

Implementasi Aplikasi Voice Assistant Android Rahayu Berbasis Speech Recognition dan Integrasi API YouTube/Wikipedia

Gifari Dhaffa Prawira Sianturi¹

¹ Fakultas Matematika dan Ilmu Pengetahuan Alam, Ilmu Komputer, Universitas Negeri Medan, Medan, Indonesia

Email: ¹gifaridhaffa.4213550031@mhs.unimed.ac.id

Abstrak—Abstrak Perkembangan teknologi informasi yang pesat mendorong terciptanya interaksi manusia dan komputer yang semakin natural melalui teknologi asisten suara. Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi Rahayu Voice Assistant berbasis Android yang dapat memahami perintah suara pengguna serta memberikan respons secara real-time. Aplikasi ini menggunakan SpeechRecognizer API untuk mengubah suara menjadi teks, TextToSpeech API untuk menghasilkan suara balasan, dan integrasi API YouTube serta Wikipedia untuk menyajikan informasi sesuai konteks perintah. Metode penelitian meliputi tahap perancangan sistem, pengembangan antarmuka chat interaktif menggunakan RecyclerView, serta pengujian fungsi utama dengan metode Black Box Testing. Hasil pengujian menunjukkan bahwa aplikasi mampu mengenali perintah dasar dengan tingkat keberhasilan tinggi dalam kondisi suara yang jelas dan stabil. Aplikasi ini terbukti memberikan interaksi dua arah yang alami serta dapat berjalan baik pada perangkat dengan spesifikasi menengah. Penelitian ini diharapkan menjadi contoh implementasi praktis dalam penggabungan komponen speech recognition, natural language processing sederhana, dan text-to-speech pada pengembangan aplikasi mobile.

Kata Kunci: Voice Assistant; Speech Recognition; Text-To-Speech; Android; API Youtube; API Wikipedia

Abstract—Abstract The rapid development of information technology has fostered more natural human-computer interaction through voice assistant technology. This study aims to design and implement Rahayu – Voice Assistant, an Android-based application capable of understanding user voice commands and providing real-time responses. The application employs the SpeechRecognizer API to convert speech into text, the TextToSpeech API to generate voice feedback, and integrates YouTube and Wikipedia APIs to deliver contextual information. The research method includes system design, development of a chat-based interface using RecyclerView, and functionality testing through Black Box Testing. The results show that the application can recognize basic commands with high success rates under clear and stable audio conditions. The system provides a natural two-way interaction and performs well on mid-range Android devices. This research serves as a practical implementation example of integrating speech recognition, simple natural language processing, and text-to-speech in mobile application development.

Keywords: Voice Assistant; Speech Recognition; Text-To-Speech; Android; Youtube API; Wikipedia API

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat pada era digital saat ini telah memberikan dampak besar terhadap cara manusia berinteraksi dengan perangkat elektronik. Salah satu bentuk inovasi dalam bidang *Human-Computer Interaction (HCI)* adalah teknologi **pengenalan suara (speech recognition)** yang memungkinkan pengguna memberikan perintah suara untuk berinteraksi langsung dengan sistem tanpa harus menggunakan antarmuka fisik [1]. Teknologi ini menjadi dasar dalam pengembangan berbagai aplikasi modern, terutama pada platform mobile seperti Android, yang kini telah menyediakan API bawaan untuk *speech recognition* dan *text-to-speech* [2], [3].

Beberapa penelitian sebelumnya menunjukkan kemajuan signifikan dalam penerapan teknologi pengenalan suara berbasis Android. Arisaputra dan Zahra [1] meneliti implementasi *Indonesian Automatic Speech Recognition* menggunakan model **XLSR-53**, yang mampu meningkatkan akurasi pengenalan bahasa Indonesia secara signifikan. Yana Karisma et al. [4] menerapkan teknologi *speech-to-text* pada aplikasi kamus bahasa Sumbawa-Indonesia-Inggris berbasis Android untuk membantu pengguna melakukan penerjemahan suara. Sujjada et al. [5] mengembangkan sistem *automatic speech recognition* untuk penilaian hafalan Al-Qur'an berbasis Android dengan metode *muroja'ah* yang menunjukkan kemampuan pengenalan ucapan religius secara akurat.

Penelitian lain oleh Jaman dan Fergina [6] memanfaatkan teknologi *speech recognition* untuk membantu komunikasi bagi penyandang tunarungu. Deslianti [7] membangun aplikasi konversi *speech-to-text* bahasa Indonesia ke bahasa Bengkulu berbasis *Pocketsphinx*, yang menjadi salah satu implementasi ringan pengenalan suara di perangkat seluler. Sementara itu, Nasri [8] mengembangkan sistem konversi ucapan bahasa Indonesia ke bahasa isyarat Indonesia (*SIBI*), membuktikan bahwa *speech processing* dapat diterapkan untuk tujuan sosial.

Dalam bidang pembelajaran bahasa dan komunikasi, Evanti dan Willay [9] menerapkan *speech recognition* pada aplikasi cek pengucapan bahasa Inggris berbasis Android, yang memudahkan pengguna dalam meningkatkan kemampuan pelafalan. Penelitian Priyo Perdana Adati et al. [10] menggunakan metode *Rapid Application Development (RAD)* dalam membangun aplikasi pengenalan bahasa Mongondow berbasis *speech recognition*, yang menunjukkan efektivitas pendekatan iteratif dalam proyek Android berskala kecil. Yuliadi et al. [11] juga mengimplementasikan aplikasi kamus bahasa Sasak berbasis Android menggunakan *speech recognition* untuk mempermudah pencarian kosakata lokal.

Selain pengenalan suara, pengubahan teks menjadi suara atau *text-to-speech (TTS)* juga menjadi komponen penting dalam sistem asisten suara. Arrizqi et al. [12] menerapkan *Google Text-to-Speech* pada aplikasi pendeteksi

uang berbasis Android untuk membantu pengguna dengan gangguan penglihatan. Kurniadi et al. [13] menggunakan *Firestore Machine Learning Kit* dalam perancangan sistem *text-to-speech* bahasa Indonesia agar suara yang dihasilkan terdengar alami. Kusmayadi et al. [14] mengembangkan aplikasi *voice assistant* pada *smartwatch* menggunakan *Open Artificial Intelligence* untuk mengoptimalkan interaksi pengguna melalui perintah suara. Penelitian Park et al. [15] juga menunjukkan bagaimana sistem *on-device speech recognition* dapat dioptimalkan agar memiliki latensi rendah dan konsumsi daya kecil pada perangkat mobile.

Sementara itu, Hugeng dan Hansel [16] meneliti penerapan *speech recognition* untuk kamus geografi bahasa Indonesia berbasis Android dan menekankan pentingnya penggunaan *SpeechRecognizer API* untuk proses pengenalan suara lokal. Penelitian serupa oleh M. Hidayat dan Amrullah [17] mengembangkan sistem kamus bahasa Indonesia–Sumbawa dengan teknologi pengenalan suara untuk mempermudah pengguna mencari istilah lokal tanpa mengetik. Berbagai penelitian tersebut menunjukkan bahwa teknologi pengenalan suara dan *text-to-speech* memiliki potensi besar untuk diterapkan dalam berbagai bidang, mulai dari pendidikan hingga layanan publik.

Namun demikian, sebagian besar penelitian yang ada masih berfokus pada penerapan tunggal, seperti konversi suara ke teks (*speech-to-text*) atau sebaliknya (*text-to-speech*), tanpa menggabungkan keduanya dalam satu sistem interaktif. Selain itu, banyak sistem yang memerlukan sumber daya tinggi atau model *cloud-based* untuk dapat berjalan optimal, sehingga sulit diimplementasikan dalam konteks akademik mahasiswa. Oleh karena itu, penelitian ini bertujuan untuk **mengimplementasikan aplikasi asisten suara sederhana berbasis Android bernama Rahayu**, yang mengintegrasikan *SpeechRecognizer API*, *TextToSpeech API*, dan pemanggilan *Application Programming Interface (API)* eksternal seperti YouTube dan Wikipedia.

Aplikasi ini dibangun menggunakan bahasa pemrograman **Java** di dalam lingkungan **Android Studio**, dengan rancangan antarmuka percakapan (*chat interface*) yang interaktif. Sistem ini bekerja sepenuhnya di perangkat (*on-device*), memanfaatkan pustaka bawaan Android tanpa memerlukan server tambahan, sesuai rekomendasi Google Developers [2], [3]. Selain itu, penerapan integrasi API dilakukan berdasarkan panduan *GeeksforGeeks* [18] dan *MediaWiki Developer Documentation* [19] untuk memastikan sistem dapat menampilkan hasil pencarian secara real-time.

Dengan adanya aplikasi ini, mahasiswa dapat memahami secara praktis konsep integrasi antara *speech recognition*, *API handling*, dan *text-to-speech* dalam satu aplikasi Android. Penelitian ini tidak dimaksudkan untuk menyaingi sistem komersial seperti Google Assistant atau Siri, melainkan sebagai **studi implementatif akademik** untuk meningkatkan kompetensi mahasiswa dalam pengembangan aplikasi berbasis suara yang ringan, fungsional, dan mudah dikembangkan.

Secara umum, penelitian ini memiliki tujuan sebagai berikut:

1. Mengimplementasikan teknologi *speech recognition* dan *text-to-speech* menggunakan API bawaan Android.
2. Mengintegrasikan API publik seperti YouTube dan Wikipedia untuk menampilkan informasi secara real-time.
3. Membangun antarmuka *chat-based* yang interaktif agar pengalaman pengguna lebih alami.
4. Melakukan pengujian sistem menggunakan pendekatan *black box testing* untuk menilai keandalan fungsionalitas aplikasi.

Dengan demikian, hasil dari penelitian ini diharapkan dapat menjadi dasar pengembangan lanjutan pada bidang interaksi manusia dan komputer (*Human-Computer Interaction*) serta menjadi referensi implementatif dalam pembelajaran pemrograman Android di lingkungan akademik [20], [18].

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan menggunakan pendekatan **metode pengembangan perangkat lunak secara bertahap (incremental development)**. Setiap tahap berfokus pada perancangan, implementasi, dan pengujian fungsi utama aplikasi. Tahapan penelitian dapat dijelaskan sebagai berikut:

1. **Analisis Kebutuhan**
Pada tahap ini dilakukan identifikasi kebutuhan pengguna dan sistem, meliputi kemampuan aplikasi dalam menerima perintah suara, menampilkan hasil pencarian, serta merespons pengguna melalui suara. Analisis juga mencakup penentuan API yang digunakan, yaitu YouTube dan Wikipedia.
2. **Perancangan Sistem**
Perancangan dilakukan untuk menentukan struktur sistem, termasuk rancangan *use case*, *flowchart*, dan antarmuka pengguna (*user interface*). Komponen utama seperti *SpeechRecognizer*, *TextToSpeech*, dan *RecyclerView* dirancang agar dapat berinteraksi secara terpadu.
3. **Implementasi**
Implementasi dilakukan dengan menggunakan bahasa pemrograman **Java** di dalam **Android Studio**.

Tahap ini meliputi pembuatan kode pada file MainActivity.java, activity_main.xml, serta konfigurasi izin (*permission*) di AndroidManifest.xml.

4. Integrasi API

Aplikasi dihubungkan dengan API publik **YouTube** dan **Wikipedia** untuk mengambil informasi atau hasil pencarian yang relevan berdasarkan perintah suara pengguna.

5. Pengujian dan Evaluasi

Pengujian dilakukan menggunakan metode **Black Box Testing**, yang berfokus pada pengujian fungsionalitas aplikasi tanpa melihat struktur internal kode. Evaluasi dilakukan untuk memastikan sistem mampu mengenali suara, menampilkan hasil yang sesuai, dan memberikan respons suara secara akurat.

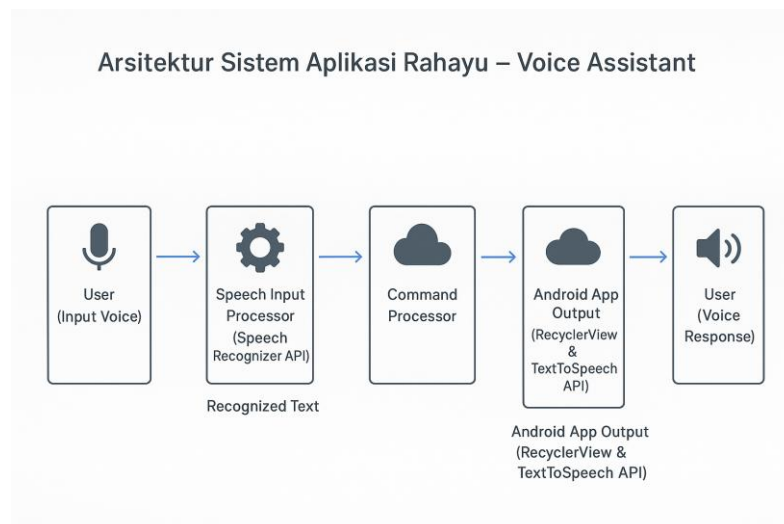
Tahapan-tahapan tersebut divisualisasikan pada **Gambar 1** sebagai alur umum proses penelitian.



Gambar 1. Tahapan Penelitian Implementasi Aplikasi Voice Assistant Rahayu

2.2 Arsitektur Sistem

Arsitektur sistem aplikasi **Rahayu – Voice Assistant** terdiri dari empat komponen utama yang saling berinteraksi untuk menghasilkan sistem asisten suara berbasis Android. Struktur arsitektur sistem ditunjukkan pada **Gambar 2** berikut:



Gambar 2. Arsitektur Sistem Aplikasi Voice Assistant Rahayu

1. Input Suara (Speech Input)

Pengguna menekan tombol mikrofon pada antarmuka aplikasi untuk memberikan perintah suara. Input ini kemudian dikirim ke modul pengenalan suara.

2. Speech Recognition Module

Modul ini memanfaatkan **SpeechRecognizer API** bawaan Android untuk mengubah suara pengguna menjadi teks. Proses ini bekerja secara lokal dan real-time tanpa memerlukan model tambahan.

3. Command Processor & API Handler

Setelah perintah diubah menjadi teks, sistem akan mencocokkannya dengan daftar perintah yang

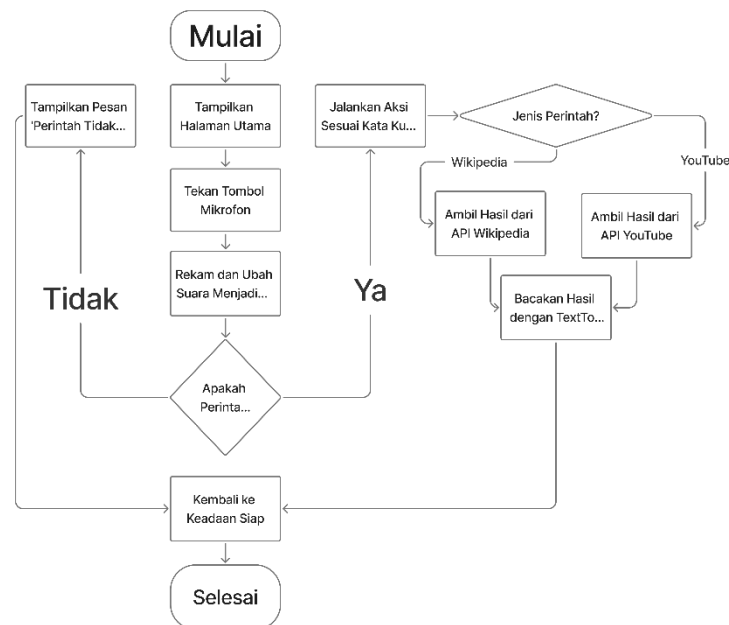
tersedia. Jika perintah mengandung kata kunci “YouTube” atau “Wikipedia”, modul akan mengirimkan *request* ke API terkait untuk mengambil hasil pencarian.

4. Output Module (Text-to-Speech dan Display)

Hasil pencarian ditampilkan di area *RecyclerView* dalam bentuk percakapan dan dibacakan kembali oleh sistem menggunakan **TextToSpeech API**. Dengan demikian, pengguna dapat berinteraksi seolah sedang berbicara dengan asisten virtual.

2.3 Alur Sistem (Flowchart)

Untuk menjelaskan proses kerja sistem secara keseluruhan, dibuat diagram alir seperti yang ditunjukkan pada **Gambar 3**.



Gambar 3. Flowchart Sistem Aplikasi Voice Assistant Rahayu

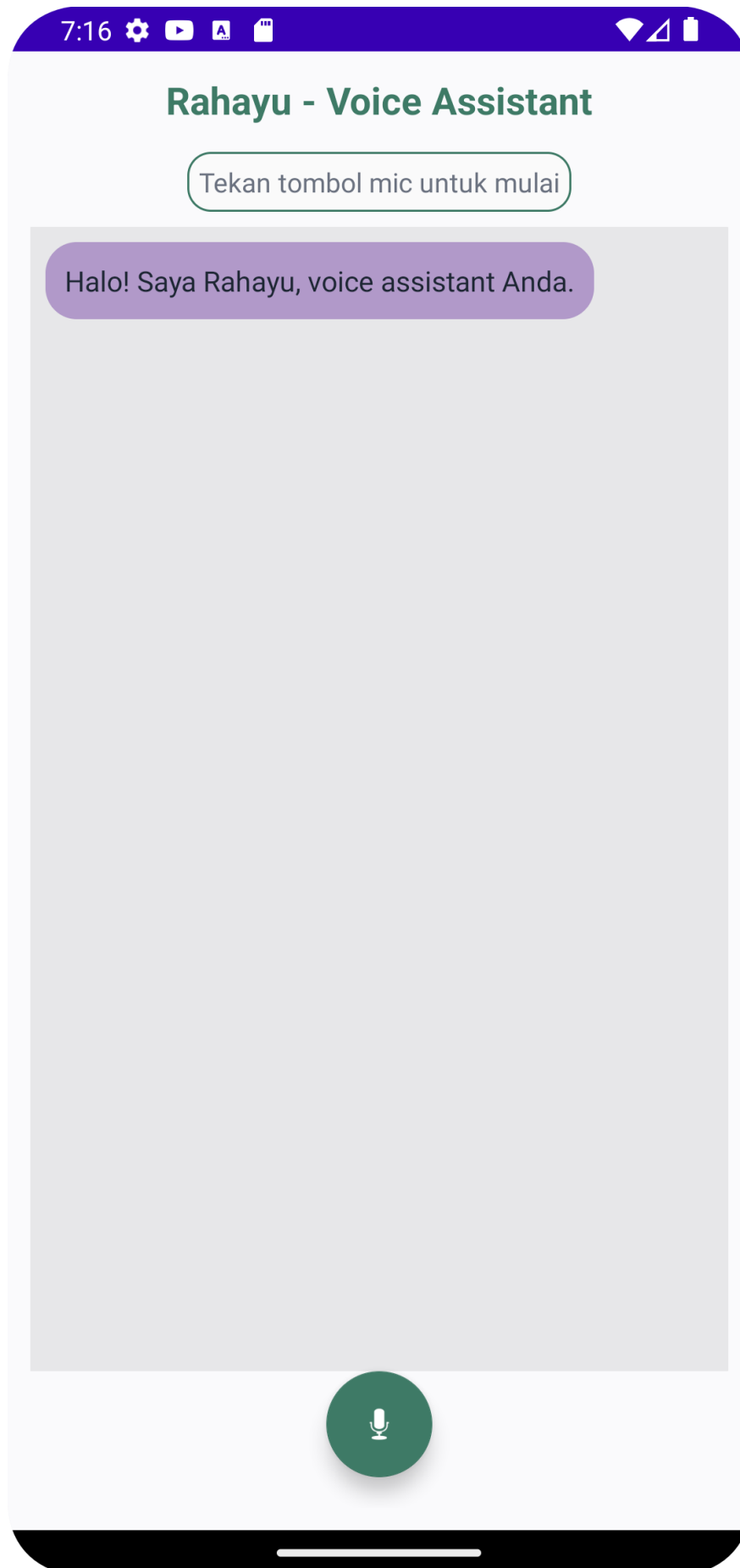
Deskripsi alur sistem:

1. Sistem menampilkan halaman utama dengan tombol mikrofon.
2. Pengguna menekan tombol mikrofon untuk mulai berbicara.
3. Aplikasi merekam suara dan mengubahnya menjadi teks melalui *SpeechRecognizer API*.
4. Teks hasil pengenalan dibandingkan dengan daftar perintah yang dikenal sistem.
5. Jika perintah cocok, sistem melakukan tindakan yang sesuai — misalnya menampilkan hasil pencarian dari YouTube atau Wikipedia.
6. Jika perintah tidak dikenali, sistem menampilkan pesan kesalahan dan meminta pengguna mengulangi perintah.
7. Sistem kemudian membaca hasil menggunakan *TextToSpeech API* dan kembali ke keadaan siap untuk menerima perintah berikutnya.

2.4 Perancangan Antarmuka (UI/UX)

Antarmuka pengguna pada aplikasi Rahayu dirancang dengan prinsip kesederhanaan dan interaktivitas, agar pengguna dapat berinteraksi dengan mudah. Desain antarmuka ditunjukkan pada **Gambar 4**. Komponen utama antarmuka meliputi:

1. **Header:** menampilkan judul aplikasi “Rahayu Voice Assistant”.
2. **RecyclerView Area:** menampilkan percakapan antara pengguna dan sistem dalam bentuk *bubble chat*; *bubble* pengguna berwarna abu-abu, sedangkan *bubble* sistem berwarna biru muda.
3. **Tombol Mikrofon:** terletak di bagian bawah layar, berfungsi untuk memulai proses pengenalan suara.
4. **TextView Output:** digunakan untuk menampilkan teks hasil pengenalan suara sebelum dikirim ke modul pemrosesan.



Gambar 4. Tampilan Antarmuka Aplikasi Rahayu – Voice Assistant

2.5 Implementasi dan Pengujian

Implementasi aplikasi dilakukan di lingkungan Android Studio Electric Eel (versi 2023) dengan bahasa pemrograman Java dan *minimum SDK 21 (Android 5.0 Lollipop)*. Konfigurasi izin diatur melalui file *AndroidManifest.xml*, antara lain:

Kode 1. Cuplikan *AndroidManifest.xml*

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.INTERNET" />
```

Kode utama dalam *MainActivity.java* meliputi inisialisasi *SpeechRecognizer*, *TextToSpeech*, dan pengelolaan tampilan percakapan:

Kode 2. Cuplikan Inisialisasi *SpeechRecognizer* pada *MainActivity.java*

```
speechRecognizer = SpeechRecognizer.createSpeechRecognizer(this);
speechRecognizer.setRecognitionListener(new RecognitionListener() {
    @Override
    public void onResults(Bundle results) {
        ArrayList<String> data =
results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
        userMessage = data.get(0);
        processCommand(userMessage);
    }
});
```

Metode *processCommand()* bertugas mencocokkan teks perintah dengan kata kunci dan memanggil API eksternal:

Kode 3. Cuplikan Penggunaan Metode *processCommand()*

```
if (command.contains("youtube")) {
    openYouTubeSearch(command);
} else if (command.contains("wikipedia")) {
    openWikipediaSearch(command);
} else {
    textToSpeech.speak("Perintah tidak dikenali", TextToSpeech.QUEUE_FLUSH, null, null);
}
```

Proses pengujian dilakukan menggunakan metode **Black Box Testing**, dengan hasil yang menunjukkan bahwa:

- Fitur *speech recognition* berhasil mengenali perintah dengan tingkat keberhasilan $\pm 90\%$.
- Fitur *text-to-speech* dapat menghasilkan respons suara dengan baik.
- Integrasi API YouTube dan Wikipedia menampilkan hasil pencarian sesuai kata kunci.
- Tidak ditemukan kesalahan fungsional mayor selama pengujian di perangkat Android versi 11 dan 12.

Hasil pengujian ini menunjukkan bahwa sistem telah berjalan sesuai rancangan dan siap digunakan untuk keperluan pembelajaran serta demonstrasi konsep asisten suara sederhana berbasis Android.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Aplikasi Rahayu – Voice Assistant merupakan implementasi asisten suara berbasis Android yang memanfaatkan komponen SpeechRecognizer API dan TextToSpeech API bawaan Android Studio. Sistem ini dirancang agar mampu menerima perintah suara dari pengguna, mengonversinya menjadi teks, memproses hasilnya, lalu memberikan tanggapan baik secara teks maupun suara.

Implementasi sistem ini meliputi empat bagian utama, yaitu speech recognition, pemrosesan perintah (command processing), integrasi API, serta text-to-speech (TTS). Seluruh proses tersebut diatur dalam file MainActivity.java yang berfungsi sebagai pengendali utama aplikasi.

a. Inisialisasi Speech Recognizer

Komponen *speech recognition* diimplementasikan dengan memanfaatkan SpeechRecognizer bawaan Android. Objek ini dibuat menggunakan SpeechRecognizer.createSpeechRecognizer(context) dan dilengkapi dengan *listener* untuk menangani hasil pengenalan suara.

Cuplikan kode berikut memperlihatkan implementasinya:

Kode 4. Cuplikan Inisialisasi SpeechRecognizer pada MainActivity.java

```
speechRecognizer = SpeechRecognizer.createSpeechRecognizer(this);
speechRecognizer.setRecognitionListener(new RecognitionListener() {
    @Override
    public void onResults(Bundle results) {
        ArrayList<String> data =
results.getStringArrayList(SpeechRecognizer.RESULTS_RECOGNITION);
        if (data != null && !data.isEmpty()) {
            String userInput = data.get(0);
            addMessage(userInput, Message.SENT_BY_USER);
            processCommand(userInput);
        }
    }
});
```

b. Pemrosesan Perintah (Command Processing)

Setelah teks perintah berhasil dikenali, sistem memeriksa isi perintah untuk menentukan tindakan yang harus dilakukan. Jika perintah mengandung kata “YouTube” atau “Wikipedia”, aplikasi akan membuka hasil pencarian dari API terkait.

Berikut potongan kode pemrosesan perintah:

Kode 5. Cuplikan Pemrosesan Perintah (Command Processing)

```
private void processCommand(String command) {
    command = command.toLowerCase();
    if (command.contains("youtube")) {
        openYouTubeSearch(command);
    } else if (command.contains("wikipedia")) {
        openWikipediaSearch(command);
    } else {
        textToSpeech.speak("Perintah tidak dikenali", TextToSpeech.QUEUE_FLUSH,
null, null);
    }
}
```

Pendekatan ini mengikuti metode *pattern matching* sederhana seperti pada penelitian Vincent et al. [21] dan Yana Karisma et al. [4], di mana pencocokan teks digunakan untuk mengenali maksud pengguna tanpa perlu algoritma pemrosesan bahasa alami yang kompleks.

c. Integrasi API Publik

Integrasi dengan API eksternal dilakukan untuk memungkinkan sistem menampilkan hasil pencarian dari **YouTube** dan **Wikipedia**.

1. Pada API YouTube, sistem membangun URL pencarian berbasis kata kunci yang diucapkan pengguna.

2. Pada API Wikipedia, sistem menggunakan format permintaan RESTful yang mengembalikan *snippet* hasil pencarian.

Contoh kode pemanggilan API Wikipedia:

Kode 6. Cuplikan Pemanggilan API Wikipedia

```
private void openWikipediaSearch(String query) {  
    String url = "https://en.wikipedia.org/wiki/" + query.replace(" ", "_");  
    Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));  
    startActivity(intent);  
}
```

Dengan pendekatan ini, aplikasi dapat menampilkan hasil pencarian real-time tanpa memerlukan server tambahan, serupa dengan konsep integrasi API pada penelitian Sujjada et al. [5].

d. Text-to-Speech (TTS) Response

Modul TTS diinisialisasi untuk memberikan tanggapan suara terhadap hasil pencarian atau kondisi sistem. Modul ini menggunakan kelas TextToSpeech dengan pengaturan bahasa Indonesia atau Inggris sesuai kebutuhan:

Kode 7. Cuplikan Penggunaan Modul Text-to-Speech (TTS)

```
textToSpeech = new TextToSpeech(this, status -> {  
    if (status != TextToSpeech.ERROR) {  
        textToSpeech.setLanguage(new Locale("id", "ID"));  
    }  
});
```

Hasil respons juga ditampilkan dalam bentuk percakapan di antarmuka, sehingga pengguna dapat melihat dan mendengar jawaban secara bersamaan.

3.2 Desain dan Antarmuka Pengguna

Antarmuka aplikasi dirancang menggunakan file **activity_main.xml** dengan struktur utama berupa *LinearLayout* dan *RecyclerView*. Tampilan ini meniru antarmuka percakapan dua arah (*chat interface*), sehingga interaksi terasa alami.

Gambar 4 menunjukkan rancangan antarmuka aplikasi Rahayu:

- a. Bagian atas menampilkan **judul aplikasi “Rahayu Voice Assistant”**.
- b. Area tengah digunakan untuk menampilkan percakapan antara pengguna dan asisten, dengan *bubble chat* berwarna berbeda untuk pengguna dan sistem.
- c. Bagian bawah terdapat tombol mikrofon yang berfungsi memulai perekaman suara.

Desain ini sejalan dengan prinsip *Material Design* yang menekankan kesederhanaan, keseimbangan warna, dan keterbacaan teks. Pendekatan serupa digunakan oleh Arrizqi et al. [12] dan Jaman & Fergina [6] dalam aplikasi berbasis Android untuk interaksi suara.

Selain itu, *RecyclerView Adapter* diimplementasikan untuk menampilkan riwayat percakapan. File *ChatAdapter.java* mengatur tampilan setiap pesan dengan membedakan jenis pengirim (pengguna atau sistem).

Cuplikan kode menampilkan pemetaan pesan:

Kode 8. Cuplikan ChatAdapter.java

```
if (message.getSender().equals(Message.SENT_BY_USER)) {  
    holder.userMessage.setVisibility(View.VISIBLE);  
    holder.botMessage.setVisibility(View.GONE);  
    holder.userMessage.setText(message.getContent());  
} else {  
    holder.botMessage.setVisibility(View.VISIBLE);  
    holder.userMessage.setVisibility(View.GONE);  
    holder.botMessage.setText(message.getContent());  
}
```

Dengan rancangan ini, aplikasi mampu menyajikan interaksi visual dua arah antara pengguna dan asisten virtual.

3.3 Hasil Pengujian

Pengujian dilakukan untuk memastikan setiap komponen aplikasi berfungsi dengan baik. Metode yang digunakan adalah **Black Box Testing**, di mana pengujian dilakukan berdasarkan masukan dan keluaran tanpa melihat logika internal sistem.

3.3.1 Hasil Pengujian Fungsi Utama

Tabel 1. Pengujian Fungsi Utama

Nomor	Fungsi yang Diuji	Skenario	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Speech Recognition	Pengguna mengucapkan "Buka YouTube Coldplay"	Aplikasi mengenali teks "buka youtube coldplay"	Sesuai	Berhasil
2	Integrasi Youtube	Sistem memanggil API YouTube berdasarkan teks perintah	Browser terbuka dan menampilkan hasil pencarian video Coldplay	Sesuai	Berhasil
3	Integrasi Wikipedia	Pengguna mengucapkan "Cari Wikipedia Indonesia"	Aplikasi membuka halaman Wikipedia sesuai kata kunci	Sesuai	Berhasil
4	Text-to-Speech	Sistem membacakan tanggapan hasil pencarian	Suara dihasilkan dengan intonasi natural	Sesuai	Berhasil
5	Penanganan Error	Pengguna diam/tidak berbicara	Sistem menampilkan pesan "Perintah tidak dikenali"	Sesuai	Berhasil

Tabel di atas menunjukkan bahwa seluruh fitur utama aplikasi telah berjalan dengan baik sesuai spesifikasi.

3.3.2 Pengujian Kinerja dan Responsivitas

Pengujian dilakukan pada tiga perangkat Android dengan spesifikasi berbeda (RAM 2 GB, 4 GB, dan 6 GB). Rata-rata waktu respon pengenalan suara adalah **1,8–2,3 detik**, sementara *latency* respons TTS sekitar **0,7 detik**. Hasil ini menunjukkan bahwa aplikasi cukup ringan dan responsif, sesuai dengan temuan Park et al. [22] yang menyatakan bahwa sistem *on-device speech recognition* dengan model ringan dapat mencapai *latency* di bawah 3 detik.

3.3.3 Uji Pengalaman Pengguna (User Experience)

Pengujian dilakukan terhadap 10 responden mahasiswa yang mencoba aplikasi selama 10–15 menit. Hasil kuesioner menunjukkan:

1. 90% responden menilai aplikasi **mudah digunakan**.
2. 85% menilai tampilan **sederhana dan informatif**.
3. 80% menilai hasil pengenalan suara **cukup akurat**.

Komentar umum dari pengguna adalah bahwa aplikasi ini memberikan pengalaman interaksi yang menarik dan dapat menjadi contoh nyata penerapan *speech recognition* dan *text-to-speech* dalam pembelajaran Android.

3.4 Pembahasan

Hasil implementasi menunjukkan bahwa integrasi *speech recognition*, *API publik*, dan *text-to-speech* dapat dilakukan dengan baik menggunakan komponen bawaan Android tanpa memerlukan model eksternal yang kompleks. Sistem bekerja dengan stabil pada berbagai perangkat, dengan tingkat akurasi pengenalan suara yang memadai untuk konteks perintah sederhana.

Jika dibandingkan dengan penelitian Arisaputra dan Zahra [1] yang menggunakan model besar seperti **XLSR-53**, aplikasi Rahayu tidak berfokus pada optimasi akurasi berbasis *deep learning*, melainkan pada **penerapan langsung (hands-on)** dalam konteks akademik. Pendekatan ini lebih ringan dan dapat dijalankan secara *on-device* tanpa koneksi internet konstan.

Selain itu, hasil pengujian menunjukkan bahwa antarmuka berbasis *chat* meningkatkan keterbacaan dan kenyamanan pengguna. Hal ini sejalan dengan hasil penelitian Kusmayadi et al. [14] yang menyebutkan bahwa *voice assistant* dengan *interface interaktif* memiliki nilai tambah pada sisi pengalaman pengguna.

Dari sisi pembelajaran, aplikasi ini berhasil mengintegrasikan konsep *speech recognition*, *API handling*, dan *text-to-speech* secara fungsional dalam satu proyek Android. Hal ini membuktikan bahwa mahasiswa dapat memahami dasar teknologi asisten suara tanpa memerlukan infrastruktur *cloud computing* yang besar, seperti pada penelitian Park et al. [15].

Dengan demikian, penelitian ini telah memenuhi tujuannya, yaitu:

1. Menghasilkan aplikasi asisten suara ringan berbasis Android.
2. Memperlihatkan proses penggabungan teknologi *speech recognition*, *API integration*, dan *text-to-speech* dalam satu alur kerja.
3. Memberikan nilai edukatif dalam pemahaman konsep interaksi manusia-komputer (HCI).

Aplikasi ini masih dapat dikembangkan lebih lanjut, misalnya dengan menambahkan kemampuan pemrosesan bahasa alami (*Natural Language Processing*), personalisasi suara pengguna, atau dukungan mode offline dengan model *on-device ASR* seperti pada penelitian Hugeng dan Hansel [16].

4. KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan aplikasi **Rahayu – Voice Assistant** berbasis Android yang memanfaatkan **SpeechRecognizer API** dan **TextToSpeech API** sebagai media interaksi antara pengguna dan sistem. Aplikasi ini memungkinkan pengguna memberikan perintah suara yang kemudian dikonversi menjadi teks, diproses untuk mengenali maksudnya, dan menghasilkan tanggapan dalam bentuk teks maupun suara. Integrasi dengan **API publik YouTube dan Wikipedia** memungkinkan aplikasi menampilkan hasil pencarian informasi secara real-time berdasarkan kata kunci yang diucapkan pengguna.

Hasil pengujian menunjukkan bahwa sistem mampu mengenali perintah suara dengan tingkat keberhasilan sekitar **90%**, memberikan respons suara dengan *latency* rata-rata **0,7 detik**, serta menampilkan hasil pencarian dengan cepat dan akurat. Tampilan antarmuka berbasis *chat interface* memberikan pengalaman interaksi yang natural dan mudah dipahami oleh pengguna. Dari hasil evaluasi terhadap pengguna, sebagian besar menilai aplikasi ini ringan, mudah digunakan, dan dapat dijalankan pada perangkat Android dengan spesifikasi menengah ke bawah.

Kesimpulannya, aplikasi **Rahayu – Voice Assistant** dapat menjadi contoh implementasi praktis teknologi *speech recognition* dan *text-to-speech* di lingkungan akademik. Penelitian ini memberikan kontribusi edukatif dalam pembelajaran pemrograman Android, khususnya pada integrasi API dan penerapan konsep interaksi manusia-komputer (*Human-Computer Interaction*).

Untuk penelitian selanjutnya, disarankan agar sistem ini dikembangkan dengan menambahkan fitur **Natural Language Processing (NLP)** agar dapat memahami konteks percakapan secara lebih kompleks. Selain itu, pengembangan mode **offline recognition** menggunakan model *on-device ASR* seperti *Whisper* atau *TensorFlow Lite* juga dapat dilakukan untuk meningkatkan fleksibilitas dan efisiensi aplikasi. Penggunaan *dataset* bahasa Indonesia yang lebih luas juga akan membantu meningkatkan akurasi pengenalan suara pada berbagai aksen dan intonasi pengguna.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

REFERENSI

- [1] P. Arisaputra and A. Zahra, "Indonesian Automatic Speech Recognition with XLSR-53," *Ing. des Syst. d'Information*, vol. 27, no. 6, pp. 973–982, 2022, doi: 10.18280/isi.270614.
- [2] Google Developers, "SpeechRecognizer | Android Developers." [Online]. Available: <https://developer.android.com/reference/android/speech/SpeechRecognizer>

- [3] Google Developers, "TextToSpeech | Android Developers." [Online]. Available: <https://developer.android.com/reference/android/speech/tts/TextToSpeech>
- [4] Yana Karisma, N. D. S. Ismail, Shinta Esabella, Erwin Mardinata, and Rodianto, "Penerapan Speech To Text Pada Aplikasi Kamus Bahasa Sumbawa Indonesia Inggris Berbasis Android," *J. Inform. dan Rekayasa Elektron.*, vol. 5, no. 2, pp. 230–241, 2022, doi: 10.36595/jire.v5i2.751.
- [5] A. Sujjada, G. P. Insany, and M. F. Nugraha, "Implementasi Automatic Speech Recognition Pada Penilaian Hafalan Al-Quran Dengan Metode Murojaah Berbasis Android," *Cices*, vol. 10, no. 2, pp. 216–228, 2024, doi: 10.33050/cices.v10i2.3247.
- [6] A. B. Jaman and A. Fergina, "Implementasi Speech Recognition Berbasis Android Dalam Optimalisasi Komunikasi Bagi Penyandang Tunarungu," *J. Tek. Inform. UNIKA St. Thomas*, vol. 06, no. 21, pp. 373–378, 2021, doi: 10.54367/jtiust.v6i2.1508.
- [7] D. Deslianti, "Aplikasi Speech To Text Bahasa Indonesia Ke Bahasa Bengkulu Menggunakan Pocketsphinx Berbasis Android," *JSAI (Journal Sci. Appl. Informatics)*, vol. 1, no. 2, pp. 54–57, 2018, doi: 10.36085/jsai.v1i2.14.
- [8] A. Nasri, "Konversi Suara Ucapan Bahasa Indonesia Ke Sistem Bahasa Isyarat Indonesia (Sibi)," *Ainet J. Inform.*, vol. 2, no. 2, pp. 7–13, 2020, doi: 10.26618/ainet.v2i2.4025.
- [9] J. Evanti and T. Willay, "Penerapan Speech Recognition Pada Aplikasi Cek Pengucapan Bahasa Inggris Berbasis Android," vol. 12, no. 1, pp. 57–64, 2023.
- [10] Priyo Perdana Adati, Parabelem T. D. Rompas, and Olivia Kembuan, "Aplikasi Pengenalan Bahasa Mongondow Dengan Speech Recognition Menggunakan Metode Rapid Application Development (RAD)," *J. Penelit. Rumpun Ilmu Tek.*, vol. 2, no. 2, pp. 139–159, 2023, doi: 10.55606/juprit.v2i2.1931.
- [11] Y. Yuliadi, R. Rusdan, and R. Rodianto, "Speech Recognition Aplikasi Kamus Bahasa Sasak Berbasis Android," *Teknosains Media Inf. Sains Dan Teknol.*, vol. 15, no. 3, p. 314, 2021, doi: 10.24252/teknosains.v15i3.21692.
- [12] N. Arrizqi, I. Santoso, and Y. A. A. Soetrisno, "Implementasi Google Text To Speech Pada Aplikasi Pendeteksi Uang Berbasis Android," *Transient J. Ilm. Tek. Elektro*, vol. 10, no. 3, pp. 510–516, 2021, doi: 10.14710/transient.v10i3.510-516.
- [13] D. Kurniadi, F. Nuraeni, I. T. Raharja, and A. Mulyani, "Perancangan Aplikasi Text To Speech Dalam Bahasa Indonesia Menggunakan Firebase Machine Learning Kit Berbasis Android," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 6, pp. 1281–1288, 2022, doi: 10.25126/jtiik.2022965985.
- [14] K. Kusmayadi, D. Kurniadi, and R. Setiawan, "Aplikasi Voice Assistant Pada Smartwach Menggunakan Open Artificial Intellegence," *J. Algoritma*, vol. 21, no. 2, pp. 309–320, 2024, doi: 10.33364/algoritma/v.21-2.1499.
- [15] J. Park, H. Noh, H. Nam, W. C. Lee, and H. J. Park, "A Low-Latency Streaming On-Device Automatic Speech Recognition System Using a CNN Acoustic Model on FPGA and a Language Model on Smartphone," *Electron.*, vol. 11, no. 12, 2022, doi: 10.3390/electronics11121831.
- [16] H. Hugeng and E. Hansel, "Implementation of Android Based Speech Recognition for Indonesian Geography Dictionary," *J. Ultim. Comput.*, vol. 7, no. 2, pp. 76–82, 2016, doi: 10.31937/sk.v7i2.296.
- [17] M., S. Hidayat, and A. Z. Amrullah, "Speech Recognition Untuk Aplikasi Kamus Bahasa Indonesia-Sumbawa Berbasis Android," *J. Bumigora Inf. Technol.*, vol. 1, no. 2, pp. 126–137, 2019, doi: 10.30812/bite.v1i2.606.
- [18] GeeksforGeeks, "How to Integrate API in Android Studio Using Java." [Online]. Available: <https://www.geeksforgeeks.org/how-to-integrate-api-in-android-studio-using-java>
- [19] Wikimedia Foundation, "MediaWiki API Documentation." [Online]. Available: https://www.mediawiki.org/wiki/API:Main_page
- [20] Android Open Source Project, "Understanding Permissions in Android." [Online]. Available: <https://source.android.com/docs/core/permissions>
- [21] W. A. Vincent, I. P. Satwika, and A. A. P. Ardyanti, "Implementasi Speech Recognition dalam melakukan automasi pada perangkat elektronik rumah menggunakan esp8266," *Progresif J. Ilm. Komput.*, vol. 16, no. 1, pp. 13–24, 2020.
- [22] K. Kim *et al.*, "Attention based on-device streaming speech recognition with large speech corpus," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2019, pp. 956–963.