

Aplikasi *Voice Assistant* Pada *Smartwatch* Menggunakan *Open Artificial Intelligence*

Kusmayadi¹, Dede Kurniadi^{2*}, Ridwan Setiawan³
^{1,2,3}Institut Teknologi Garut, Indonesia

*email: dede.kurniadi@itg.ac.id

Info Artikel

Dikirim: 3 Oktober 2023

Diterima: 30 November 2023

Diterbitkan: 30 November 2024

Kata kunci:

Open Artificial Intelligence;

Smartwatch;

Voice Assistant.

ABSTRAK

Voice assistant digunakan sebagai teknologi untuk memproses bahasa alami dari pengguna dan memberikan respon perintah dari pengguna. *Voice assistant* digunakan sebagai *assistant* digital mempermudah pekerjaan pengguna dengan memberi perintah suara, *voice assistant* akan memberikan respon pencarian pada *device* yang terkoneksi dengan adanya fitur yang memberikan kontrol pada *device*, *voice assistant* pada saat ini banyak di terapkan pada *device smartphone*, untuk melakukan perintah pada *voice assistant* tidak bisa menanggapi perintah untuk tanya jawab dari pengguna secara langsung, *voice assistant* pada *smartphone* tidak memberi tingkat kepercayaan bahwa penggunaan sebuah *system* akan meningkatkan kinerja (*perceived usefulness*). Tujuan dari penelitian ini mendapatkan respon tanya jawab pengguna dengan *system* dari proses *open artificial intelligence*, kemudian memberikan kinerja kerja bagi pengguna dimana *voice assistant* diterapkan pada *smartwatch*, untuk melengkapi fitur yang ada pada penelitian sebelumnya dan menambahkan bahasa dukungan *text to speech*. Penelitian ini menggunakan metode *Extreme Programming (XP)* dengan tahapan yang digunakan yaitu tahapan *planning, design, coding, dan testing* dengan pemodelan *Unified Modelling Language (UML)*. Hasil dari penelitian ini adalah dibangunnya aplikasi *voice assistant* pada *smartwatch* yang dapat berfungsi dengan baik, yang divalidasi menggunakan pengujian *usability testing*, kemudian dapat melakukan tanya jawab perintah suara dari pengguna berupa informasi yang di tanyakan secara *random* dan memberikan *perceived usefulness* bagi pengguna.

1. PENDAHULUAN

Kemajuan teknologi sendiri dalam sejarah perkembangan manusia, hal ini tidak dapat dihindari. *Assistant virtual*, juga dikenal sebagai *assistant artificial intelligence*, juga berkembang dengan cepat. Perusahaan - Perusahaan besar dalam industri teknologi bersaing untuk membuat *virtual assistant* terbaik. Seperti *Google Assistant*, *Siri* dari *Apple*, dan *Alexa* dari *Amazon*, antara lain. Salah satu jenis *virtual assistant* yang sedang menjadi perdebatan hangat di masyarakat adalah *voice assistant* [1]. *Voice assistant* ini merupakan asisten digital yang menggunakan pengenalan suara atau proses pemrosesan bahasa natural (NLP) untuk melaksanakan perintah yang diberikan oleh pengguna [1]. Contoh *Digital assistant* diantaranya contara, *siri*, dan *google now* untuk ketiga *assistant digital* ini hanya merespons suatu program yang telah diatur hanya untuk memberi perintah prediksi cuaca, pengaturan alarm, pencarian dengan ketik pencarian, pengenalan musik, kalender, pencarian dengan ketik, akses dengan aplikasi, melakukan panggilan, mengirim pesan dan membuka

email pada ketiga *assistant digital* ini [2]. Pada *digital assistant* untuk *voice assistant* tersebut dirasa kurang *efisien*, dari teknologi tersebut untuk merespon pertanyaan *random* dari pengguna dimana saat penanyakan suatu informasi yang belum di ketahui oleh pengguna belum bisa menjawab atau menjelaskan kemudian untuk merespon pertanyaan dari pengguna akan di arahkan terlebih dahulu ke mesin pencarian seperti *google*, *bing*, *yahoo* dan mesin pencarian lainnya [2]. *Voice assistant* ini banyak diterapkan pada *device smartphone* yang memberikan tidak adanya *perceived usefulness* dimana pada penelitian ini akan diterapkan pada *smartwatch* ada faktor yang mendorong orang untuk terus menggunakan jam tangan pintar: kesehatan, persepsi kegunaan (*Usefulness*), kebiasaan (*Habits*), motivasi hedonis, persepsi estetis, barang pelengkap, dan mobilitas individu. Perusahaan juga harus mempertimbangkan kesenangan dan keindahan saat mendesain jam tangan pintar. Produsen jam tangan pintar juga harus terus membuat aplikasi yang memudahkan penggunaan jam tangan pintar dan bekerja sama dengan pengembang aplikasi populer seperti Strava [3].

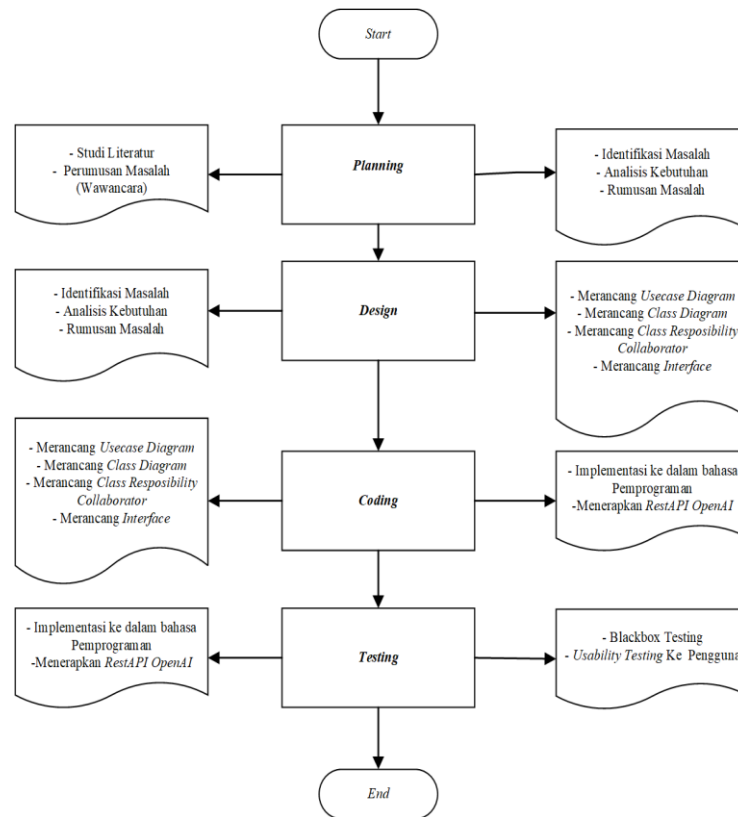
Dalam penelitian oleh Meganingrum (2021) membahas mengenai memanfaatkan informasi yang diberikan oleh *Google Assistant* saat mahasiswa membutuhkannya. Terbukti bahwa pemanfaatan informasi *Google Assistant* pada *smartphone* Android memenuhi kebutuhan informasi siswa dengan efektif. Seseorang memiliki lima kebutuhan informasi: Kebutuhan kognitif, afektif, integratif sosial, dan pelepasan ketegangan. Menurut hasilnya, penelitian ini memenuhi kebutuhan integratif sebagian besar orang adalah pernyataan ke-6 dan ke-7, dengan R 0,768. Ini menunjukkan bahwa *Google Assistant* dapat membantu mahasiswa dalam banyak hal, terutama mencari informasi. Dengan memberikan informasi yang tepat, *Google Assistant* dapat meningkatkan kepercayaan mahasiswa [4]. Pada penelitian kedua ini yang dilakukan oleh Kurniadi et al. (2022) yaitu penerapan *firebase machine learning kit* pada aplikasi *text to speech* berbasis android, berupa aplikasi yang dapat digunakan sebagai alat bantu pembelajaran bahasa asing dan alat digitalisasi teks serta terjemah ke dalam Bahasa Indonesia dan 34 dialek bahasa untuk keluaran suara *text to speech* [5]. Penelitian ketiga yang di lakukan oleh Mulyana et al. (2018) yaitu perancangan aplikasi *Speech to Text* yang merupakan sebuah mesin pencari suara yang digunakan di Elevenia.co.id. Aplikasi ini memungkinkan pelanggan menemukan produk tanpa mengetik di papan ketik dan merekam suara yang mengatakan produk yang mereka cari [6]. Penelitian rujukan yang keempat oleh Rizal et al. (2017) menerapkan perintah suara menggunakan Raspberry Pi, Dengan menggunakan Google Speech API dimana menggunakan *smartphone* pintar untuk melakukan terjemahan *promt voice* yang diberikan oleh *user* ke *system* [7]. Penelitian yang kelima oleh Haryanto et al. (2021) hasil yang didapat dari penelitian ini yaitu pada penelitian ini akan mengembangkan model sistem *Smart Home* yang menggunakan sensor suara dan NodeMCU sebagai pusat pengendali. *Google Assistant* akan berfungsi sebagai alat pengendali dan teknologi wifi akan digunakan untuk mengakses NodeMCU dengan modul ESP8266 [8]. Penelitian yang keenam oleh Setiawan (2019) sistem yang dapat mengontrol kompor pintar menggunakan perintah suara. Dengan menggunakan *engine* *Google Assistant* yang ada di *smartphone* [9]. Banyaknya *virtual assistant* yang bisa digunakan seperti *voice assistant* untuk meningkatkan kinerja kerja dan mempermudah aktivitas. *Speech recognition* untuk mengenali perintah kata dari suara manusia dan kemudian di terjemahkan ke bahasa komputer menggunakan *device* [10].

Maka tujuan dari penelitian ini yaitu membuat sebuah aplikasi *voice assistant* yang dapat menanggapi jawaban dari pertanyaan pengguna dengan tanggapan jawaban tersebut secara langsung, dari proses *open artificial intelligence*. *Open artificial intelligence* adalah mesin cerdas yang dilatih untuk bisa menirukan percakapan manusia menggunakan teknologi NLP (*Natural Language Processing*) [11]. dengan menggunakan *smartwatch* untuk melengkapi *fitur* yang ada pada penelitian sebelumnya dan menambahkan bahasa dukungan *text to speech* yang dapat mengeluarkan suara, dan memberikan tanggapan yang sesuai dengan perintah pertanyaan yang diberikan oleh pengguna untuk mendapatkan respons jawaban yang sesuai. Penelitian ini menggunakan metode *Extreme Programming* dengan pemodelan sistem *Unified Modelling Language* menurut [10] hasil analisis dan pengujian menunjukkan bahwa metode XP dalam tahap perancangan aplikasi mempercepat proses perencanaan untuk setiap perubahan yang diperlukan untuk fungsi-fungsi yang diinginkan.

2. METODOLOGI PENELITIAN

Metodologi yang digunakan pada penelitian ini yaitu *Extreme Programming* (XP) adalah adalah model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih

efisien, adaptif dan fleksibel [12]. Penelitian ini dilaksanakan dengan beberapa fase aktivitas perencanaan hingga pengujian aplikasi, dimana fase tersebut merupakan pencapaian tujuan penelitian yang direncanakan. Dimana fase aktivitas tersebut dituangkan pada kerangka penelitian dapat dilihat pada gambar 1 berikut:



Gambar 1. Kerangka Penelitian

Dalam penelitian ini menerapkan pendekatan metodologi *Extreme Programming*, dimana uraian penjelasan dari Gambar 1 adalah sebagai berikut:

- 1) *Planning*
Tahapan ini merupakan langkah awal dalam pembangunan sistem dimana dalam tahapan ini dilakukan beberapa kegiatan perencanaan yaitu, identifikasi permasalahan, menganalisa kebutuhan sampai dengan penetapan jadwal pelaksanaan pembangunan sistem.
- 2) *Design*
Tahapan design, dibuatlah model sistem berdasarkan hasil pada tahap sebelumnya. Merupakan pendekatan dari bahasa berorientasi objek dengan pemodelan sistem menggunakan UML, tahapan pemodelan system dengan membuat *Usecase diagram*, *class diagram*, *Class Responsibility Collaborator*, dan Merancang *Interface*.
- 3) *Coding*
Tahapan ini merupakan proses implementasi perancangan (yang telah di bangun di tahap sebelumnya), ke dalam proses pengkodean (*coding*) menggunakan bahasa pemrograman dan menerapkan RestAPI OpenAI.
- 4) *Testing*
Tahapan pengujian sistem untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Pada tahap ini, menggunakan *alpha testing* aplikasi diuji internal dengan *blackbox testing* untuk menemukan *bug* dan memastikan stabilitas sebelum tahap *beta testing*. *Beta testing* melibatkan *usability testing* dengan pengguna lebih luas untuk mengevaluasi interaksi dan memastikan kepuasan pengguna sebelum peluncuran resmi.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

Pembuatan aplikasi *voice assistant* pada *smartwatch* menggunakan *open artificial intelligence* menggunakan metodologi *Extreme Programming (XP)*. Berikut merupakan hasil pembahasan penelitian aktivitas yang terdapat pada metodologi XP yang dijabarkan dalam kerangka pemikiran.

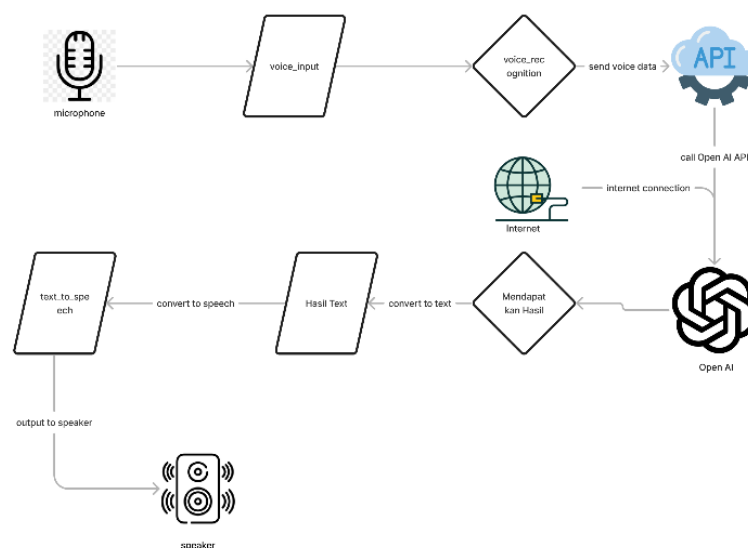
1) *Planning*

Tahap *planning* merupakan langkah pertama yang dilakukan dalam pembuatan aplikasi ini. Terdapat beberapa aktivitas yang dilakukan pada tahap *planning* ini, diantaranya identifikasi masalah, analisa kebutuhan, dan rumusan masalah.

- a. Identifikasi masalah dimana aktivitas pada tahap ini yaitu melakukan studi literatur, wawancara dan melakukan *quisioner*.
- b. Analisa kebutuhan, melakukan analisa mengenai permasalahan yang muncul pada penelitian sebelumnya.
- c. Rumusan masalah, pada aktivitas ini merujuk pada tahap studi litelatur, wawancara, dan kuisisioner dalam dibuatkan rumusan masalah bagaimana merancang bangun aplikasi *voice assistant* pada *smartwatch* dengan menerapkan teknologi *open artificial intelligence*.

2) *Design*

Tahap pemodelan melibatkan seluruh proses *modeling* berdasarkan analisis kebutuhan dari tahap sebelumnya. Adapun model sistem yang telah di desain mengintegrasikan beberapa komponen kunci, disajikan pada Gambar 2.



Gambar 2. Model Sistem

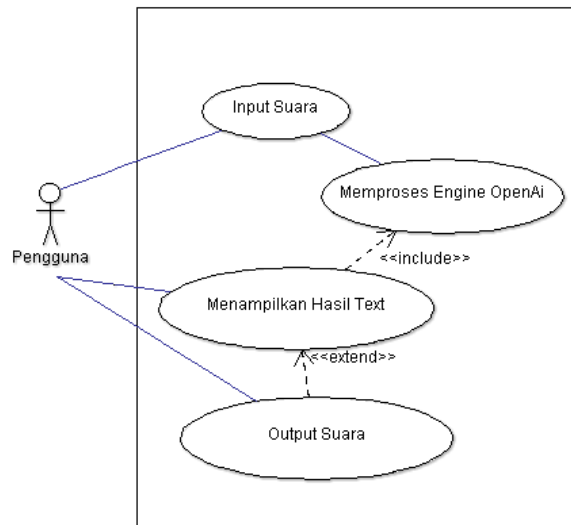
Pada model *system* ini pengguna melakukan perintah suara *input* ke *microphone* kemudian menerima *input* suara kemudian di proses menggunakan *voice recognition* yang akan di kirim ke *engine* openai melalui *rest API* yang terkoneksi jaringan internet, kemudian dari proses *engine* openai. *System* melakukan pengambilan hasil (*Get result*) data yang di *convert* menjadi *text* yang akan ditampilkan *text* tersebut pada layar kemudian proses kembali menjadi keluaran suara melalui *external speaker* atau *speaker internal* perangkat.

Pada tahap ini meliputi beberapa aktivitas perancangan diagram dan perancangan *interface*. Penggambaran awalnya di implementasikan kedalam *use case diagram*, *class diagram*, *class*

responsibility collaboration (CRC), dan perancangan *interface*.

1) *Use Case Diagram*

Adapun untuk usecase yang telah dirancang dapat dilihat pada Gambar 3. berikut :



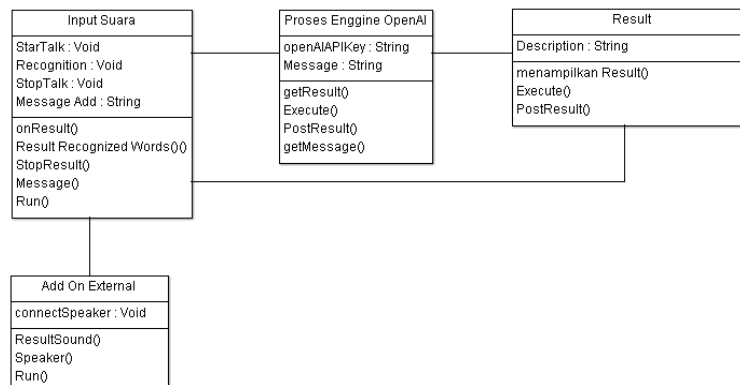
Gambar 3. *Use Case Diagram*

Penjelasan setiap use case diatas diantaranya:

- a Input Suara:
 - a) Memulai *microphone* agar dapat di pakai saat memulai *Speech Recognition*.
 - b) Melakukan *speech recognition* kemudian di proses *speech to text*.
 - c) Menghentikan *microphone* yang sedang di jalankan. Kemudian menghentikan proses *speech recognition* dan melakukan pengiriman data suara yang telah di proses menjadi *text* ke *engine openai* sebagai *prompt* dari pengguna.
- b Proses *Engine OpenAI*: Melakukan proses respon pertanyaan, melalui pengeriman *rest API*.
- c Menampilkan Hasil *Text*: Menampilkan hasil *text* yang telah di proses oleh *engine openAI*
- d *Output Suara*: Melakukan *convert* ke suara dari *text* yang telah di hasilkan, dan di tampilkan melalui speaker *internal* atau *external* dari perangkat

2) *Class Diagram*

Adapun *class diagram*, yang digunakan untuk menggambarkan struktur system berdasarkan kelas-kelas data yang saling terhubung melalui atribut dan metode, disajikan dalam Gambar 4. untuk aplikasi *Voice Assistant*.



Gambar 4. *Class Diagram*

3) *Class Responsibility Collaboration (CRC)*

Class Responsibility Collaboration (CRC), diagram yang digunakan untuk menggambarkan tanggung jawab dan kolaborasi antar kelas dalam *system voice assistant*, disajikan dalam Gambar 5. Untuk CRC *input* suara.

Class Name : Input Suara	
Responsibility	Collaboration
<ul style="list-style-type: none"> • Mengelola pengenalan suara di ubah menjadi text kata • Menghasilkan text kata • Menunggu perintah Button stop • Memberikan text kata ke engine restapi openai 	<ul style="list-style-type: none"> • Proses Engine OpenAI

Gambar 5. *Class Responsibility Collaboration Input Suara*

4) Perancangan *Interface*

Adapun perancangan *interface* ini dibuat dengan menggunakan aplikasi Figma. Perancangan *interface* digunakan untuk menciptakan tampilan visual sebelum di implementasikan ke bahasa pemrograman. Perancangan ini disajikan pada Gambar 6. sebagai berikut:



Gambar 6. *Interface Halaman Utama Smartwatch Round*

3) Coding

Setelah melakukan tahapan *planning*, dan *design* untuk tahap selanjutnya yaitu tahapan coding (pengkodean), peneliti akan menggunakan algoritma *voice assistant* dan penerapan *engine* OpenAI untuk mengimplementasikan hasil *desain* ke dalam bahasa pemrograman . Berikut merupakan tahapannya.

a. Algoritma *Voice Assistant*

Input (In):

1. *SpeechToText*: Objek yang digunakan untuk mengenali teks dari suara.
2. *FlutterTts*: Objek yang digunakan untuk mengubah teks menjadi suara (*Text-to-Speech*).
3. *LastWords*: Variabel yang menyimpan hasil teks yang dikenali terakhir dari suara.
4. *OpenAIService*: Objek untuk berkomunikasi dengan layanan *OpenAIService*.
5. *GeneratedContent*: Variabel untuk menyimpan konten yang dihasilkan.
6. *GeneratedImageUrl*: Variabel untuk menyimpan URL gambar yang dihasilkan.
7. *Start*: Nilai awal pengaturan *delay* (*start delay*).
8. *Delay*: Nilai pengaturan *delay* antara langkah-langkah.

Output (Out):

1. Interaksi dengan aplikasi yang memiliki fitur suara (*Speech-to-Text* dan *Text-to-Speech*).
2. Konten yang dihasilkan (teks atau URL gambar).
3. Aplikasi dengan antarmuka yang menampilkan elemen-elemen seperti tombol suara, *chat bubble*, dan fitur daftar (*features list*).

Step (Langkah-Langkah):

1. Inisialisasi:
 - Membuat *class HomePage* yang merupakan turunan dari *StatefulWidget*.
 - *Constructor HomePage* menerima parameter *super.key*.
 - Metode *createState()* mengembalikan objek *_HomePageState*.
 - Dalam *_HomePageState*:
 - Inisialisasi *speechToText*, *flutterTts*, *lastWords*, *openAIService*, *generatedContent*, *generatedImageUrl*, *start*, dan *delay*.

- Metode `initState()`:
 - Memanggil `initSpeechToText()` untuk inialisasi *Speech-to-Text*.
 - Memanggil `initTextToSpeech()` untuk inialisasi *Text-to-Speech*.
 - Metode `initTextToSpeech()`:
 - Menyetel `flutterTts` sebagai instance bersama.
 - Metode `initSpeechToText()`:
 - Menginisialisasi `speechToText`.
2. Interaksi Suara:
- Metode `startListening()`:
 - Memanggil `speechToText.listen` dengan *callback onSpeechResult*.
 - Metode `stopListening()`:
 - Memanggil `speechToText.stop`.
 - Metode `onSpeechResult(result)`:
 - Memperbarui `lastWords` dengan kata-kata yang dikenali dari hasil suara.
 - Metode `systemSpeak(content)`:
 - Memanggil `flutterTts.speak` dengan isi konten sebagai parameter.
 - Metode `setVoice(voice)`:
 - Memanggil `flutterTts.setVoice` dengan suara sebagai parameter.
3. Interaksi dengan Pengguna:
- Metode `dispose()`:
 - Memanggil `speechToText.stop`.
 - Memanggil `flutterTts.stop`.
 - Metode `build(context)`:
 - Mengembalikan sebuah *Scaffold* yang berisi:
 - *AppBar* dengan judul "Stella".
 - *SingleChildScrollView* yang berisi *Column* dengan elemen-elemen berikut:
 - Gambar *asisten virtual*.
 - Balon *chat*.
 - Daftar fitur.
 - Mengatur logika untuk tombol asisten virtual:
 - Jika `speechToText` memiliki izin dan sedang tidak mendengarkan, maka panggil `startListening()`.
 - Jika `speechToText` sedang mendengarkan:
 - Panggil `openAIService.isArtPromptAPI` dengan `lastWords`.
 - Jika hasilnya mengandung 'https', set `generatedImageUrl`, hapus `generatedContent`.
 - Jika tidak, set `generatedContent`, hapus `generatedImageUrl`.
 - Panggil `systemSpeak` dengan isi hasil suara.
 - Panggil `stopListening()`. Jika tidak, panggil `initSpeechToText()`.

b. Algoritma Penerapan REST API OpenAI

Input (In):

1. *Prompt*: Sebuah *string* yang berisi pesan atau input dari pengguna.
2. *Messages*: Sebuah list berisi pesan-pesan yang digunakan dalam interaksi dengan model AI.
3. *OpenAIAPIKey*: Kunci API yang digunakan untuk mengautentikasi akses ke layanan OpenAI.

Output (Out):

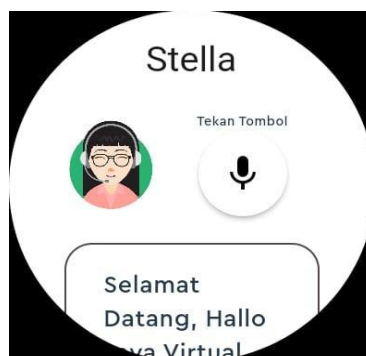
1. Pesan berisi konten yang dihasilkan oleh model AI (teks atau URL gambar).
2. Pemberitahuan atau pesan error jika terjadi masalah dalam akses atau pengolahan data.

Step (Langkah-Langkah):

1. *IsArtPromptAPI(prompt)*:
 - Menambah pesan awal (*role: user*) ke dalam *messages*.
 - Melakukan *POST request* ke 'https://api.openai.com/v1/chat/completions':
 - Mengirimkan headers dengan konten JSON dan bearer token.
 - Mengirimkan data JSON berisi model, pesan awal, dan *prompt*.
 - Memeriksa status kode response:
 - Jika 200:
 - Mengambil isi konten dari hasil respon.
 - Menghapus *whitespace* di sekitar konten.
 - Memeriksa apakah konten berisi 'Yes' atau 'yes':
 - Jika ya, melakukan panggilan ke *dalleAPI(prompt)* dan mengembalikan hasilnya.
 - Jika tidak, melakukan panggilan ke *chatGPTAPI(prompt)* dan mengembalikan hasilnya.
 - Jika bukan 200, mengembalikan pesan error "Maaf Koneksi Jaringan Kamu Tidak Terhubung".
 - Jika ada kesalahan dalam proses, mengembalikan pesan error yang menjelaskan kesalahan.
2. *ChatGPTAPI(prompt)*:
 - Menambah pesan baru (*role: user*) ke dalam *messages*.
 - Melakukan *POST request* ke 'https://api.openai.com/v1/chat/completions':
 - Mengirimkan headers dengan konten JSON dan bearer token.

- Mengirimkan data JSON berisi model dan pesan-pesan yang sudah ada dalam *messages*.
 - Memeriksa status kode response:
 - Jika 200:
 - Mengambil isi konten dari hasil respon.
 - Menghapus whitespace di sekitar konten.
 - Menambah pesan dari AI (*role: assistant*) ke dalam *messages*.
 - Mengembalikan konten.
 - Jika bukan 200, mengembalikan pesan error "Maaf Koneksi Jaringan Kamu Tidak Terhubung".
 - Jika ada kesalahan dalam proses, mengembalikan pesan error yang menjelaskan kesalahan.
3. *DalleAPI(prompt)*:
- Menambah pesan baru (*role: user*) ke dalam *messages*.
 - Melakukan *POST request* ke 'https://api.openai.com/v1/images/generations':
 - Mengirimkan headers dengan konten JSON dan bearer token.
 - Mengirimkan data JSON berisi prompt dan jumlah gambar yang diinginkan.
 - Memeriksa status kode response:
 - Jika 200:
 - Mengambil URL gambar dari hasil respon.
 - Menghapus whitespace di sekitar URL gambar.
 - Menambah pesan dari AI (*role: assistant*) berisi URL gambar ke dalam *messages*.
 - Mengembalikan URL gambar.
 - Jika bukan 200, mengembalikan pesan error "Maaf Koneksi Jaringan Kamu Tidak Terhubung".
 - Jika ada kesalahan dalam proses, mengembalikan pesan error yang menjelaskan kesalahan.

Hasil dari penerapan algoritma *voice assistant* serta penerapan REST API OpenAI kemudian diimplementasikan ke dalam bahasa pemrograman. Berikut ini adalah hasil dari pelaksanaan implementasi tersebut ke dalam bahasa pemrograman.



Gambar 7. Tampilan Awal Aplikasi *Voice Assistant*

Gambar 7. merupakan tampilan awal pada aplikasi *voice assistant* yang telah dipasang pada *smartwatch*, dimana pada tampilan awal ini terdapat *button* klik *microphone* yang jika akan akan menggunakannya klik tombol *button* tersebut maka sistem akan melakukan *recognition* dari suara yang pengguna ucapkan



Gambar 8. Hasil *Prompt* Dari Pengguna

Pada Gambar 8. di atas merupakan hasil sistem setelah melakukan pengenalan suara pada saat digunakan pada perangkat *smartwach* dan telah dilakukan pertanyaan (Selamat malam) dan hasil respon seperti tampilan pada gambar diatas

4) *Testing*

Pada tahapan terakhir yaitu *testing*, dimana pada tahapan ini dilakukan pengujian agar dapat menemukan *bug*, *error* dan kesalahan lainnya sehingga mengakibatkan sistem tidak berjalan. Untuk metode yang digunakan dalam pembuatan aplikasi ini adalah menggunakan *alpha testing* yaitu menggunakan *blockbox testing*, metode pengujian *blackbox testing* digunakan untuk menguji fungsionalitas fitur. Menurut [13], beberapa definisi *black box testing* adalah sebagai berikut: pengujian dilakukan berdasarkan apa yang dilihat, fokus pada fungsionalitas dan output perangkat lunak; pengujian dilakukan oleh penguji yang independen, tidak memerlukan keahlian khusus dalam *programming* atau penulisan kode program dan *black box testing* dilakukan setelah *white box test* selesai. dan *beta testing* di lakukan *usability testing* kepengguna, digunakan untuk mengevaluasi seberapa puas pengguna dengan menggunakan aplikasi, teknologi, atau produk untuk memenuhi tujuan [14]. Dimana untuk hasil pengujiannya ditampilkan sebagai berikut.

- a. Tahapan ini digunakan untuk pengujian fungsionalitas aplikasi. Pengujian *blackbox testing* ini apakah sesuai dengan analisis kebutuhan apakah sudah berjalan dengan lancar, pada pengujian ini masih menggunakan perangkat smartwatch Samsung Galaxy Watch 4. Pengujian ini di sajikan dalam tabel berikut.

Tabel 1. Hasil dari aktivitas pengujian *Blackbox Testing*

No	Aktivitas	Kelas Uji	Skenario Uji	Hasil yang diharapkan	Keterangan
1	<i>Input</i> suara	Menjalankan <i>Microphone</i>	Melakukan <i>klik</i> pada tombol <i>Play Micropone</i>	Tombol <i>play</i> dapat <i>running</i>	Sesuai
		Melakukan <i>Speech Recognition</i>	Melakukan percakapan perintah	Dapat merekam suara dari pengguna	
		Mendapatkan hasil <i>speech to text</i>	Memproses dari suara menghasilkan data text	Menghasilkan data <i>text</i>	
		Memberhentikan perekaman suara	Memberhentikan <i>running</i> dari <i>play microphone</i>	Dapat berhenti	
		Mendapatkan data <i>text</i> dan	Data <i>text</i> di kirim ke <i>engine</i> openai melalui RestAPIOpenAI	Dapat di kirim	
2	Prosos <i>Engine</i> OpenAi	Memproses data	Memproses data <i>text</i>	Dapat di proses	Sesuai
		Memberikan hasil data	Dapat di ambil		
3	Menampilkan <i>text</i>	Menampilkan <i>text</i>	Mendapatkan data <i>text</i> dari <i>engine</i> openai untuk di tampilan hasil perintah pengguna	Dapat di lihat dari aplikasi	Sesuai
4	<i>Output</i> Suara	<i>Output</i> Suara	Mengeluarkan suara bahasa dialek sesuai dengan <i>text</i>	Dapat di dengar suara berupa dialek	Sesuai

- b. Tahapan kedua *Usability testing* peneliti menggunakan metode *Measuring Usability with the USE Questionnaire* untuk mengukur kemudahan penggunaan (*usability*) suatu produk atau aplikasi melalui penggunaan kuesioner khusus yang mencakup tiga dimensi utama: Kemanafaatan (*Usefulness*), Kepuasan (*Satisfaction*), dan Kemudahan Penggunaan (*Ease of Use*). Untuk sistem internal, item-item yang berkontribusi pada *Ease of Use* untuk produk lain sebenarnya dapat dipisahkan menjadi dua faktor, *Ease of Learning* dan *Ease of Use* [15]. Penguji akan mengawasi pengguna saat menggunakan aplikasi untuk menyelesaikan tugas skenario. Jumlah pengguna yang dibutuhkan untuk pengujian bergantung pada skala proyek: proyek kecil membutuhkan minimal 5 pengguna, proyek besar membutuhkan 15 pengguna. *Testing* kemampuan dapat dilakukan dimanapun peserta berada [16]. Untuk menilai aspek *usability testing*, kriteria dari terendah 1 hingga

tertinggi 5 digunakan sebagai acuan untuk menilai persentase. Pengambilan sampling non probabilitas atau sampel non acak. Pengambilan Sampel Bertujuan Purposive Sampling artinya teknik pengambilan sampel yang cukup sering digunakan. Metode ini menggunakan kriteria yang telah dipilih peneliti untuk menentukan sampel. Kriteria pemilihan sampel dibagi menjadi kriteria inklusi dan eksklusi. Kriteria inklusi merupakan kriteria sampel yang diinginkan peneliti sesuai dengan tujuan penelitian. Sedangkan kriteria eksklusi merupakan kriteria khusus yang mengakibatkan calon responden yang memenuhi kriteria inklusi harus dikeluarkan dari kelompok penelitian. Misalnya calon responden menggunakan smartwatch atau voice assistant yang dapat mempengaruhi hasil penelitian. Model purposive sampling: penelitian mengenai voice assistant pada smartwatch. Kriteria inklusi yang digunakan meliputi pengguna masyarakat atau ahli yang tau ke teknologi usia 15-64 tahun mampu membaca dan menulis [17].

Tabel 2. Kriteria *Usability Testing*

Kriteria	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Diberikan kepada responden, kuesioner terdiri dari berbagai pertanyaan yang memenuhi kriteria pengujian *usability*. Setiap elemen yang dinilai kemudian dinilai dalam proporsi berdasarkan total skor dibagi skor maksimal di kali persentase.

Tabel 3. Pengujian *Usability Testing* Oleh Pengguna

Responden	Pertanyaan										Tortal Skor	Skor Maksimal	
	Usefulness					Ease of use		Ease of learning		Satisfaction			
	1	2	3	4	5	1	2	1	2	1			2
1	5	5	4	4	5	5	4	4	5	5	46	50	
2	4	4	4	4	4	4	5	4	4	4	41	50	
3	5	4	4	4	5	4	5	5	5	5	46	50	
4	5	5	5	5	5	5	5	5	5	5	50	50	
5	4	5	4	4	5	4	5	4	5	5	45	50	
6	4	4	4	3	4	3	4	3	4	4	37	50	
7	5	5	5	5	5	5	5	4	5	5	49	50	
8	5	5	5	5	5	5	5	5	5	5	50	50	
9	4	4	4	5	4	3	4	4	4	4	40	50	
10	4	5	4	4	5	5	5	4	5	5	46	50	
11	4	4	5	4	4	3	4	3	4	4	39	50	
12	5	5	5	5	5	5	5	5	5	5	50	50	
13	4	4	4	4	4	4	4	4	3	4	39	50	
14	5	4	5	5	5	3	5	5	5	5	47	50	
15	5	4	5	4	4	5	4	5	4	4	44	50	
	Total										634	750	

Berdasarkan hasil pengujian usability, yang dapat dilihat pada tabel 3. di atas, dengan total skor 669, hasil pengujian usability dapat dihitung sebagai berikut.

$$\text{Presentase indeks} = \frac{\text{Total Skor}}{\text{Skor Maksimal}} \times 100\% = \frac{634}{750} \times 100\% = 84.53 \%$$

Berdasarkan hasil pengujian usability, mendapatkan nilai

Indeks 0% - 19,99% : Sangat Tidak Setuju

Indeks 20% - 39,99% : Tidak Setuju

Indeks 40% - 59,99% : Netral

Indeks 60% - 79,99% : Setuju

Indeks 80% - 89,99% : Sangat Setuju

Nilai indeks yang kita dapatkan dari perhitungan adalah 84.53%, maka dapat disimpulkan bahwa responden “SANGAT SETUJU”, *testing* aplikasi berhasil.

3.2 Pembahasan Hasil

Hasil penelitian adalah aplikasi *voice assistant* pada *smartwach* menggunakan *open artificial intelligence* yang bertujuan membantu memudahkan pengguna dalam mendapatkan *respon* jawaban dari pertanyaan *random* dengan respon jawaban yang sesuai hasil dari proses *open artificial intelligence*, kemudian memberikan peningkatan kinerja kerja pengguna dimana *voice assistant* diterapkan pada *smartwatch* dengan system operasi wear os. Dengan adanya aplikasi yang dibuat ini juga dapat memberikan peningkatan kinerja kerja pengguna *voice assistant* dimana sebagai *virtual assistant* yang diterapkan pada *smartwatch* dengan system operasi wear os versi one UI dengan perangkat Samsung galaxy watch 4.

Berdasarkan penelitian ini dapat menjawab permasalahan bahwa *voice assistant* tidak bisa menanggapi perintah untuk menjawab pertanyaan dari pengguna secara langsung, kemudian tidak dapat berinteraksi secara dialek dengan beberapa bahasa untuk tanggapan yang ditanyakan, dan *voice assistant* ini pada perangkat *smartwatch* memberikan tingkat kepercayaan seseorang bahwa penggunaan sebuah *system* yang khusus akan meningkatkan kinerja (*perceived usefulness*).

4. KESIMPULAN

Berdasarkan hasil dan pembahasan, penelitian ini menghasilkan aplikasi *voice assistant* yang mampu memahami dan merespons perintah suara pengguna dengan baik. Aplikasi ini diharapkan dapat memberikan fitur-fitur yang memudahkan pengguna seperti mendapatkan informasi dan saran, jawaban terhadap pertanyaan kesehatan, metode pembelajaran, budidaya tanaman, saran pemilihan keputusan, dan pertanyaan umum lainnya. Penelitian ini juga memperhatikan aspek kinerja dan kehandalan aplikasi. Pengujian dilakukan untuk memastikan bahwa aplikasi dapat berjalan dengan baik pada perangkat *smartwatch*, memberikan respons, dan mengatasi berbagai tantangan yang mungkin timbul.

Pada penelitian ini pengguna *smartwatch* dapat memanfaatkannya sebagai alat bantu yang berguna untuk berinteraksi dengan perangkat melalui perintah suara, yang memungkinkan pengguna untuk dengan cepat mengakses informasi dengan mudah. Manfaat yang dirasakan oleh pengguna, karena dapat memberikan pengalaman pengguna yang lebih efektif, efisien, dan praktis dalam berinteraksi dengan perangkat *smartwatch*. Namun kekurangan dari penelitian ini pengguna belum dapat mengakses dan mengontrol berbagai fitur dan aplikasi di perangkat *smartwatch*, seperti menjawab pesan teks, memutar musik, mengatur alarm.

REFERENSI

- [1] F. Hartono, R. Lim, and L. P. Dewi, “Pembuatan Sistem Rumah Pintar dengan Voice Assistant di Raspberry Pi,” Surabaya, 2020.
- [2] I. Supiandi, “Analisis Digital Assistant versi Cortana, Siri dan Google Now,” *Infotech Journal*, 2016.
- [3] D. Sari, H. R. Yuliharto, K. Kunci, A. Faktor, T. Pintar, and P. Berkelanjutan, “Faktor-faktor yang Memotivasi Penggunaan Berkelanjutan Jam Tangan Pintar di Indonesia,” 2020.
- [4] Dr. Dra. H. S. H. M. S. dan A. O. S. Rita Wahyu Meganingrum, “Efektivitas Pemanfaatan Informasi Google Assistant pada Smartphone Android Terhadap Pemenuhan Informasi bagi Mahasiswa Universitas Bhayangkara Jakarta Raya,” *Jurnal Daring Mahasiswa Komunikasi*, vol. 1, no. 2, 2021.
- [5] D. Kurniadi, F. Nuraeni, I. T. Raharja, A. Mulyani, and P. Korespondensi, “Perancangan Aplikasi Text To Speech Dalam Bahasa Indonesia Menggunakan Firebase Machine Learning Kit Berbasis Android Text To Speech Application Design In Indonesian Language Using Firebase Machine Learning Kit Based on Android,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 6, pp. 1281–1288, Dec. 2022, doi: 10.25126/jtiik.202295985.
- [6] D. I. Mulyana and I. Baihaqi, “Rancang Bangun Aplikasi Speech To Text Berbasis Android Pada Elevenia Selection and peer-review under responsibility of The 11th STIKOM CKI on SPOT,” *CKI On SPOT*, vol. 11, no. 1, 2018.

- [7] I. F. Rizal, I. Wayan, A. Arimbawa, and R. Afwani, "Rancang Bangun Digital Home Assistant dengan Perintah Suara Menggunakan Raspberry Pi dan Smartphone (Design and Built Digital Home Assistant with Voice Commands Using Raspberry Pi and Smartphone)," Lombok, Dec. 2018. [Online]. Available: <http://jcosine.if.unram.ac.id/>
- [8] S. Haryanto and M. Ramdhani Raharjo, "Rancangan Bangun Smart Home Berbasis IoT Menggunakan Konsep IFTTT (If This Then That) dengan ESP8266 dan Google Assistant," kalimantan, Jun. 2021.
- [9] M. S. H. A. Mega maulana Setiawan, "Naskah Publikasi Tugas Akhir_Mega Maulana Setiawan_515071114," *NASKAH PUBLIKASI*, 2019.
- [10] Eri Bayu Pratama, "Pendekatan Metodologi Extreme Programming pada Aplikasi e-Commerce Berbasis M-Commerce Studi Kasus: Toko Buku An'Nur di Pontianak," *jurnal Khatulistiwa informatika*, vol. 5, no. 2, 2017, [Online]. Available: www.kaskus.co.id.
- [11] A. Setiawan and U. K. Luthfiyani, "Penggunaan ChatGPT Untuk Pendidikan di Era Education 4.0: Usulan Inovasi Meningkatkan Keterampilan Menulis," *Jurnal PETISI*, vol. 04, no. 01, 2023, [Online]. Available: <https://chat.openai.com>.
- [12] N. A. Septiani and F. Y. Habibie, "Penggunaan Metode Extreme Programming Pada Perancangan Sistem Informasi Pelayanan Publik," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 3, no. 3, p. 341, Mar. 2022, doi: 10.30865/json.v3i3.3931.
- [13] H. Nurfauziah and I. Jamaliyah, "Perbandingan Metode Testing Antara Blackbox Dengan Whitebox Pada Sebuah Sistem Informasi," vol. 8, no. 2, 2022.
- [14] A. Qashlim, T. Prahasto, R. Gernowo, U. Al, A. Mandar, and S. Barat, "Evaluasi Human Machine Interface Menggunakan Kriteria Usability Pada Sistem E-learning Perguruan Tinggi," 2014.
- [15] A. Lund and 94publikasi 767kutipan, "Mengukur Kegunaan dengan Kuesioner USE," 2001. [Online]. Available: <https://www.researchgate.net/publication/230786746>
- [16] J. Sains, D. Teknologi, P. Sukmasetya, A. Setiawan, and E. R. Arumi, "Penggunaan Usability Testing Sebagai Alat Evaluasi Website Krs Online Pada Perguruan Tinggi".
- [17] physipol, "Sampling Techniques in Research," PDAI Medan Area University.