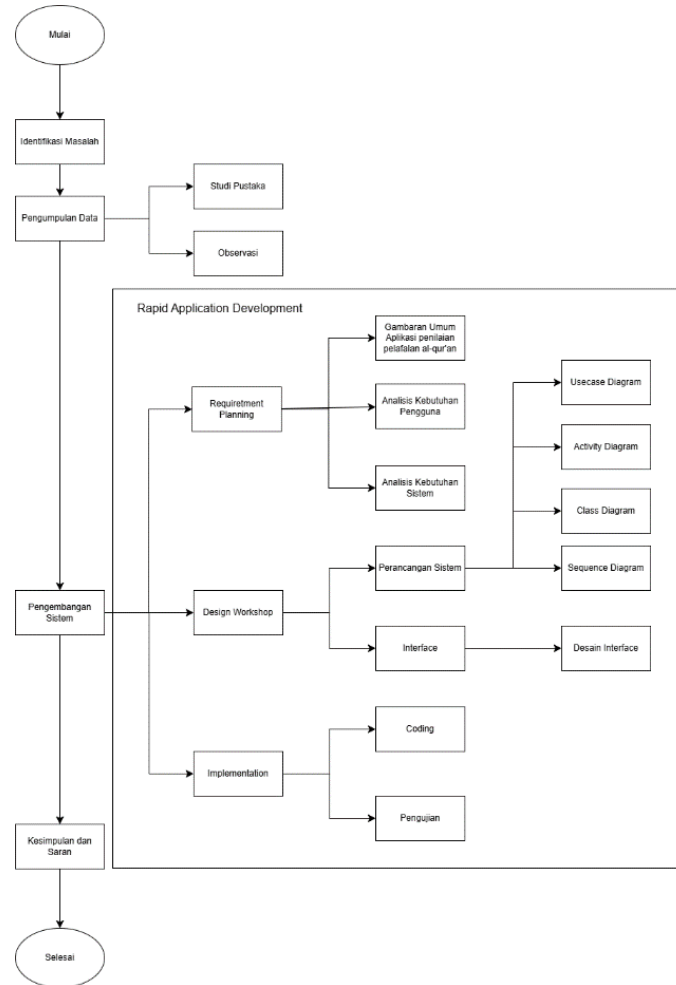
ASR adalah teknologi inovatif yang memungkinkan komputer untuk secara komprehensif mengenali dan mentranskripsikan setiap ucapan manusia dengan tepat menjadi teks tertulis yang akurat. Penerapan ASR dalam pembelajaran pelafan Al-Quran menjadi langkah yang signifikan dalam mengatasi permasalahan tersebut secara efisien.(Zenik et al. 2023)

Selain manfaat yang disebutkan, kemudahan akses yang ditawarkan oleh aplikasi mobile dengan teknologi ASR membuat pembelajaran menjadi lebih fleksibel tanpa perlu meminta bantuan para pengajar secara fisik.

Berdasarkan latar belakang yang sudah dijelaskan maka pada penelitian ini bertujuan untuk mengimplementasikan *Automatic Speech Recognition (ASR)* pada penilaian hafalan Al-Quran berbasis *android* dengan menggunakan *Google Speech API Service*. Aplikasi *android* yang dibangun akan memungkinkan pengguna untuk belajar hafalan Al-Quran dengan mudah. Salah satu fitur utama yang disediakan adalah ASR yang terintegrasi dengan *Google Speech API*. Fitur ini memungkinkan pengguna untuk melafalkan ayat-ayat Al-Quran dan aplikasi akan secara otomatis mengenali serta mengevaluasi hafalan mereka. Diharapkan aplikasi ini dapat membantu pengguna dalam memperbaiki keterampilan hafalan Al-Quran mereka dengan lebih efektif.

2. METODE PENELITIAN

Metode penelitian yang digunakan dalam implementasi *automatic speech recognition* pada penilaian hafalan al-quran dengan metode *muroja'ah* berbasis *android* dapat di Gambarkan melalui *flowchart* pada Gambar 1.



Gambar 1. Metode Penelitian

2.1 Metode Pengumpulan Data

Teknik piengumpulan data yang akan digunakan dalam penelitian ini adalah :

1) Studi Literatur

Studi pustaka dilakukan dengan cara mempelajari penelitian terkait dengan penelitian yang dilakukan penulis. Studi pustaka dilakukan sebelum peneliti melakukan penelitian sehingga dapat mengetahui referensi berdasarkan penelitian sebelumnya dan memperoleh landasan teori mengenai masalah yang diteliti. Pada metode pengumpulan data dengan studi pustaka ini, peneliti membaca, menelusuri, dan mempelajari buku, artikel, jurnal, atau penelitian sejenis lainnya melalui internet yang berkaitan dengan implementasi *automatic speech recognition* pada penilaian hafalan al-qur'an.

2) Observasi

Observasi dilakukan dengan mengamati aktivitas dan perilaku pengunjung di museum Megalodon Surade, serta mengidentifikasi aspek-aspek yang perlu ditingkatkan dalam pengembangan museum *virtual*.

Dalam metode pengumpulan data dengan observasi ini, peneliti mengamati berbagai aspek terkait dengan aplikasi yang telah ada sebelumnya tersebut, termasuk antarmuka pengguna, fitur-fitur yang ada dalam sistem, *navigasi*, interaksi antara pengguna dengan sistem, serta alur kerja yang terjadi di dalamnya. Melalui pengamatan ini, peneliti dapat memperoleh pemahaman yang mendalam tentang kelebihan, kekurangan, serta potensi pengembangan dari aplikasi yang sedang diamati. Hasil dari tahap observasi ini akan

menjadi dasar dalam merancang dan mengembangkan solusi yang tepat bagi aplikasi yang ingin dikembangkan.

2.2 Perencanaan Penelitian

1) Perumusan Masalah

Saat melakukan penelitian untuk merancang aplikasi atau mengembangkan sistem, seringkali dihadapkan pada keterbatasan informasi yang dapat menghambat proses pembuatan sistem tersebut. Oleh karena itu, diperlukan referensi yang dapat membantu dalam proses pengumpulan data untuk pembuatan sistem yang sedang dibahas.

2) Penentuan Tujuan Penelitian

Tujuan yang hendak dicapai dalam penelitian ini adalah untuk mengimplementasikan *automatic speech recognition service* pada sebuah rancang bangun aplikasi penilaian hafalan Al-Qur'an berbasis android guna meningkatkan progres pembelajaran hafalan Al-Qur'an.

2.3 Implementasi Konten

Sumber data Al-qur'an ini berasal dari kombinasi beberapa parameter dari *api.alquran.cloud* dan *quran.kemenag.go.id* dengan menggabungkan datanya ke dalam satu file *JSON* dengan struktur baru untuk penggunaan dan kinerja yang lebih baik.

2.4 Pengujian dan Evaluasi

Pengujian dilakukan untuk mendapatkan output yang sesuai harapan, dan jika terdapat kesalahan agar dapat ditemukan solusi atau jalan keluar untuk memperbaiki kesalahan tersebut.

2.5 Perancangan Aplikasi

Dalam pengembangan aplikasi penilaian hafalan al-Qur'an ini, penulis dengan cermat memilih metode *Rapid Application Development (RAD)* yang terbukti efektif dan banyak digunakan dalam proses pengembangan sistem. Pemilihan metode RAD ini didasari oleh keinginan untuk mengatasi tantangan-tantangan umum yang sering muncul dalam pengembangan sistem, seperti pemenuhan tenggat waktu yang ketat dan peningkatan efisiensi pengembangan. Pendekatan RAD juga sangat mengutamakan pembuatan aplikasi dengan memperhatikan kebutuhan pengguna sistem, guna mencapai solusi yang sesuai dengan permasalahan yang dihadapi. (Parlika et al. n.d.) Beberapa tahapan-tahapan pengembangan sistem dengan pendekatan RAD dapat dijelaskan sebagai berikut:

1) *Requirment Planning* (Perencanaan)

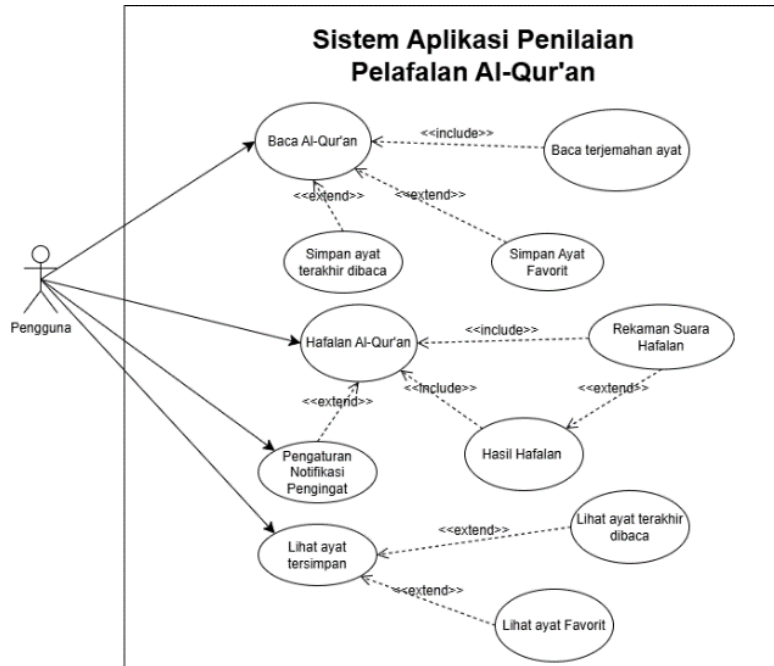
Memperoleh pemahaman menyeluruh tentang *automatic speech recognition* melalui studi pustaka dan observasi. Studi pustaka dilakukan dengan mencari sumber daya yang terpercaya yang terkait dengan *automatic speech recognition*. Sementara observasi dilakukan dengan menganalisis aplikasi yang sedang berjalan atau aplikasi yang telah ada sebelumnya.

2) *Design Workshop (Workshop Desain)*

Pada tahap ini, penulis membuat perancangan aplikasi sistem penilaian hafalan al-qur'an. Dalam hal ini, penulis membuat diagram alur kerja sistem dan melakukan perancangan menggunakan alat pemodelan *Unified Modeling Language (UML)*, dengan tahapan sebagai berikut:

a. Membuat *Use Case Diagram*

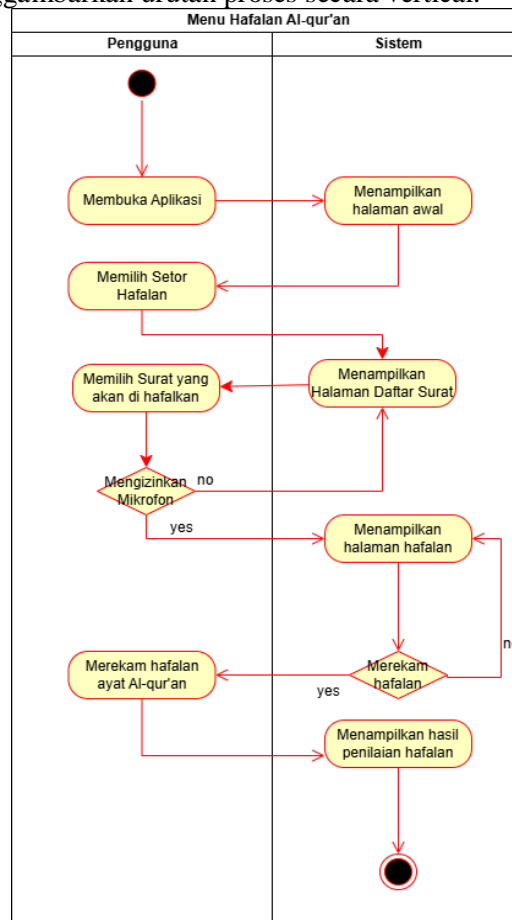
Use Case diagram menjelaskan bagaimana aktor, seperti pengguna aplikasi dan perancang aplikasi, berinteraksi dengan sistem aplikasi.



Gambar 2. Use case diagram pengguna

b. Membuat Activity Diagram

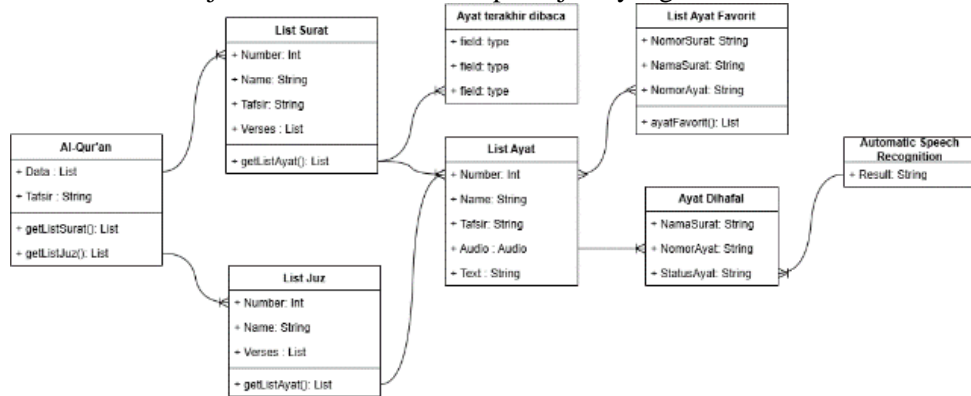
Activity Diagram adalah representasi visual dari proses-proses yang terjadi dalam suatu sistem. Diagram ini menggambarkan urutan proses secara vertical.



Gambar 3. Activity Diagram Menu Hafalan

c. Membuat Class Diagram

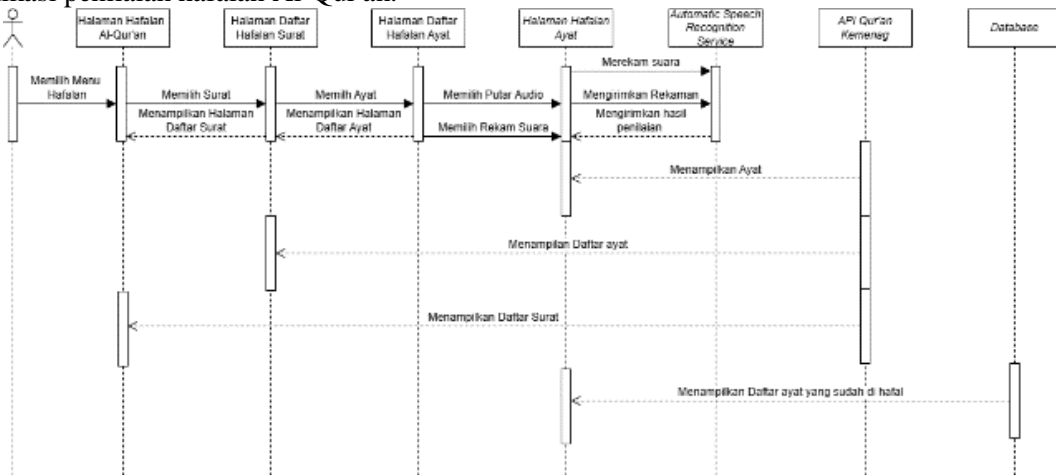
Class Diagram adalah tipe diagram struktur pada UML yang secara jelas menggambarkan struktur serta detail dari setiap kelas, atribut, metode, dan hubungan antar objek-objeknya. Hal ini memperlihatkan interaksi antar class dalam aplikasi penilaian hafalan al-Qur'an serta bagaimana sistem bekerja bersama untuk mencapai tujuan yang sama.



Gambar 4. Class Diagram aplikasi

d. Membuat Class Diagram

Memberikan penjelasan secara terperinci mengenai urutan proses yang dijalankan oleh aplikasi penilaian hafalan Al-Qur'an.



Gambar 5. Sequence Diagram Hafalan Al-qur'an

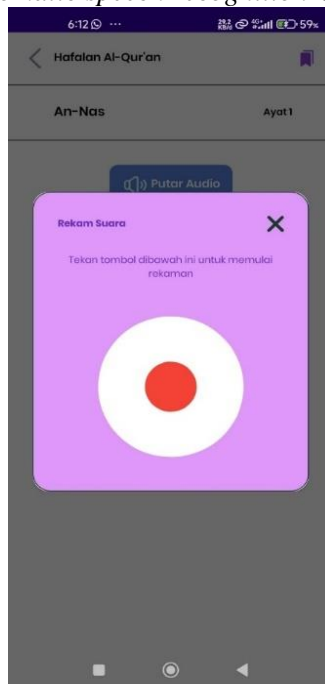
3. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Metode Muroja'ah

3.1.1 Implementasi Automatic Speech Recognition

Dalam penelitian ini, implementasi *automatic speech recognition* yang dilakukan pada saat perancangan aplikasi dilakukan dengan menggunakan *service Google Speech*. Implementasi *Google Speech API* pada pembuatan aplikasi penilaian pelafalan al-qur'an dapat dilakukan dengan lebih sederhana karena perancangan aplikasi menggunakan bahasa pemrograman Kotlin. Kotlin dapat mengakses *library* milik *Java* termasuk *Apache*. (Mahesa Putera Darajat et al. 2023) Perekaman suara dapat dilakukan dengan memanggil fungsi "*startRecording()*" dan memanggil fungsi "*stopRecording()*" untuk menghentikan rekaman. Setelah rekaman berhenti maka *Google Speech API* akan memberikan hasil *speech to text* secara

otomatis yang bisa digunakan untuk melakukan perhitungan nilai kesamaan dengan lafadz ayat Al-Qur'an.(Hukkeri, Goudar, and Scholar 2022) Berikut adalah tampilan aplikasi untuk merekam suara menggunakan *automatic speech recognition Google Speech API service*.



Gambar 6. Hasil Implementasi *Automatic Speech Recognition*

3.1.2 Implementasi Jaro-Winkler Similarity

Langkah pertama untuk implementasi Jaro-Winkler Similarity dilakukan dengan konfigurasi pada file "*build.gradle(:app)*". Selanjutnya, panggil library "*JaroWinklerSimilarity()*" melalui sebuah variabel dan *output* dari nilai kesamaan antar dua *text*. Variabel "*similarity*" dapat digunakan untuk mengatur *respon* aplikasi terhadap nilai yang dihasilkan dari kesamaan teks hasil deteksi dengan lafadz ayat Al-Qur'an. (Sahfitri et al. n.d.) Hasil implementasi nilai kesamaan dapat dilihat pada gambar berikut ini.



Gambar 7. Hasil Implementasi *Jaro-Winkler Similarity*

Dalam studi kasus ini, penulis membandingkan teks asli "قُلْ أَغُوذُ بِرَبِّ النَّاسِ" dengan hasil transkripsi dari *Google Speech-to-Text API* menggunakan algoritma *Jaro-Winkler*. Panjang teks asli adalah 23 karakter, sementara hasil transkripsi tanpa tanda baca memiliki panjang 17 karakter. Dengan menghitung kesamaan karakter antara kedua teks, ditemukan 15 karakter yang cocok. Jumlah transposisi, yaitu karakter yang perlu ditukar posisinya, adalah 0.

String 1 (S1): "قُلْ أَغُوذُ بِرَبِّ النَّاسِ"

String 2 (S2): "قل اعوذ برب الناس"

Rumus Jaro Distance adalah:

$$dj = \frac{1}{3} \left(\frac{c}{|s_1|} + \frac{c}{|s_2|} + \frac{c-t}{c} \right)$$

Dengan $c = 15$, $t = 0$:

$$dj = \frac{1}{3} \left(\frac{15}{23} + \frac{15}{17} + \frac{15-0}{15} \right)$$

$$dj = \frac{1}{3} (0.65217 + 0.88235 + 1)$$

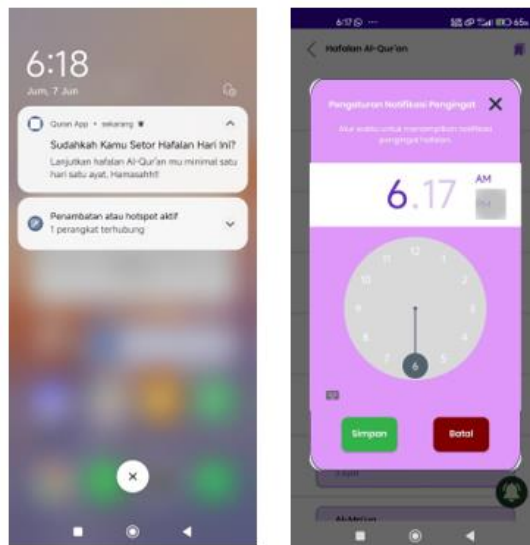
$$dj = \frac{1}{3} (2.53452)$$

$$dj = 0.84484$$

Nilai Jaro-Winkler Distance antara "قُلْ أَغُوذُ بِرَبِّ النَّاسِ" dan hasil dari *Google Speech-to-Text API* adalah sekitar 0.84484, nilai ini menunjukkan tingkat kemiripan yang cukup tinggi antara kedua teks, dengan nilai kemiripan 84%

3.1.3 Implementasi Notifikasi Pengingat

Implementasi fitur notifikasi pengingat dilakukan dengan menggunakan *library SharedPreference*, *class NotificationManager*, *class AlarmManager* dari *Kotlin*. Pengaturan notifikasi pengingat membutuhkan perizinan "*Notification*" terhadap aplikasi. Notifikasi pengingat dapat diatur dengan memanggil fungsi "*requestNotificationPermission()*" sehingga aplikasi akan meminta perizinan kepada pengguna dan pengguna dapat mengatur waktu notifikasi setelah perizinan aplikasi diterima. Berikut adalah hasil implementasi dari notifikasi pengingat.



Gambar 8. Hasil Implementasi Fitur Notifikasi Pengingat

3.1.4 Hasil Implementasi REST API

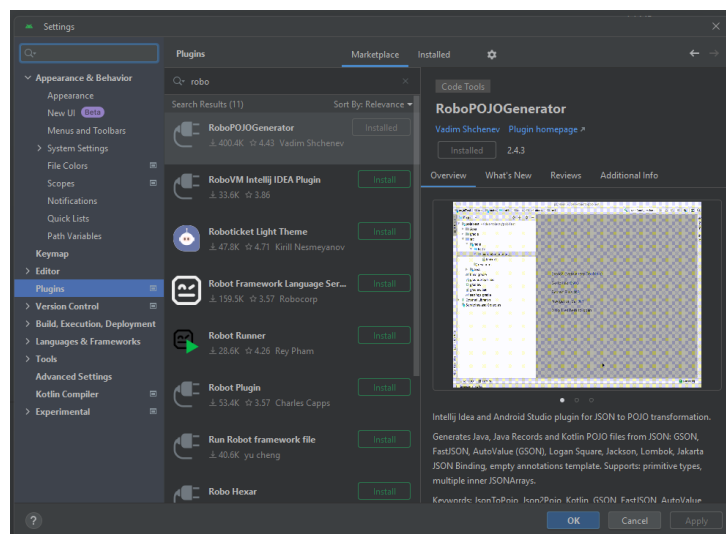
Implementasi *REST API* dilakukan dengan menggunakan *library GSON* dan *Retrofit*. *Library GSON* berfungsi untuk mengonversi *JSON* sebagai *response* dari *request* suatu *API*, sedangkan *Retrofit* berfungsi untuk menyederhanakan proses komunikasi sistem aplikasi terhadap suatu *REST API*.

1) Konfigurasi API Quran Kemenag

Sumber data *API* ini berasal dari kombinasi beberapa parameter dari *api.alquran.cloud* dan *quran.kemenag.go.id* dengan menggabungkan datanya ke dalam satu file *JSON* dengan struktur baru untuk penggunaan dan kinerja yang lebih baik. (Puspitasari 2022) Selain itu, *API* ini menggunakan teknik *query* terindeks dengan waktu akses $O(1)$ yang sangat mempengaruhi kinerja. Untuk melakukan konfigurasi *API Quran Kemenag*, langkah pertama adalah masuk ke Repositori *Github* dan melakukan *fork repository* dari <https://github.com/gadingnst/quran-api>. Selanjutnya, masuk ke *platform Vercel*, membuat proyek di *Vercel* dengan cara mengimpor dari *Github* dan memilih repositori *Github* yang berisi data Quran Kemenag. Terakhir, dapatkan *URL API Quran Kemenag* dari *URL* yang terdapat pada “*Domain*” setelah proses *deploy* selesai.

2) Konfigurasi Library GSON dan Retrofit

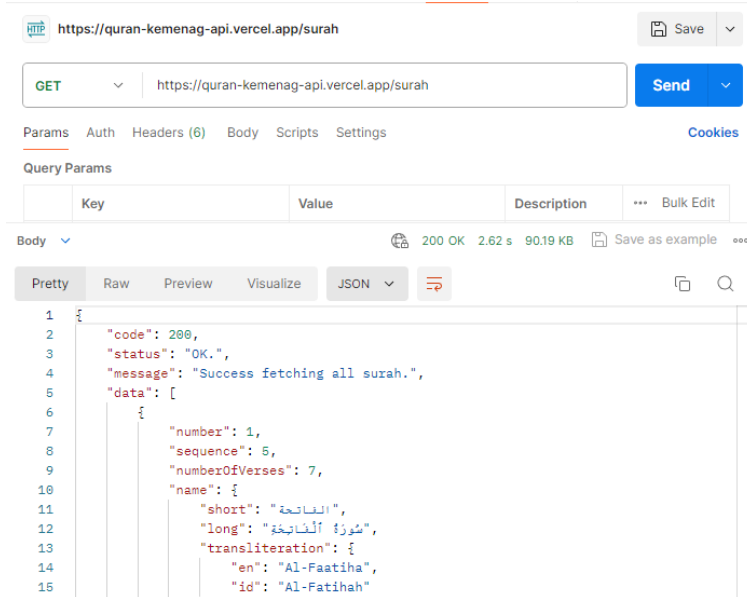
Untuk melakukan implementasi kedua *library* tersebut, perlu menambahkan kode konfigurasi pada file “*build.gradle(:app)*”. Lalu untuk mempermudah penggunaan *GSON*, disarankan untuk menggunakan *plugin* yang telah disediakan oleh *Android Studio*. Caranya adalah dengan masuk ke menu “*Settings*”, kemudian pilih opsi “*Plugins*” dan cari *plugin* bernama “*RoboPOJOGenerator*” untuk diinstal. Selanjutnya adalah penulisan kode untuk implementasi *REST API* menggunakan kedua *library* tersebut.



Gambar 9. Instalasi *Plugin RoboPOJOGenerator*

Membuat file “*QuranApiService*” sebagai sebuah *interface* yang akan digunakan untuk memproses data Al-Qur'an melalui *API*. File ini akan berperan sebagai tempat definisi dari metode-metode yang dipakai untuk berinteraksi dengan *API* Al-Qur'an, seperti mengambil ayat atau mendapatkan informasi lain yang berkaitan dengan Al-Qur'an.

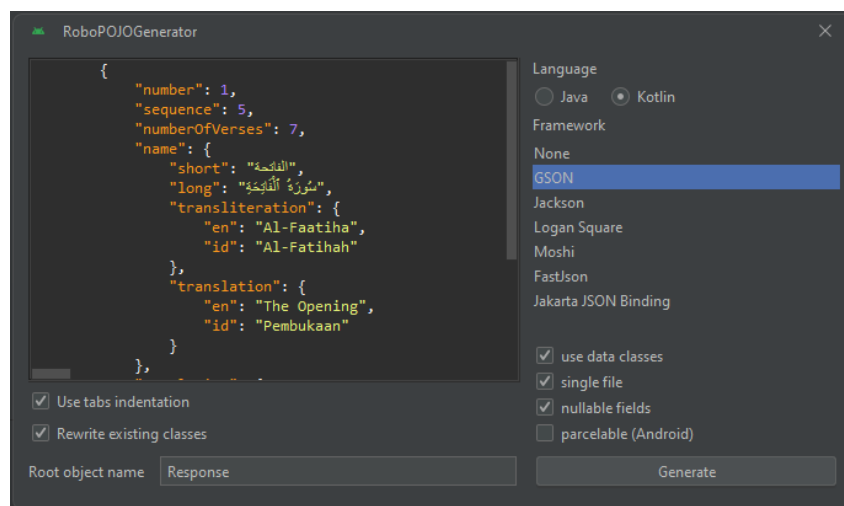
Setelah file “*QuranApiService*” selesai dibuat, dilakukan pengujian *request API* melalui *Postman* guna mencoba implementasi *interface* yang telah disiapkan. Dengan menggunakan *Postman*, akan dilakukan pengiriman permintaan ke *server API* Al-Qur'an dan memverifikasi kesesuaian data yang diterima. Langkah ini bertujuan untuk memastikan bahwa *interface* “*QuranApiService*” berfungsi secara optimal dalam mengakses data Al-Qur'an melalui *API*. (Hamzah, Harahap, and Nasir n.d.)



Gambar 10. Response API di Postman

Setelah file "*QuranApiService*" selesai dibuat, dilakukan pengetesan *request API* melalui *Postman* guna mencoba implementasi *interface* yang telah disiapkan. Dengan menggunakan *Postman*, akan dilakukan pengiriman permintaan ke *server API* Al-Qur'an dan memverifikasi kesesuaian data yang diterima. Langkah ini bertujuan untuk memastikan bahwa *interface* "*QuranApiService*" berfungsi secara optimal dalam mengakses data Al-Qur'an melalui *API*.

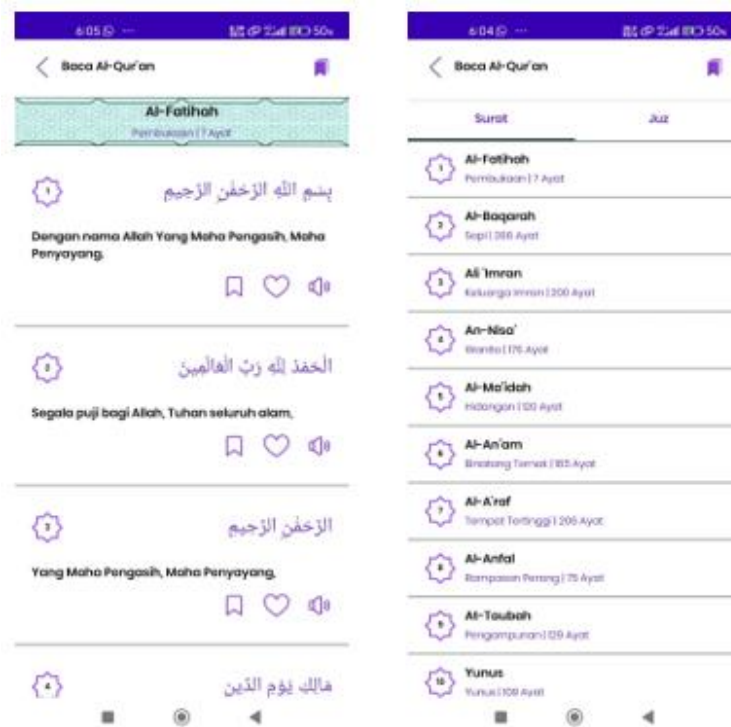
Untuk memudahkan konversi respons API dari *JSON* ke *GSON*, keberadaan *RoboPOJOGenerator* sangat bermanfaat. Fitur ini dapat diakses dengan menggunakan opsi "*Generate POJO from JSON*". Dengan fitur ini, proses pembuatan kelas-kelas POJO (Plain Old Java Object) sesuai dengan struktur data respons API akan menjadi lebih efisien dan membutuhkan upaya yang lebih sedikit. Hal ini dapat membantu dalam mengintegrasikan respons API ke dalam kode program, serta menyederhanakan proses pengelolaan data saat mengembangkan aplikasi.



Gambar 11. Konfigurasi RoboPOJOGenerator

Implementasi *REST API* untuk menampilkan list ayat Al- Qur'an dari *API* Qur'an Kemenag dapat dilakukan dengan memanggil fungsi "*getListAyat(nomorSurat: String)*" pada file "*activity*". Setelah *REST API* dari Qur'an Kemenag berhasil di implementasi, maka aplikasi telah

dapat menampilkan daftar surat dan daftar ayat Al-Qur'an seperti yang dapat dilihat dari tangkapan layar aplikasi dibawah ini.



Gambar 12. Hasil Implementasi REST API

3.1.5 Pengujian Black-box Testing

Pada tahap ini aplikasi diuji menggunakan pendekatan *blackbox testing*. Pengujian *Black Box* adalah metode pengujian yang memverifikasi keluaran aplikasi berdasarkan masukan yang diberikan (data uji) untuk memastikan bahwa aplikasi telah memenuhi persyaratan fungsional. (Arkan et al. 2023) adalah hasil pengujian:

Tabel 1. Tabel Uji Coba Fitur

No	Aktivitas Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1	Membuka aplikasi	Sistem menampilkan halaman beranda	Valid
2	Navigasi halaman al-qur'an	Sistem menampilkan halaman al-qur'an	Valid
3	Mengatur notifikasi pengingat hafalan	Muncul notifikasi sesuai jam yang ditentukan	Valid
4	Menyimpan ayat terakhir dibaca	Muncul pada halaman penanda dan halaman beranda	Valid
5	Melakukan rekaman hafalan al-qur'an	Muncul tombol rekaman berwarna merah untuk mematikan dan berwarna hijau jika sedang merekam	Valid
6	Melakukan penilaian hafalan setelah selesai merekam	Muncul hasil penilaian	Valid

Berdasarkan tabel 1 uji coba fitur menggunakan Blackbox Testing, menunjukkan bahwa hasil dari aktivitas pengujian yang dilakukan adalah *valid* dan tidak terdapat error.

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan, aplikasi yang dirancang dengan bahasa pemrograman *Kotlin* telah berhasil mengimplementasikan *automatic speech recognition* dengan metode *Muroja'ah* pada beberapa fitur utama. Fitur pertama adalah fitur rekam suara dengan layanan *automatic speech recognition* yang memungkinkan pengguna untuk merekam suara dan mengkonversinya menjadi teks secara otomatis. Fitur ini telah diimplementasikan dengan menggunakan teknologi *speech recognition* terkini yang memastikan akurasi dan kecepatan dalam pengenalan suara.

Selain itu, aplikasi juga telah berhasil mengimplementasikan fitur hasil deteksi dengan algoritma *Jaro-Winkler Similarity*. Algoritma ini memungkinkan pengguna untuk membandingkan teks hasil *speech recognition* dengan teks referensi untuk menentukan tingkat kesamaannya. Hal ini memungkinkan pengguna untuk mendapatkan hasil deteksi yang akurat dan dapat diandalkan.

Terakhir, aplikasi juga telah berhasil mengimplementasikan fitur notifikasi pengingat dengan menggunakan *AlarmManager*. Fitur ini memungkinkan pengguna untuk mengatur pengingat berdasarkan teks hasil *speech recognition* yang telah diinputkan sebelumnya. Dengan demikian, pengguna dapat dengan mudah mengatur pengingat berdasarkan hasil rekaman suara tanpa perlu mengetik secara manual.

Hasil dari pengujian *blackbox* menunjukkan bahwa aplikasi telah berhasil mengimplementasikan fitur-fitur tersebut dengan baik dan memberikan hasil yang sesuai dengan harapan. Dengan demikian, dapat disimpulkan bahwa penggunaan bahasa pemrograman *Kotlin* dalam merancang aplikasi penilaian hafalan al-qur'an dengan metode *muroja'ah* telah memberikan hasil yang memuaskan dan dapat diandalkan.

DAFTAR PUSTAKA

- [1] Arkan, Mohammad, Ridhwan Razan, Issa Arwani, and Hariz Farisi. 2023. 7 Pengembangan Aplikasi Monitoring Dan Evaluasi Kinerja Siswa Magang Dari Indonesia Learning Center Berbasis Android. <http://j-ptiik.ub.ac.id>.
- [2] Haikal, M Fikri. 2023. "Manajemen Pendidikan Islam Perspektif Al-Qur'an Dan Hadist." *Journal on Education* 05(04).
- [3] Hamzah, Donny, P Harahap, and Nursina Nasir. BACKEND CRUD APLIKASI WEBSITE E-COMMERCE.
- [4] Hukkeri, Geeta S, R H Goudar, and Research Scholar. 2022. 13 IJACSA) International Journal of Advanced Computer Science and Applications Erratic Navigation in Lecture Videos Using Hybrid Text Based Index Point Generation. www.ijacsa.thesai.org.
- [5] Mahesa Putera Darajat, Bintang, Adhi Rizal, Budi Arief Dermawan, Singaperbangsa Karawang Jl HSRonggo Waluyo, Telukjambe Timur, and Jawa Barat. 2023. 7 Jurnal Mahasiswa Teknik Informatika PENGEMBANGAN APLIKASI TEKNIK PEMBELAJARAN BELADIRI BERBASIS ANDROID MENGGUNAKAN METODE RESEARCH AND DEVELOPMENT.
- [6] Parluka, Rizky, Muhammad Afifudin, Ilham Akbar Pradana, Yohanes Dimas, Wisnu Wiratama, and Mohammad Nur Holis. "Positif: Jurnal Sistem Dan Teknologi Informasi

STUDI LITERATUR EFISIENSI MODEL RAPID APPLICATION DEVELOPMENT DALAM PENGEMBANGAN PERANGKAT LUNAK (2014-2022).” <http://mcastud.com/student-project-development-go/>.

- [7] Puspitasari, Debi Ayu. 2022. “Kebijakan Pentashihan Aplikasi Al-Qur’an Digital Di Indonesia: Studi Perkembangan Aplikasi ‘Al-Quran Kementerian Agama’ Dan Permasalahannya.” *J-PAI: Jurnal Pendidikan Agama Islam* 8(1). doi:10.18860/jpai.v8i1.13425.
- [8] Rohimah, Rt. Bai, and Istinganatul Ngulwiyah. 2023. “Tren Metode Pembelajaran Al-Qur’an Di Indonesia Tahun 2019-2023: Sebuah Systematic Review.” *Jurnal Pendidikan Abad Ke-21* 1(2): 85–94. doi:10.53889/jpak.v1i2.329.
- [9] Sahfitri, Vivi, Ibnu Batutah Zarizal, Universitas Bina Darma, Jalan Jenderal Ahmad Yani No, and Palembang Sur-el. “APPROXIMATE STRING MATCHING UNTUK PENCARIAN KATA DALAM KAMUS BAHASA INDONESIA MENGGUNAKAN ALGORITMA JARO WINKLER.” *Jurnal Ilmiah MATRIK* 24(3): 2022.
- [10] Zenik, Andre, Catherine Olivia Sereati, Kumala Indriati, Linda Wijayanti, and Prodi Teknik Elektro. 2023. 16 Analisis Audio Capture Untuk Pengumpulan Data Pada Smart Speaker.