**IMPLEMENTASI *CHATBOT* DENGAN MEMANFAATKAN *API CHATGPT* MELALUI *GRADIO APP* MENGGUNAKAN METODE *NATURAL LANGUAGE PROCESSING (NLP)***

SKRIPSI



**OLEH:**

**MUHAMMAD IRGI AL GHITHRAF**

**201011400874**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**TANGERANG SELATAN**

**2023**

**IMPLEMENTASI *CHATBOT* DENGAN MEMANFAATKAN *API CHATGPT* MELALUI *GRADIO APP* MENGGUNAKAN METODE *NATURAL LANGUAGE PROCESSING (NLP)***

**SKRIPSI**

Diajukan Untuk Melengkapi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer



**OLEH:**

**MUHAMMAD IRGI AL GHITHRAF**

**201011400874**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**TANGERANG SELATAN**

**2023/2024**

## 



FAKULTAS ILMU KOMPUTER

PROGRAM STUDI TEKNIK INFORMATIKA

## LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : MUHAMMAD IRGI AL GHITHRAF

NIM : 201011400874

Program Studi : Teknik Informatika

Fakultas : Ilmu Komputer

Jenjang Pendidikan : Strata 1

Menyatakan bahwa skripsi yang saya buat dengan judul:

IMPLEMENTASI *CHATBOT* DENGAN MEMANFAATKAN *API CHATGPT* MELALUI *GRADIO APP* MENGGUNAKAN METODE *NATURAL LANGUAGE PROCESSING (NLP).*

1. Merupakan hasil karya tulis ilmiah sendiri, bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya ijinkan untuk dikelola oleh Universitas Pamulang sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai aturan yang berlaku apabila di kemudian hari pernyataan ini tidak benar.

Tangerang Selatan, 2024

Meterai 10000 IDR



Muhammad Irgi Al Ghithraf

FAKULTAS ILMU KOMPUTER

PROGRAM STUDI TEKNIK INFORMATIKA

## LEMBAR PERSETUJUAN

NIM : 201011400874

Nama : MUHAMMAD IRGI AL GHITHRAF

Program Studi : TEKNIK INFORMATIKA

Fakultas : ILMU KOMPUTER

Jenjang Pendidikan : STRATA 1

Judul Skripsi : IMPLEMENTASI *CHATBOT* DENGAN MEMANFAATKAN *API CHATGPT* MELALUI *GRADIO APP* MENGGUNAKAN METODE *NATURAL LANGUAGE PROCESSING (NLP)*

Skripsi ini telah diperiksa dan disetujui oleh pembimbing untuk persyaratan sidang skripsi

Tangerang Selatan, .......................2024

Pembimbing

|  |
| --- |
| Dr. Thoyyibah T, S.Kom., M.Kom. |
| NIDN : 0423058704 |

Mengetahui,

Ketua Program Studi

|  |
| --- |
| Achmad Udin Zailani, S.Kom., M.Kom. |
| NIDN : 0429058303 |



FAKULTAS ILMU KOMPUTER

PROGRAM STUDI TEKNIK INFORMATIKA

## LEMBAR PENGESAHAN

NIM : 201011400874

Nama : MUHAMMAD IRGI AL GHITHRAF

Program Studi : TEKNIK INFORMATIKA

Fakultas : ILMU KOMPUTER

Jenjang Pendidikan : STRATA 1

Judul Skripsi : IMPLEMENTASI *CHATBOT* DENGAN MEMANFAATKAN *API CHATGPT* MELALUI *GRADIO APP* MENGGUNAKAN METODE *NATURAL LANGUAGE PROCESSING (NLP)*

Skripsi ini telah dipertahankan di hadapan dewan penguji ujian skripsi fakultas Ilmu Komputer, program studi Teknik Informatika dan dinyatakan LULUS.

Tangerang Selatan, .......................2024

Penguji I Penguji II

|  |  |
| --- | --- |
| Nama Penguji I, S.Kom., M.Kom. | Nama Penguji II, S,Kom.,M.Kom |
| NIDN : | NIDN : |

|  |
| --- |
| Pembimbing |
|  |
|  |
|  |
| Dr. Thoyyibah T, S.Kom., M.Kom. |
| NIDN : 0423058704 |
| Mengetahui,  Ketua Program Studi | | |
|  | | |
|  | | |
|  | | |
| Achmad Udin Zailani, S.Kom., M.Kom. | | |
| NIDN : 0429058303 | | |

## *ABSTRACT*

## ABSTRAK

## KATA PENGANTAR

Assalamu’alaikum warahmatullahi wabarakatuh

Alhadulillah, Puji syukur panjatkan kehadirat Allah Yang Maha Esa, atas berkat, Kesehatan, Rahmat dan Karunia-Nya, Penulis dapat menyelesaikan skripsi ini dengan sebaik-baiknya dan di berikan kemudahan dalam pengerjaannya dengan judul “Implementasi *Chatbot* dengan Memanfaatkan *API ChatGPT* melalui *Gradio App* Menggunakan Metode *Natural Language Processing (NLP)*” ini dapat diselesaikan. Penulisan ini dilaksanakan penulis sebagai salah satu syarat kelulusan Strata Satu (S1) pada Program Studi Teknik Informatika Universitas Pamulang, Tangerang Selatan.

Dalam penyelesaian Skripsi ini, Penulis telah berusaha dengan sebaik mungkin dan telah berusaha dengan kemampuan penulis dan tentunya tidak lupa pula dengan bantuan seluruh pihak yang telah meluangkan waktunya untuk membantu kelancaran penulisan Skripsi ini. Maka dalam kesempatan ini Penulis memberikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan begitu banyak nikmat dan karunia diantaranya Iman dan Islam serta sehat dan umur panjang sehingga dapat menyelesaikan skripsi ini.
2. Bapak Dr. Pranoto, MM Selaku Ketua Yayasan Sasmita Jaya.
3. Bapak Dr. Drs. E. Nurzaman AM., M.M., M.Si., selaku Rektor Universitas Pamulang.
4. Bapak Dr. Ir. H. Sarwani, M.T., M.M, selaku Dekan Fakultas Ilmu Komputer Universitas Pamulang.
5. Bapak Achmad Udin Zailani, S. Kom., M. Kom., selaku Ketua Program Studi Teknik Informatika Universitas Pamulang.
6. Ibu Dr. Thoyyibah T, S. Kom., M. Kom, selaku Pembimbing Skripsi Pada Program Studi Teknik Informatika Universitas Pamulang.
7. Kedua Orang Tua dan Adik saya tercinta yang telah memberikan doa, kasih sayang, semangat, juga motivasi kepada penulis untuk berbagi hal dan khususnya dalam penyusunan dalam penyusunan skripsi ini.
8. Teman – teman seperjuangan kelas 08TPLP008 Universitas Pamulang yang telah memberikan dukungandan semangatnya dalam pembuatan skripsi ini.
9. Seluruh pihak-pihak yang telah mendukung penulis baik secara langsung maupun tidak langsung.
10. ” Membaca bukanlah tujuan akhir, melainkan alat untuk mencapai tujuan” – *Adolf Hitler*.

Penulis memahami bahwa masih banyak kekurangan dalam skripsi ini, maka dari itu penulis sangat mengharapkan kritik dan saran yang bersifat membangun untuk karya yang lebih baik ke depannya. Akhir kata, penulis berharap semoga skripsi ini dapat bermanfaat bagi semuanya, dan dapat dimanfaatkan dengan semestinya. Semoga Allah SWT membalas kebaikan dan selalu mencurahkan hidayahnya serta taufik-Nya, Amin.

Wassalamu’aikum warahmatullahi wabarakatuh

|  |
| --- |
| Tangerang Selatan,……..2024 |
| Hormat saya |
|  |
| (Muhammad Irgi Al Ghithraf) |

## DAFTAR ISI

[LEMBAR PERNYATAAN ii](#_Toc168160298)

[LEMBAR PERSETUJUAN iii](#_Toc168160299)

[LEMBAR PENGESAHAN iv](#_Toc168160300)

[*ABSTRACT* v](#_Toc168160301)

[ABSTRAK vi](#_Toc168160302)

[KATA PENGANTAR vii](#_Toc168160303)

[DAFTAR ISI ix](#_Toc168160304)

[DAFTAR GAMBAR xii](#_Toc168160305)

[DAFTAR TABEL xiv](#_Toc168160306)

[BAB I PENDAHULUAN 1](#_Toc168160307)

[1.1 Latar Belakang 1](#_Toc168160308)

[1.2 Identifikasi Masalah 2](#_Toc168160309)

[1.3 Rumusan Masalah 2](#_Toc168160310)

[1.4 Batasan Penelitian 3](#_Toc168160311)

[1.5 Tujuan Penelitian 3](#_Toc168160312)

[1.6 Manfaat Penelitian 4](#_Toc168160313)

[1.6.1 Manfaat Bagi Penulis 4](#_Toc168160314)

[1.6.2 Manfaat Bagi Pengguna 4](#_Toc168160315)

[1.7 Metodelogi Penelitian 4](#_Toc168160316)

[1.7.1 Metode Pengumpulan Data 4](#_Toc168160317)

[1.7.2 Metode Perancangan Sistem 5](#_Toc168160318)

[1.8 Sistematika Penulisan 6](#_Toc168160319)

[BAB II LANDASAN TEORI 7](#_Toc168160320)

[2.1 Penelitian Terkait 7](#_Toc168160321)

[2.2 Teori Umum 11](#_Toc168160322)

[2.2.1 Kecerdasan Buatan *(Artificial Intelligence)* 11](#_Toc168160323)

[2.2.2 *Chatbot* 12](#_Toc168160324)

[2.2.3 *Application Programming Interface (API)* 13](#_Toc168160325)

[2.2.4 *Natural Language Processing (NLP)* 15](#_Toc168160326)

[2.2.5 *Open AI ChatGPT* 17](#_Toc168160327)

[2.2.6 *Gradio App* 18](#_Toc168160328)

[2.2.7 Basis Data (*Database*) 18](#_Toc168160329)

[2.2.8 *MySQL* 19](#_Toc168160330)

[2.3 *Flowchart Diagram* 20](#_Toc168160331)

[2.4 *Unified Modelling Language (UML)* 21](#_Toc168160332)

[2.4.1 *Use Case Diagram* 21](#_Toc168160333)

[2.4.2 *Activity Diagram* 24](#_Toc168160334)

[2.4.3 *Sequence Diagram* 26](#_Toc168160335)

[2.5 Pengujian Sistem 28](#_Toc168160336)

[2.5.1 Tujuan Pengujian 28](#_Toc168160337)

[2.5.2 Pengujian *Black Box Testing* 28](#_Toc168160338)

[2.5.3 Pengujian *White Box Testing* 28](#_Toc168160339)

[2.6 Perangkat Lunak Pendukung 28](#_Toc168160340)

[2.6.1 Bahasa Pemrograman *Python* 29](#_Toc168160341)

[2.6.2 *Visual Studio Code (VS Code)* 29](#_Toc168160342)

[2.6.3 *Uptime Robot* 30](#_Toc168160343)

[2.6.4 *Draw.io* 30](#_Toc168160344)

[2.6.5 *Figma* 30](#_Toc168160345)

[2.6.6 PhpMyAdmin 31](#_Toc168160346)

[2.6.7 *XAMPP* 32](#_Toc168160347)

[BAB III ANALISA DAN PERANCANGAN 24](#_Toc168160348)

[3.1 Analisa Kebutuhan 24](#_Toc168160349)

[3.2 Analisa Sistem 24](#_Toc168160350)

[3.2.1 Analisa Sistem Berjalan 24](#_Toc168160351)

[3.2.2 Analisa Sistem Usulan 26](#_Toc168160352)

[3.3 Perancangan *Database* 27](#_Toc168160353)

[3.3.1 Spesifikasi *Database* 27](#_Toc168160354)

[3.4 Perancangan *Unified Modelling Language (UML)* 28](#_Toc168160355)

[3.4.1 *Use Case* Diagram *Chatbot (ALICE)* 28](#_Toc168160356)

[3.4.2 *Activity Diagram* Berjalan *Chatbot (ALICE)* 29](#_Toc168160357)

[3.4.3 *Activity Diagram* Usulan *Chatbot (ALICE)* 30](#_Toc168160358)

[3.4.4 *Activity Diagram* Penerjemah Bahasa Asing *(Translate)* 32](#_Toc168160359)

[3.4.5 *Activity Diagram* Paraprase 34](#_Toc168160360)

[3.4.6 *Activity Diagram* Analisis Sentimen 36](#_Toc168160361)

[3.4.7 *Activity Diagram* Pemilihan *Dataset* 38](#_Toc168160362)

[3.4.8 *Sequence Diagram Chatbot (ALICE)* 40](#_Toc168160363)

[3.5 Perancangan Antarmuka *(User Interface)* 41](#_Toc168160364)

[BAB IV IMPLEMENTASI DAN PENGUJIAN 43](#_Toc168160365)

[4.1 Spesifikasi 43](#_Toc168160366)

[4.1.1 Spesifikasi Perangkat Lunak 43](#_Toc168160367)

[4.1.2 Spesifikasi Perangkat Keras 44](#_Toc168160368)

[4.2 Implementasi Program 45](#_Toc168160369)

[4.2.1 Halaman *Chatbot (ALICE)* 45](#_Toc168160370)

[4.3 Implementasi Program 47](#_Toc168160371)

[4.3.1 Sistem *Black Box* 49](#_Toc168160372)

[4.3.2 Sistem *White Box* 53](#_Toc168160373)

[4.4 *User Response* (Kuesioner) 57](#_Toc168160374)

[BAB V PENUTUP 34](#_Toc168160375)

[5.1 Kesimpulan 34](#_Toc168160376)

[5.2 Saran 34](#_Toc168160377)

[DAFTAR PUSTAKA 7](#_Toc168160378)

## DAFTAR GAMBAR

[Gambar 2.1 Kecerdasan Buatan *(Artficial Intelligence)* 11](#_Toc167795148)

[Gambar 2.2 *Chatbot* 12](#_Toc167795149)

[Gambar 2.3 *Aplication Programming Interface (API)* 13](#_Toc167795150)

[Gambar 2.4 *Natural Language Processing (NLP)* 15](#_Toc167795151)

[Gambar 2.5 *OpenAI ChatGPT* 17](#_Toc167795152)

[Gambar 2.6 *Gradio App* 18](#_Toc167795153)

[Gambar 2.7 Basis Data *(Database)* 18](#_Toc167795154)

[Gambar 2.8 *MySQL* 19](#_Toc167795155)

[Gambar 2.9 *Logo Python* 29](#_Toc167795156)

[Gambar 2.10 *Logo Visual Studio Code (VS Code)* 29](#_Toc167795157)

[Gambar 2.11 *Draw.io* 30](#_Toc167795158)

[Gambar 2.12 *Figma* 30](#_Toc167795159)

[Gambar 2.13 *PhpMyAdmin* 31](#_Toc167795160)

[Gambar 2.14 *XAMPP* 32](#_Toc167795161)

[Gambar 3.1 Analisa Sistem Berjalan *Chatbot (ALICE)* 17](#_Toc168131472)

[Gambar 3.2 Analisa Sistem Usulan *Chatbot (ALICE)* 18](#_Toc168131473)

[Gambar 3.3 *Use Case* *Diagram* *Chatbot (ALICE)* 20](#_Toc168131474)

[Gambar 3.4 *Activity Diagram* Berjalan *Chatbot (ALICE)* 21](#_Toc168131475)

[Gambar 3.5 *Activity Diagram* Usulan *Chatbot (ALICE)* 22](#_Toc168131476)

[Gambar 3.6 *Activity Diagram* Penerjemah Bahasa *(Translate) Chatbot (ALICE)* 24](#_Toc168131477)

[Gambar 3.7 *Activity Diagram* Paraprase *Chatbot (ALICE)* 26](#_Toc168131478)

[Gambar 3.8 *Activity Diagram* Analisis Sentimen *Chatbot (ALICE)* 28](#_Toc168131479)

[Gambar 3.9 *Sequence Diagram Chatbot (ALICE)* 30](#_Toc168131480)

[Gambar 3.10 Rancangan Tampilan Halaman *Web Browser Chatbot (ALICE)* pada Perangkat *Laptop* 31](#_Toc168131481)

[Gambar 3.11 Rancangan Tampilan Halaman *Web Browser Chatbot (ALICE)* pada Perangkat *Smartphone* 32](#_Toc168131482)

[Gambar 4.1 Halaman *Web Browser Chatbot (ALICE)* pada Perangkat *Laptop* 26](#_Toc167795170)

[Gambar 4.2 Halaman *Web Browser* *Chatbot (ALICE)* pada Perangkat *Smartphone* 27](#_Toc167795171)

[Gambar 4.3 Halaman *Chatbot (ALICE) Demo* pada Perangkat *Laptop* 28](#_Toc167795172)

[Gambar 4.4 Tampilan *Form* Pertanyaan pada *Chatbot (ALICE) Demo* pada Perangkat *Laptop* 28](#_Toc167795173)

[Gambar 4.5 Tampilan *Form* Jawaban pada *Chatbot (ALICE) Demo* pada Perangkat *Laptop* 29](#_Toc167795174)

[Gambar 4.6 Tampilan *Form Audio* jawaban pada *Chatbot (ALICE)* 29](#_Toc167795175)

[Gambar 4.7 Tampilan Database Histori Chat 30](#_Toc167795176)

## DAFTAR TABEL

[Tabel 2.1 Penelitian Terkait 7](#_Toc167795178)

[Tabel 2.2 Simbol – Simbol *Flowchart Diagram* 20](#_Toc167795179)

[Tabel 2.3 Simbol – Simbol *Use Case Diagram* 22](#_Toc167795180)

[Tabel 2.4 Simbol – Simbol *Activity Diagram* 24](#_Toc167795181)

[Tabel 2.5 Simbol – Simbol *Sequence Diagram* 26](#_Toc167795182)

[Tabel 3.1 Rincian Spesifikasi Database Riwayat Chat 19](#_Toc167795185)

[Tabel 4.1 Perangkat Lunak 24](#_Toc167795199)

[Tabel 4.2 Perangkat Keras pada *Laptop* 25](#_Toc167795200)

[Tabel 4.3 Perangkat Keras pada *Smartphone* 25](#_Toc167795201)

[Tabel 4.4 Pengujian *Black Box pada Chatbot (ALICE)* 30](#_Toc167795202)

[Tabel 4.5 Modul Integrasi *API Chatgpt* 35](#_Toc167795203)

[Tabel 4.6 Modul Intergrasi *Audio Player Automatic* 37](#_Toc167795204)

[Tabel 4.7 Skala Perhitungan 39](#_Toc167795205)

[Tabel 4.8 Contoh Perhitungan Persentase 39](#_Toc167795206)

## BAB I PENDAHULUAN

### Latar Belakang

Kecerdasan Buatan atau yang lebih dikenal sebagai *AI (Artificial Intelligence)* adalah suatu teknologi di bidang ilmu komputer yang meniru kemampuan kecerdasan manusia dalam komputer. Tujuannya adalah untuk memecahkan berbagai masalah dan melaksanakan tugas-tugas tertentu, baik sebagaimana yang dilakukan oleh manusia atau bahkan dengan performa yang lebih baik dari manusia itu sendiri (Pratama, 2022). Salah satu aspek yang sangat diperhatikan adalah Pemrosesan Bahasa Alami atau *Natural Language Processing (NLP)*, di mana komputer memiliki kemampuan untuk memahami bahasa manusia sebagaimana manusia memahaminya.

Seiring dengan perkembangan teknologi *AI* dan *NLP*, munculnya sebuah arsitektur model yang mampu mengatasi masalah pemrosesan bahasa dengan sangat baik, telah menjadi dasar bagi beberapa model terkemuka dalam pemrosesan bahasa alami. Salah satu implementasi terkemuka adalah *GPT (Transformer Terlatih Generatif) yang dikembangkan oleh OpenAI* (Setiawan, 2023).

Penggunaan *API ChatGPT* dari *OpenAI* untuk membangun *chatbot* yang lebih canggih dan responsif telah menjadi fokus penelitian dan pengembangan. *API ChatGPT* memungkinkan akses langsung ke model *GPT* yang dilatih secara besar-besaran, yang mampu menghasilkan teks yang lebih alami dan kontekstual. Aplikasi *Chatbot* sangat bergantung pada data train yang dibuat atau masukan dari pengembang chatbot saat mengenali dan merespons. Bisa dengan menggunakan metode penandaan untuk setiap pola kalimat untuk meningkatkan akurasi sistem (Chandra, 2021).

Dalam penelitian ini, penulis akan menggunakan *API ChatGPT* dalam implementasi *chatbot* yang terhubung dengan *Gradio app*. *Gradio app* adalah sebuah *framework* yang memfasilitasi pembuatan antarmuka pengguna interaktif tanpa memerlukan pengetahuan mendalam dalam pengembangan antarmuka. Dengan menggabungkan kemampuan *ChatGPT*, *Gradio app*, dan konsep *NLP (Natural Language Processing)* yang sederhana. Judul yang penulis ajukan yang berjudul “**IMPLEMENTASI *CHATBOT* DENGAN MEMANFAATKAN *API CHATGPT* MELALUI *GRADIO APP* MENGGUNAKAN METODE *NATURAL LANGUAGE PROCESSING (NLP)***”. Tujuan utamanya adalah menciptakan sebuah *chatbot* yang mampu memberikan pengalaman berkomunikasi yang lebih alami, efektif, dan responsif bagi pengguna. Penggunaan metode *NLP* *(Natural Language Processing)* nanti akan digunakan untuk memungkinkan *chatbot* memahami bahasa manusia, mengenali makna dalam teks yang dimasukkan oleh pengguna, dan memberikan respon yang sesuai dan kontekstual.

### 1.2 Identifikasi Masalah

Bedasarkan latar belakang di atas, terdapat beberapa masalah yang dapat diidentifikasikan yaitu sebagai berikut:

1. Bagaimana memastikan bahwa respon yang dihasilkan oleh *Chatbot* mampu memenuhi harapan pengguna dan memberikan solusi yang tepat?
2. Bagaimana mengatasi variasi dalam gaya berbahasa dan penggunaan kata yang berbeda-beda antara pengguna?
3. Bagaimana memastikan bahwa pengguna dapat berinteraksi dengan *Chatbot* secara nyaman dan tanpa hambatan yang menghambat?

### 1.3 Rumusan Masalah

Berdasarkan konteks, latar belakang, dan identifikasi masalah yang telah diuraikan sebelumnya, perumusan masalah dalam penelitian ini adalah:

1. Bagaimana mengatasi kendala dalam menghasilkan respon *Chatbot* yang relevan dan akurat terhadap permintaan dan pertanyaan pengguna?
2. Bagaimana melatih *Chatbot* agar mampu menyesuaikan diri dengan beragam ragam bahasa dan gaya ekspresi yang digunakan oleh pengguna?
3. Bagaimana merancang antarmuka *Gradio App* yang mampu memberikan pengalaman pengguna yang optimal dalam berinteraksi dengan *Chatbot*?

### 1.4 Batasan Penelitian

Dengan merujuk pada latar belakang dan hasil pengamatan yang telah dilakukan, beberapa batasan penelitian yang dihadapi dapat diuraikan, antara lain:Top of Form

1. Penelitian ini akan berfokus pada *Chatbot* berbasis teks yang menggunakan *API ChatGPT* untuk menghasilkan respon teks.
2. Metode yang diterapkan dalam penelitian ini ialah menggunakan pendekatan *Natural Language Processing (NLP)* untuk meningkatkan kemampuan *chatbot* dalam memahami dan merespons bahasa manusia dengan lebih baik.
3. Dalam penelitian ini, akan memanfaatkan *Free API ChatGPT 3.5 Turbo*, di mana setiap satu kunci *API* produk memiliki batasan kapasitas hingga 4097 *token* yang diperkirakan jumlah kata akan berkisar antara 2048 hingga 2931 kata dan hanya mampu memberikan respons dalam bentuk teks.

### 1.5 Tujuan Penelitian

Penelitian ini bertujuan untuk mengatasi permasalahan yang telah diidentifikasi dalam implementasi *Chatbot*. Pendekatan yang digunakan adalah memanfaatkan *API ChatGPT* melalui *Gradio App* dengan menerapkan metode *Natural Language Processing (NLP)*. Tujuan penelitian dapat dijabarkan sebagai berikut:

1. Mengimplementasikan *Chatbot* yang mampu memberikan respon yang relevan, akurat, dan kontekstual terhadap berbagai permintaan dan pertanyaan pengguna dengan menggunakan *API ChatGPT* melalui *Gradio App* berbasis *NLP*.
2. Meningkatkan kualitas respon *Chatbot* dengan mengoptimalkan penggunaan memanfaatkan model bahasa berbasis GPT (Generative Pre-trained Transformer) memungkinkan dihasilkannya respons yang tidak hanya lebih tepat tetapi juga sangat informatif.
3. Membuat antarmuka *Gradio App* yang memenuhi kebutuhan pengguna dengan memberikan pengalaman berinteraksi yang intuitif, responsif, dan menyenangkan dengan *chatbot*.

### Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi dan manfaat sebagai berikut:

### Manfaat Bagi Penulis

1. Penulis dapat lebih memahami dan memperluas wawasan dan dapat mengembangkan ilmu yang didapat dari proses perkuliahan yang berlangsung.
2. Penelitian ini merupakan salah satu syarat untuk menyelesaikan program studi Strata 1 (S1).

### 1.6.2 Manfaat Bagi Pengguna

Penulis berharap dari penelitian ini pengguna mendapatkan interaksi dengan baik, solusi yang memadai, dan respon yang relevan akan berkontribusi pada peningkatan kepuasan pengguna terhadap penggunaan *Chatbot*.

### 1.7 Metodelogi Penelitian

Dalam Implementasi *Chatbot* dengan Memanfaatkan *API ChatGPT* melalui *Gradio App* Menggunakan Metode *Natural Language Processing (NLP)*, Penulis menerapkan metode penelitian sebagai berikut:

### Metode Pengumpulan Data

1. Metode Studi Pustaka

Teknik Pengumpulan data dilakukan dengan cara menghimpun dan memeriksa buku-buku referensi serta sumber-sumber yang relevan dengan topik penelitian.

1. Metode Kuesioner

Metode pengumpulan data dilakukan melalui penyediaan kuesioner yang memuat serangkaian pertanyaan atau pernyataan tertulis kepada responden untuk diisi dan dijawab.

1. Metode Observasi

Teknik pengumpulan data dilakukan melalui observasi langsung pada *chatbot* yang telah dibuat sebelumnya, serta pengamatan terkait beberapa permasalahan yang berkaitan dengan kepentingan pembuatan *chatbot* sebagai layanan informasi yang akurat bagi penggunanya.

### Metode Perancangan Sistem

Dalam proses perancangan aplikasi ini, penulis menerapkan Metode *Waterfall*, sebuah pendekatan pengembangan perangkat lunak yang mengikuti langkah-langkah secara berurutan atau sekuensial (Kurniawati, 2021). Tahapan dalam metode *waterfall* meliputi sebagai berikut:

1. Rekayasa Sistem dan Pemodelan Sistem

Tahap ini sangat menekankan pada masalah pengumpulan kebutuhan pengguna pada tingkat sistem dengan mendefinisikan konsep sistem. Rekayasa sistem meliputi pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis.

1. Desain

Proses pembuatan desain tahap awal merancang tampilan *website*, sehingga menghasilkan desain *interface* yang interaktif.

1. Perkodean dan Pengujian

Dibuat program dengan menggunakan bahasa pemrograman *Python* yang mengacu kepada desain yang telah dibuat sehingga siap dijalankan dan diuji.

1. Uji Coba

Setelah program selesai dibuat, maka dilakukan pengujian terhadap sistem tersebut, apakah aplikasi tersebut berjalan sesuai dengan yang diharapkan atau tidak. Apabila masih terdapat kesalahan pada aplikasi maka dilakukan tahapan analisis kembali untuk mengetahui kesalahan yang muncul.

1. Penerapan

Tahap ini bisa dikatakan terakhir dalam pembuatan sebuah aplikasi. Setelah melakukan analisa, desain, dan pengkodean maka aplikasi yang sudah jadi digunakan oleh *user*.

### 1.8 Sistematika Penulisan

Sistematika Penulisan ini dibuat dengan tujuan memberikan gambaran mengenai isi proposal secara singkat, sehingga pembaca lebih mudah untuk memahami.

**BAB I PENDAHULUAN**

Dalam bab ini, akan diuraikan secara singkat tentang latar belakang, identifikasi masalah, rumusan masalah, batasan penelitian, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Pada bab ini, berisi mengenai landasan teori sebagai parameter rujukan untuk terlaksana nya peneltian ini, bab ini juga mencakup studi terkait yang penulis gunakan sebagai referensi.

**BAB III ANALISA DAN PERANCANGAN**

Pada bab ini, akan membahas tentang analisis kebutuhan, *Diagram* Aktivitas, *Diagram* Alur, dan *Diagram* Kasus Pengguna.

**BAB IV IMPLEMENTASI DAN PENGUJIAN**

Pada bab ini, akan membahas tentang implementasi pada *chatbot*, yang meliputi tahapan implementasi pengujian, pengujian menggunakan metode *blackbox*, *whitebox* dan hasil analisis.

**BAB V PENUTUP**

Pada bab ini, terdapat rangkuman dari sistem yang telah dibuat oleh penulis, yang mencakup semua kegiatan yang telah dilakukan, serta saran-saran untuk pengembangan di masa mendatang.

## BAB II LANDASAN TEORI

### 2.1 Penelitian Terkait

Dalam menyusun skripsi ini, penulis terinspirasi dan mereferensi dari penelitian-penelitian terdahulu yang masih berkaitan dengan latar belakang masalah pada skripsi ini. Berikut ini penelitian terdahulu yang berkaitan dengan skripsi, antara lain:

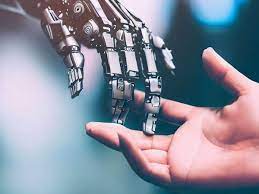
**Tabel 2.1** **Penelitian Terkait**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Judul Penelitian** | **Metode** | **Hasil** | **Kesimpulan** |
| 1. | *Chatbot Telegram* Menggunakan *Natural Language Processing*.  Peneliti: Muhammad Furqan.  Tahun: 2023 | *NATURAL LANGUAGE PROCESSING (NLP)* | *Chatbot* Informasi Mahasiswa Berbasis *Telegram* Pada sistem ini telah berjalannya pengembangan *chatbot* berbasis *telegram* dengan konten berisi informasi mahasiswa, Yang dimana mahasiswa dapat dengan mudah mengetahui Jadwal kelas, Tugas Mata kuliah dan Pengumuman Penting Mahasiswa. | Bot perlu di training terlebih dahulu beberapa kata yang akan digunakan di dalam Chat Bot agar Bot dapat menentukan kata yang memiliki akurasi tertinggi untuk mengirimkan trigger ke server bot. |
| 2. | Sistem Notifikasi Kepangkatan Dosen Secara *Realtime* Berbasiskan *Telegram Bot Application Programming Interface.*  Peneliti: I Ketut Gede Sugita.  Tahun: 2023 | *Research and Development (R&D).* | Sistem Notifikasi Kepangkatan Dosen Secara *Realtime* Berbasiskan *Telegram Bot API* dibuat dengan memperhitungkan 3 entitas utama, mulai dari administrasi, dosen, serta pimpinan yang terlibat memiliki peran masing-masing dalam sistem. | Sistem Notifikasi Kepangkatan Dosen Secara *Realtime* Berbasiskan *Telegram* *Bot Application Programming Interface* menghasilkan sebuah sistem yang mencakup kenaikan pangkat dan menggunakan *Telegram bot* untuk memberikan notifikasi dan sebagai media interaksi dua arah untuk mendapatkan informasi seputar kepegawaian dosen dan kenaikan pangkat. |
| 3. | Pengembangan *Chatbot* Informasi Mahasiswa Berbasis *Telegram* dengan Metode *Natural Language Processing*.  Peneliti: Alfan Adi Chandra.  Tahun: 2021 | *Natural Language Processing (NLP).* | Untuk dapat menjalankan *ChatBot* dengan baik diperlukan *Server* yang baik dan cepat kemudian memerlukan Desain Arsitektur *ChatBot* untuk mengetahui masing masing kegunaan perintah pada *ChatBot*. | Data yang diperlukan untuk dikirimkan kepada pengguna harus di input ke dalam *source code* terlebih dahulu agar bisa mengirimkan pesan sesuai *trigger* yang dikirimkan dari website wit.ai. |
| 4. | Pengujian *Black Box* pada Aplikasi Sistem Seleksi *Sales* Terbaik Menggunakan Teknik *Equivalence Partitions.*  Peneliti: Fadhila Cahya Ningrum.  Tahun: 2019 | *BlackBox* | Aplikasi seleksi sales terbaik ditemukan kesalahan dalam mengedit data di mana tidak sesuai antara pesan yang muncul yaitu data berhasil disimpan sedangkan yang terjadi data tersebut tida berubah. | Pengujian dengan metode *Black Box* berbasis *Equivalence Partitions* dapat membantu proses pembuatan case pengujian, uji kualitas dan menemukan kesalahan yang tidak terdeteksi yang disebabkan oleh kesalahan pengetikan. |
| 5. | Penerapan Media *Chat GPT* pada Pembelajaran Manajemen Pendidikan terhadap Kemandirian Mahasiswa.  Peneliti: Hary Murcahyanto.  Tahun: 2023. | *Pre-experimental* | Hasil penelitian menunjukkan bahwa penggunaan *Chat GPT* pada pembelajaran manajemen pendidikan dapat meningkatkan kemandirian mahasiswa, sehingga dapat menjadi alternatif metode pembelajaran yang efektif dan perlu dilakukan filtrasi dan diarahkan oleh dosen. | Meskipun demikian, penelitian ini memiliki keterbatasan yaitu hanya dilakukan pada satu kelompok sampel yang relatif kecil, sehingga perlu dilakukan penelitian lebih lanjut dengan sampel yang lebih besar dan dari berbagai perguruan tinggi. Penggunaan *Chat GPT* dalam pembelajaran juga memiliki tantangan nilai etika dan orisinalitas karya, sehingga perlu dipertimbangkan dengan kritis. |

### 2.2 Teori Umum

Pada bagian ini akan membahas mengenai semua teori yang digunakan peneliti untuk dasar penulisan dalam mengimplementasikan sebuah *chatbot* dengan memanfaatkan *API ChatGPT* melalui *Gradio App* menggunakan metode *Natural Language Processing (NLP)*.

### 2.2.1 Kecerdasan Buatan *(Artificial Intelligence)*

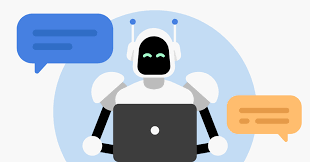


**Gambar 2.1 Kecerdasan Buatan *(Artficial Intelligence)***

Menurut (Kalsum, 2022), Kecerdasan Buatan *(Artificial Intelligence)* adalah suatu inovasi baru dalam bidang ilmu pengetahuan. Kemunculan *AI* dimulai sejak era komputer *modern* pada tahun 1940 dan 1950. Pada awalnya, *AI* dianggap sebagai alat atau komputer yang mampu melaksanakan tugas-tugas yang biasanya dilakukan oleh manusia, sehingga *AI* juga bisa diterapkan dalam konteks permainan.

Menurut (Mardiana, 2024), Kecerdasan Buatan telah menjadi faktor kunci dalam transformasi di berbagai sektor, menghasilkan inovasi dan efisiensi yang luar biasa. Di tengah dominasi teknologi dalam *era* ini, *AI* muncul sebagai kekuatan revolusioner yang menjanjikan dampak yang signifikan. *AI* berkaitan dengan proses berpikir dan penalaran, serta membahas perilaku yang mengukur keberhasilan dalam konteks kinerja manusia yang rasional, ideal, dan berdasarkan pengetahuan yang dimiliki. Dalam definisi ini, suatu sistem dianggap rasional jika mampu melakukan tindakan yang tepat sesuai dengan konteksnya.

### 2.2.2 *Chatbot*

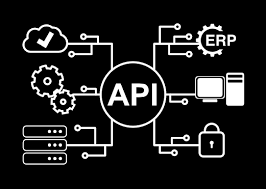


**Gambar 2.2 *Chatbot***

Menurut (Mhd. Furqan, 2023), *Chatbot* adalah implementasi kecerdasan buatan yang mensimulasikan percakapan *real time* antara mesin dan manusia melalui tulisan, suara dan visual. Aplikasi *Chatbot* sangat bergantung pada data train yang dibuat atau masukan dari pengembang *chatbot* saat mengenali dan merespons. Bisa dengan menggunakan metode penandaan untuk setiap pola kalimat untuk meningkatkan akurasi sistem.

Menurut (Nurjanah, 2023), Chatbot merupakan sebuah program yang diciptakan untuk mengotomatiskan aktivitas percakapan yang dilakukan oleh teknologi komputer terhadap manusia. Pengembangan teknologi chatbot sering digunakan pada sistem layanan pelanggan di perusahaan. Terdapat beberapa keuntungan penggunaan chatbot, seperti respon cepat dan ketersediaan yang selalu ada sepanjang waktu, serta biaya operasional yang lebih terjangkau jika dibandingkan dengan menggunakan layanan manusia. Namun, chatbot tidak memiliki tingkat emosi.

### 2.2.3 *Application Programming Interface (API)*



**Gambar 2.3 *Aplication Programming Interface (API)***

Menurut Afriansyah (2021), *Application Programming Interface (API)* adalah antarmuka yang dibuat oleh pengembang sistem untuk memungkinkan akses terprogram terhadap sebagian atau keseluruhan fungsi sistem. *Application Programming Interface (API)* adalah representasi terfokus dari seluruh fungsionalitas dalam suatu modul perangkat lunak yang dapat diakses oleh pihak yang membutuhkan melalui layanan yang telah ditetapkan. Representasi terfokus dari fungsi-fungsi yang dinyatakan dalam *API* bertujuan untuk menyediakan serangkaian layanan yang spesifik untuk tujuan tertentu.

Menurut (Muhammad, 2024), *API* adalah antarmuka yang dirancang pada sistem *server-side (backend)* untuk bertukar data dan memungkinkan interaksi antara sisi *client (frontend)* dan sisi *server (backend)*. *API* dapat digunakan dalam beragam bahasa pemrograman serta dapat beroperasi di berbagai jenis *server* seperti *Apache, NGINX, Tomcat,* dan sebagainya.

*API* terbagi menjadi 4 jenis yang digolongkan menurut hak aksesnya:

* 1. *Public API*

*API* Publik atau dikenal juga sebagai *API* Terbuka adalah jenis *API* yang dapat diakses secara bebas tanpa memerlukan otorisasi khusus. *API* ini kompatibel dengan berbagai sistem operasi dan cukup mudah digunakan setelah pendaftaran. Contoh dari *API* Publik adalah *Open Weather API* untuk memantau kondisi cuaca dan *News API* untuk mendapatkan berita terbaru.

* 1. *Private API*

*API* Privat adalah *API* yang dirancang khusus untuk digunakan oleh tim internal suatu organisasi dalam pengembangan aplikasi. Fungsi utamanya adalah untuk menyediakan data antara *backend* dan *frontend* pada sebuah *website*.

* 1. *Partner API*

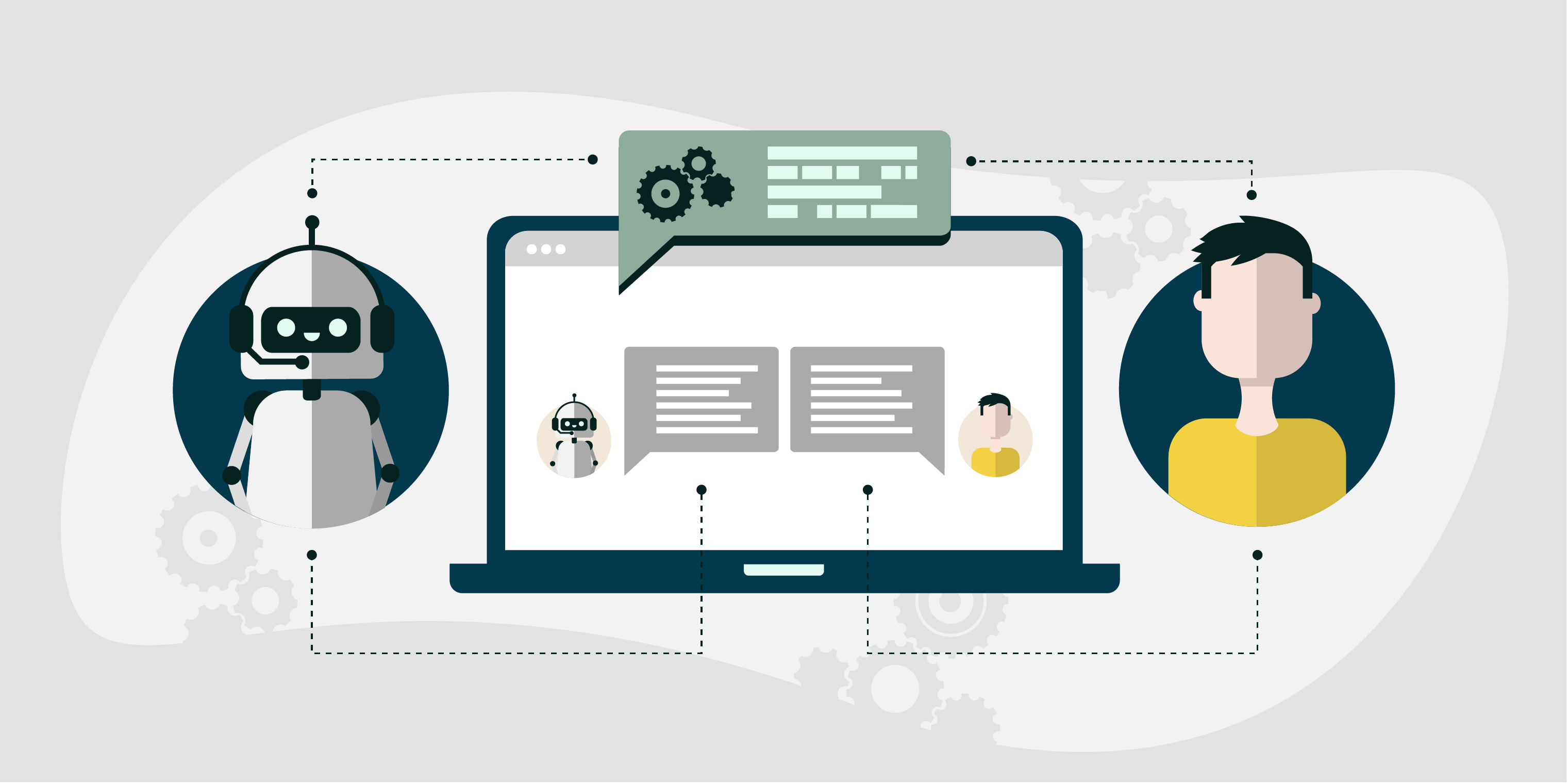
*API* Mitra adalah jenis *API* yang bisa diakses oleh masyarakat umum, namun dengan batasan akses yang telah ditentukan. Untuk menggunakan *API* ini, pengguna harus mendaftar dan mendapatkan izin dari pemilik layanan *API*. Penggunaan *API* Mitra juga hanya diizinkan untuk aplikasi tertentu sesuai dengan syarat dan ketentuan yang berlaku. Contoh dari *API* Mitra adalah *API ChatGPT*.

1. *Composite API*

*API* Komposit adalah *API* yang menggabungkan data dari berbagai sumber atau *server* ke dalam satu tempat. Hal ini bertujuan untuk mempercepat proses akses data oleh pengguna karena mereka dapat mengakses berbagai jenis data dalam satu kunjungan saja.

*ChatGPT OpenAI* merupakan teknologi mesin berbasis kecerdasan buatan yang dilatih untuk bisa menirukan percakapan manusia menggunakan teknologi *NLP (Natural Language Processing)*.

### 2.2.4 *Natural Language Processing (NLP)*



**Gambar 2.4 *Natural Language Processing (NLP)***

Menurut H. Husamuddin (2020), dalam penelitian yang dilakukan menggunakan metode *Natural Language Processing (NLP)*. Metode ini memudahkan pengguna untuk menggunakan sistem ini. Pertama, pengguna memasukkan teks di klien obrolan yang disediakan. Teks dapat berupa jawaban atas pertanyaan atau pernyataan. Selain itu, input teks diteruskan ke sistem *chatbot*, untuk mensimulasi percakapan dengan menggunakan bahasa sehari hari.

Beberapa metode yang umum digunakan dalam *Natural Language Processing (NLP)*:

1. Tokenisasi

Tokenisasi adalah proses memecah teks menjadi unit-unit yang lebih kecil, seperti kata atau frasa. Ini adalah tahap awal dalam pemrosesan teks yang membantu dalam analisis lebih lanjut.

1. *Part-of-Speech Tagging*

Metode ini melibatkan pengidentifikasian bagian dari pidato *(part-of-speech)* dari setiap kata dalam suatu kalimat, seperti kata benda, kata kerja, kata sifat, dan sejenisnya.

1. Model Bahasa dan Pembelajaran Mesin

Pendekatan terbaru menggunakan model bahasa berbasis pembelajaran mesin, seperti *GPT (Generative Pre-trained Transformer)* yang diberi tahu dari korpus teks besar untuk memahami dan menghasilkan teks yang lebih alami.

1. Konversi Teks menjadi Suara

Menggunakan *gTTS* *(Google Text-to-Speech)* untuk mengubah teks jawaban yang dihasilkan oleh model menjadi file suara.

1. Penerjemahan Mesin

Menerjemahkan teks dari satu bahasa ke bahasa lain dengan menggunakan bantuan komputer.

1. Analisis Sentimen

Menentukan sentimen *(positif, negatif, atau netral)* dalam teks dapat digunakan untuk menganalisis pandangan orang terhadap suatu topik.

1. Pengenalan Nama Entitas

Melibatkan mengidentifikasi entitas seperti nama orang, tempat, tanggal, dan sebagainya dalam teks.

### 2.2.5 *Open AI ChatGPT*



**Gambar 2.5 *OpenAI ChatGPT***

Menurut (Setiawan, 2023), *ChatGPT* merupakan mesin cerdas yang dilatih untuk menirukan percakapan manusia dengan menggunakan teknologi Pemrosesan Bahasa Alami atau *Natural Language Processing* *(NLP)*. Dengan menggunakan *ChatGPT*, pengguna dapat menghasilkan tulisan ilmiah yang substansial dengan merumuskan *prompt* awal secara efektif dan baik.

Teknologi tersebut diperkenalkan oleh perusahaan Amerika, *OpenAI*, pada tanggal 30 November 2022. Model tersebut memiliki kemampuan untuk memahami dan merespons *input* pengguna dengan cara yang meniru percakapan manusia, menciptakan interaksi yang lebih alami dan menarik. Selain itu, *ChatGPT* mampu menghasilkan teks dalam berbagai gaya dan format, termasuk artikel berita, surel, dan puisi, menjadikannya sebagai teknologi yang serbaguna dan bermanfaat untuk berbagai aplikasi.

Menurut (George, 2023), *ChatGPT* adalah sebuah model pemrosesan bahasa alami atau *Natural Language Processing (NLP)* yang mengombinasikan *GPT-2*, sebuah model bahasa berbasis *transformer* yang dikembangkan oleh *OpenAI*, dengan teknik pembelajaran terawasi dan penguatan untuk mengolahnya kembali (pendekatan *transfer learning*) pada kelompok pola bahasa besar *GPT-3* yang dikembangkan oleh *OpenAI*. Model ini memungkinkan pengguna berinteraksi secara alami dengan sistem kecerdasan buatan melalui percakapan berbasis teks. *ChatGPT* dapat dimanfaatkan untuk aplikasi layanan pelanggan dan untuk menciptakan *asisten virtual* untuk percakapan suara dan teks. Selain itu, *ChatGPT* juga menyertakan fitur seperti deteksi topik, deteksi emosi, dan kemampuan analisis sentimen untuk membantu pengguna memahami mitra percakapannya dengan lebih baik. Model ini juga mampu menghasilkan beberapa percakapan secara bersamaan untuk menciptakan interaksi yang lebih realistis antara pengguna dan *bot*.

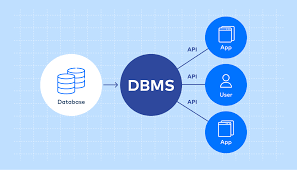
### 2.2.6 *Gradio App*

**

**Gambar 2.6 *Gradio App***

*Gradio App* adalah sebuah platform yang mempermudah pengembang dalam menciptakan antarmuka pengguna *User Interface* *(UI)* interaktif untuk *model* pembelajaran mesin dengan cara yang lebih sederhana dan efisien. Dengan *Gradio App*, pengembang dapat membuat antarmuka yang memungkinkan pengguna berinteraksi langsung dengan model pembelajaran mesin tanpa perlu menulis kode pemrograman yang rumit.

### 2.2.7 Basis Data (*Database*)



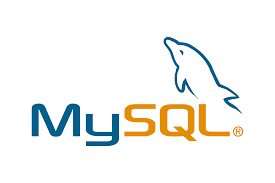
**Gambar 2.7 Basis Data *(Database)***

Menurut (Sitinjak, 2020), Basis Data atau *database* mengacu pada kumpulan tabel yang berisi data-fakta yang menjadi sumber informasi. Data-data tersebut disimpan dalam format digital dan dapat diakses melalui program komputer, memudahkan kegiatan pengambilan informasi. Menurut (Aswiputri, 2021) Hubungan antara *entri* dalam basis data dapat berfungsi sebagai sumber informasi yang berharga bagi pengguna. Hingga saat ini, masih terdapat banyak *record* yang tersedia.

Dalam sistem informasi ini, ketiga komponen ini akan bekerja sama untuk membuat atau menginput data (Simanullang, 2021), yaitu:

* 1. Proses *input* data adalah kegiatan memindahkan *data* ke dalam sistem komputer agar *data* dapat digunakan dan disimpan melalui penggunaan penyimpanan dalam bentuk *file*.
  2. Pemrosesan data adalah proses pengolahan yang mengubah *data* menjadi informasi yang bermanfaat.
  3. Proses *output* menghasilkan produk informasi dan kumpulan *data* yang dikelola menjadi nilai yang menghasilkan informasi.

### 2.2.8 *MySQL*



**Gambar 2.8 *MySQL***

Menurut (Prahast, 2022) *MySQL* merupakan perangkat lunak (*software*) *database* tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan. *MySQL* bekerja menggunakan *SQL* *Language* *(Structure Query Language)*. Itu dapat diartikan bahwa *MySQL* merupakan *standar* penggunaan *database* di dunia untuk pengolahan data. Pada umumnya, perintah yang paling sering digunakan *MySQL* adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus), selain itu, *SQL* juga menyediakan perintah untuk membuat *database*, *field*, atau pun *index* untuk menambah atau menghapus data dapat dijalankan secara langsung dalam *system* operasi.

### 2.3 *Flowchart Diagram*

Menurut (Achlison, 2020) *Flowchart* adalah gambaran grafis dari alur kerja suatu proses dalam suatu sistem. Dibuat untuk memudahkan pemahaman dan penjelasan, *flowchart* menggunakan simbol-simbol khusus yang menggambarkan urutan proses secara mendetail dan hubungan antarproses (instruksi) dalam suatu program.

**Tabel 2.2 Simbol – Simbol *Flowchart Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1. |  | *Arrows* | Penghubung yang menunjukkan hubungan antara bentuk-bentuk representatif. |
| 2. |  | *Start/end* | Oval melambangkan titik awal atau akhir. |
| 3. |  | *Input/Output* | Jajar genjang melambangkan *input* atau *output* (*READ, WRITE*). |
| 4. |  | *Process* | Persegi panjang melambangkan sebuah proses (Menyatakan *assignment* *statement*). |
| 5. |  | *Decision* | Belah ketupat mengindikasikan sebuah keputusan (*True, False*). |

### 2.4 *Unified Modelling Language (UML)*

Menurut (Putra, 2019) *Unified Modeling Language (UML)* merupakan salah satu standar bahasa yang umum digunakan dalam industri untuk mendefinisikan persyaratan, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

### 2.4.1 *Use Case Diagram*

Menurut (Aliman, 2021)*, Use Case* adalah langkah awal dalam pemodelan suatu sistem yang digunakan untuk menggambarkan kebutuhan fungsional sistem. Setiap *Use Case* dijelaskan sebagai skenario tindakan yang dilakukan oleh aktor dalam sistem dan dibatasi oleh batas sistem. Hubungan antar *Use Case* direpresentasikan dengan garis notasi.

**Tabel 2.3 Simbol – Simbol *Use Case Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1. |  | Aktor | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case* diagram |
| 2. |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*Dependency*) *independent* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
| 3. |  | *Association* | Abstraksi dari penghubung antara actor dengan *use case*. |
| 4. |  | *Generalization* | Menunjukan spesialisasi aktor untuk dapat berpartisipasi dengan *use case*. |
| 5. |  | *Extend* | Menspesifikasikan bahwa target *use case* memperluas perilaku dari sumber *use case* pada suatu titik yang diberikan. |
| 6. |  | *Include* | Menspesifikasikan sumber *Use Case* secara eksplisit dapat dilakukan dengan mengidentifikasi aktor atau entitas yang terlibat langsung dalam menjalankan tindakan dalam *Use Case* tersebut. |
| 7. |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8. |  | *Use Case* | Deskripsi dari urutan aksi- aksi yang metapilkan oleh sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |

### 2.4.2 *Activity Diagram*

Menurut (Arianti, 2022), *Activity diagram* adalah representasi grafis yang mengilustrasikan aliran data atau kontrol, aksi terstruktur, dan desain proses dengan baik dalam suatu sistem. *Diagram* ini memvisualisasikan alur kerja dari *use case* yang sedang diproses, dimulai dari titik awal hingga titik akhir. Setiap aktivitas dijelaskan dengan notasi-notasi yang sesuai dengan fungsinya.

**Tabel 2.4** **Simbol – Simbol *Activity Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1. |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| 2. |  | *Activity Final Node* | Bagaimana objek dibentuk dan diakhiri. |
| 3. |  | *Activity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain/ |
| 4. |  | *Action* | *State* dari sistem yang mencerminkan eksekusi dari suatu aksi. |
| 5. |  | *Decision* | Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu. |
| 6. |  | *Line Connector* | Digunakan untuk menghubungkan satu simbol dengan simbol lainnya. |

### 2.4.3 *Sequence Diagram*

Menurut (Prasetya, 2022) *Sequence diagram* atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu *sequence diagaram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek- objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan.

**Tabel 2.5 Simbol – Simbol *Sequence Diagram***

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1. |  | *Actor* | Menggambarkan orang yang sedang berinteraksi dengan sistem. |
| 2. |  | *Entity Class* | Gambaran sistem sebagai landasan dalam menyusun basis data. |
| 3. |  | *Boundary Class* | Menangani komunikasi antar lingkungan sistem. |
| 4. |  | *Control Class* | Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika. |
| 5. |  | *Activation* | Mewakili proses durasi aktivasi sebuah operasi. |
| 6. |  | *Life Line* | Komponen yang digambarkan garis putus-putus terhubung dengan objek. |
| 7. |  | *A Message* | Menggambarkan pengiriman pesan |

### 2.5 Pengujian Sistem

### 2.5.1 Tujuan Pengujian

Menurut Rosalina (2020) dalam Sulistyanto & SN (2014), tujuan pengujian sistem perangkat lunak digunakan untuk mendapatkan informasi tentang kualitas perangkat lunak yang sedang diuji. Pengujian perangkat lunak memiliki signifikansi yang sangat penting karena bertujuan untuk mengidentifikasi kesalahan yang mungkin terjadi selama proses pembuatan perangkat lunak.

### 2.5.2 Pengujian *Black Box Testing*

Menurut (Ningrum, 2019) Metode *Blackbox* *Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang output pakai.

### 2.5.3 Pengujian *White Box Testing*

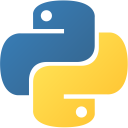
Menurut (Londjo, 2021) Pengujian *White Box*, adalah suatu metode pengujian aplikasi yang menggunakan penjelasan struktur kontrol sebagai bagian dari *component-level design* untuk membuat *test cases*.

Dalam Pengujian *White Box* para penguji perlu mengetahui secara dalam *source code* yang akan diuji. Pengujian *White Box* dapat mengungkap kesalahan implementasi dari sebuah aplikasi. Pengujian ini dapat diterapkan pada tingkatan integrasi, unit dan sistem

### 2.6 Perangkat Lunak Pendukung

Perangkat lunak yang mendukung pembuatan *chatbot* melibatkan elemen-elemen kunci atau komponen penting yang diperlukan. Beberapa di antaranya meliputi:

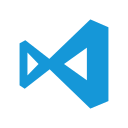
### 2.6.1 Bahasa Pemrograman *Python*



**Gambar 2.9 *Logo Python***

Menurut (Clinton, 2019), Bahasa Pemrograman *Python* adalah jenis bahasa pemrograman yang bersifat *freeware* atau bebas dalam arti sebenarnya, tanpa pembatasan dalam hal penyalinan atau distribusi. *Python* dilengkapi dengan kode sumber, alat pemecah masalah *(debugger),* dan alat pemrofilan *(profiler).* Selain itu, bahasa pemrograman ini juga menyediakan antarmuka untuk layanan sistem, fungsi *GUI* (antarmuka pengguna grafis), dan pengelolaan basis data.

### 2.6.2 *Visual Studio Code (VS Code)*



**Gambar 2.10 *Logo Visual Studio Code (VS Code)***

Menurut (A. Y. Permana, 2019), *Visual Studio Code (VS Code)* ini adalah sebuah teks *editor* ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux, Mac,* dan *Windows*. *Teks* *editor* ini secara langsung mendukung bahasa pemrograman *JavaScript, Typescript,* dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang *via marketplace Visual Studio Code* (seperti C++, C#*, Python, Go, Java,* dst).

### 2.6.3 *Uptime Robot*

Menurut (Gutama, 2021), *Uptime Robot* merupakan sebuah aplikasi *web* layanan monitoring yang berfungsi untuk memantau *server* atau *website* apakah dapat diakses dengan baik atau tidak. Tidak bisa teraksesnya *website* biasanya disebabkan oleh *hacker*, *CPanel stuck*, gangguan *hardware*, *maintenance* rutin, *over CPU*, dan sebagainya. Dengan menggunakan aplikasi *Uptime Robot* kita dapat mengetahui penyebab dari *website* tersebut sehingga tidak bisa diakses. Setiap kali aplikasi mengalami *downtime*, *Uptime Robot* akan mengirimkan pemberitahuan melalui *email* yang terdaftar sehingga kita dapat mengetahuinya sedini mungkin.

### 2.6.4 *Draw.io*



**Gambar 2.11 *Draw.io***

Menurut Purnomo (2022), *Draw.io* adalah sebuah situs *web* daring yang digunakan untuk membuat berbagai jenis *diagram*. Fitur-fitur yang tersedia di situs ini dapat diakses melalui peramban *web* yang mendukung *HTML 5*. *Draw.io* memiliki antarmuka yang menarik dan responsif, memudahkan akses baik melalui telepon pintar maupun komputer pribadi.

### 2.6.5 *Figma*



**Gambar 2.12 *Figma***

Menurut(Muhyidin, 2020), *Figma* adalah salah satu *design tool* yang biasanya digunakan untuk membuat tampilan aplikasi *mobile, desktop, website* dan lain-lain. *Figma* bisa digunakan di sistem operasi *windows, linux* ataupun *mac* dengan terhubung ke *internet*. Umumnya *Figma* banyak digunakan oleh seseorang yang bekerja dibidang *UI/UX, web design* dan bidang lainnya yang sejenis. Selain mempunyai kelengkapan fitur layaknya *Adobe XD*, *Figma* memiliki keunggulan yaitu untuk pekerjaan yang sama dapat dikerjakan oleh lebih dari satu orang secara bersama-sama walaupun ditempat yang berbeda. Hal tersebut bisa dikatakan kerja kelompok dan karena kemampuan aplikasi *figma* tersebut lah yang membuat aplikasi ini menjadi pilihan banyak *UI/UX designer* untuk membuat *prototype website* atau aplikasi dengan waktu yang cepat dan efektif.

### 2.6.6 PhpMyAdmin



**Gambar 2.13 *PhpMyAdmin***

Menurut (Firmansyah, 2021), *PhpMyAdmin* merupakan sebuah aplikasi *web* yang digunakan untuk administrasi dan manajemen basis data. Melalui *PhpMyAdmin*, pengguna dapat mengelola *database*, memberikan akses, serta melakukan berbagai tindakan administratif terhadap *database* tersebut. Proses pembuatan tabel dapat dilakukan dengan menggunakan antarmuka pengisian tabel yang disediakan oleh *PhpMyAdmin* atau alternatifnya, pengguna dapat langsung menulis *script SQL*.

### 2.6.7 *XAMPP*



**Gambar 2.14 *XAMPP***

Menurut (Hartiwati, 2022), *XAMPP* adalah sebuah perangkat lunak sumber terbuka yang mendukung berbagai sistem operasi. *XAMPP* terdiri dari kompilasi beberapa program dan berfungsi sebagai *server* mandiri *(localhost)*. Program-program utama yang termasuk dalam *XAMPP* meliputi *Apache HTTP Server*, *MySQL* *database*, serta penerjemah bahasa yang ditulis menggunakan bahasa pemrograman *PHP* dan *Perl*.

## BAB III ANALISA DAN PERANCANGAN

### 3.1 Analisa Kebutuhan

Analisa Kebutuhan pengguna dan pemangku kepentingan dalam implementasi *Chatbot* dengan memanfaatkan API *ChatGPT* melalui *Gradio App* menggunakan Metode *Natural Language Processing (NLP)*. Analisa kebutuhan ini menjadi landasan untuk merancang *Chatbot* yang responsif dan sesuai dengan harapan pengguna.

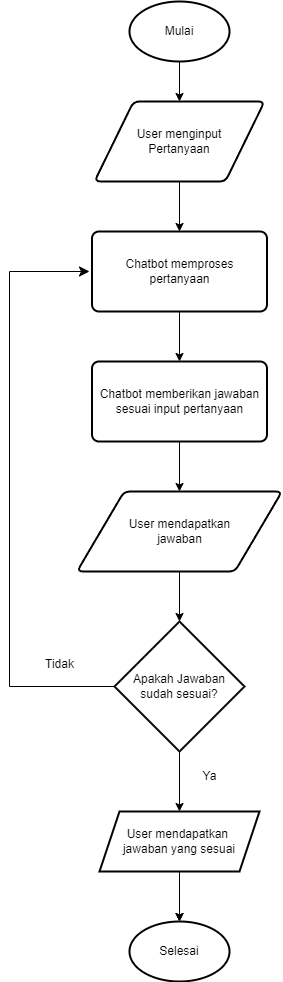
Kebutuhan yang dimaksud adalah memahami jenis pertanyaan atau permintaan yang paling sering diajukan oleh pengguna kepada *Chatbot*. penulis perlu memahami preferensi pengguna terkait dengan bahasa dan gaya komunikasi yang pengguna sukai. Selain itu, penulis harus mengetahui sejauh mana pengguna mengharapkan *Chatbot* dapat memberikan solusi atau informasi yang relevan.

Selain itu, kebutuhan kemudahan penggunaan menjadi fokus utama. penulis harus memahami tingkat pengetahuan dan keterampilan pengguna dalam menggunakan teknologi dan *Chatbot*. Identifikasi hambatan atau kendala yang mungkin dihadapi oleh pengguna dalam berinteraksi dengan *Chatbot* juga menjadi bagian penting dalam analisa ini. Penulis juga harus memahami preferensi pengguna terkait dengan tampilan dan navigasi antarmuka pengguna.

### 3.2 Analisa Sistem

### 3.2.1 Analisa Sistem Berjalan

Analisisa sistem berjalan dalam implementasi *Chatbot* dengan memanfaatkan *API ChatGPT* melalui *Gradio App* menggunakan Metode *Natural Language Processing (NLP)* melibatkan evaluasi menyeluruh terhadap kinerja dan fungsionalitas sistem yang tengah aktif. Analisa sistem berjalan ini memungkinkan pengguna untuk mengajukan pertanyaan dalam bentuk teks dan menerima respons dari *Chatbot* berdasarkan kumpulan aturan yang telah ditentukan sebelumnya.



**Gambar 3.1 Analisa Sistem Berjalan *Chatbot (ALICE)***

Berikut ini adalah penjelasan pada Gambar 3.1 Analisa Sistem Berjalan di atas:

1. *User* masuk ke halaman *Chatbot (ALICE)* melalui *link* yang tersedia.

2. *User* memasukkan pertanyaan yang ingin diajukan.

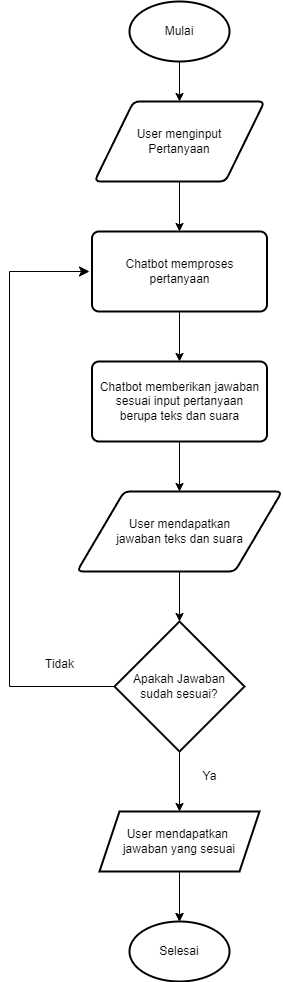
3. *Chatbot (ALICE)* mengolah *input* pertanyaan dari *user*.

4. *Chatbot (ALICE)* memberikan respons yang sesuai dengan pertanyaan yang diajukan.

5. *User* menerima respons dan memastikan respons tersebut sesuai dengan pertanyaan yang dimasukkan.

### 3.2.2 Analisa Sistem Usulan

Analisa sistem yang diajukan dalam penerapan *Chatbot* dengan memanfaatkan *API ChatGPT* melalui *Gradio App* menggunakan Metode Pemrosesan Bahasa Alam atau *Natural Language Processing* *(NLP)* yang memungkin pengguna untuk mengajukan pertanyaan dalam format teks dan menerima respons dari *Chatbot*. Respons ini dapat berupa teks dan suara, yang dihasilkan berdasarkan kumpulan aturan yang telah ditentukan sebelumnya. Tujuan analisisa ini adalah untuk memastikan bahwa sistem yang diusulkan mampu menggabungkan fungsionalitas teks dan suara, sehingga meningkatkan pengalaman interaksi pengguna dengan *Chatbot*.



**Gambar 3.2** **Analisa Sistem Usulan *Chatbot (ALICE)***

Dibawah ini merupakan penjelasan dari Analisa Sistem Usulan pada Gambar 3.2:

1. *User* memasuki halaman *Chatbot (ALICE)* melalui tautan yang telah disediakan.
2. *User* memasukkan pertanyaan yang ingin ditanyakan.
3. *Chatbot (ALICE)* memproses input pertanyaan dari *user*.
4. *Chatbot (ALICE)* memberikan jawaban yang sesuai dengan pertanyaan yang telah diinputkan, baik dalam bentuk teks maupun suara.
5. *User* menerima jawaban dalam bentuk teks dan suara, kemudian memverifikasi apakah jawaban tersebut sesuai dengan pertanyaan yang diajukan.

### 3.3 Perancangan *Database*

### 3.3.1 Spesifikasi *Database*

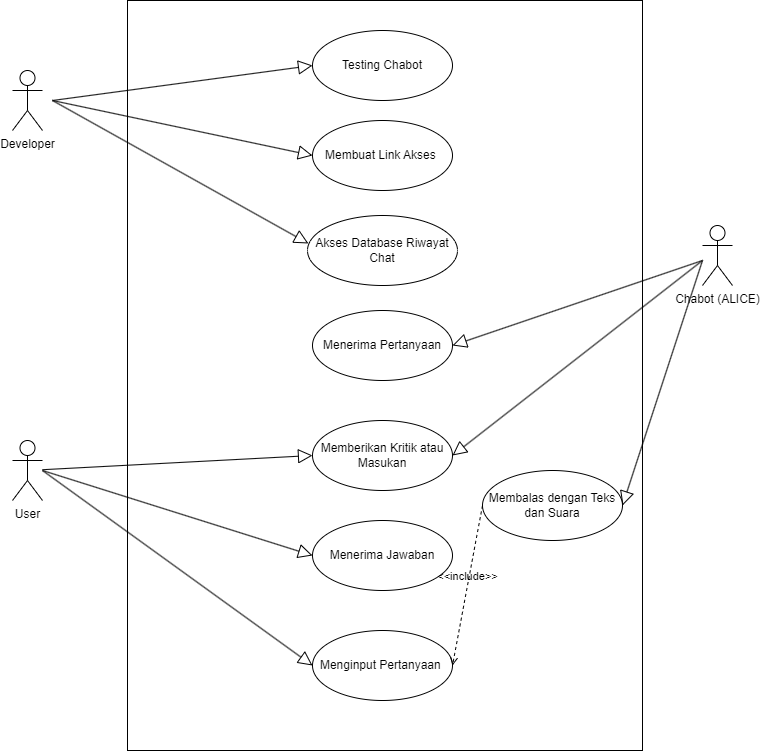
**Tabel 3.1 Rincian Spesifikasi Database Riwayat Chat**

|  |  |
| --- | --- |
| **Nama** | Riwayat Chat |
| **Alias** | *chat\_history* |
| ***Primary Key*** | Id |
| ***Foreign Key*** | - |
| **Aliran Proses** | *Developer* *view* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Tipe Data** | **Panjang Record** | **Keterangan** |
| 1. | id | int | 11 | Id histori *chat* |
| 2. | role | varchar | 225 | Pertanyaan dari *user* |
| 3. | content | text | - | Jawaban dari *chatbot (ALICE)* |

### 3.4 Perancangan *Unified Modelling Language (UML)*

### 3.4.1 *Use Case* Diagram *Chatbot (ALICE)*

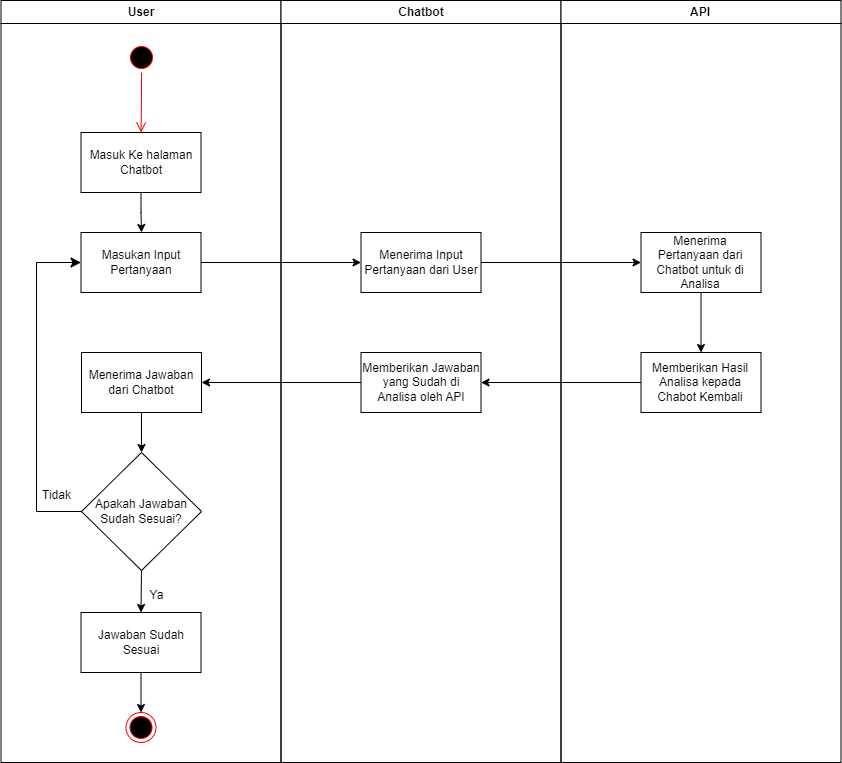


**Gambar 3.3 *Use Case* *Diagram* *Chatbot (ALICE)***

Dibawah ini merupakan penjelasan dari *Diagram Use Case* pada Gambar 3.3:

1. *Developer* adalah aktor utama yang bertanggung jawab untuk membuat *Chatbot (ALICE)*, menciptakan *link* akses *Gradio App* untuk *user* agar dapat mengakses halaman *Chatbot (ALICE)*, mengakses *database*, dan melakukan pengujian *Chatbot (ALICE)* sebelum digunakan oleh *user*.
2. *User* adalah aktor yang dapat memasukkan pertanyaan, menerima jawaban, dan memberikan kritik atau masukan kepada *Chatbot (ALICE)*.
3. *Chatbot (ALICE)* adalah aktor yang menerima *input* pertanyaan dari *user*, memberikan balasan berupa teks dan suara yang sesuai dengan *input* pertanyaan dari *user*, dan memberikan kritik atau masukan kepada *user* berdasarkan pertanyaan yang diajukan.

### 3.4.2 *Activity Diagram* Berjalan *Chatbot (ALICE)*

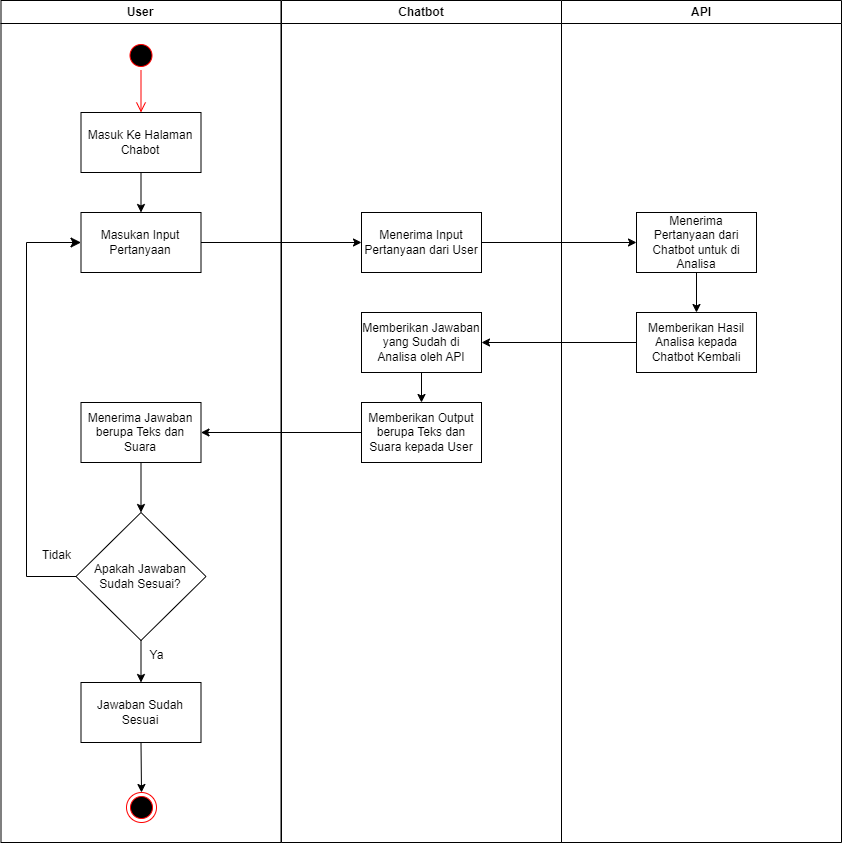


**Gambar 3.4** ***Activity Diagram* Berjalan *Chatbot (ALICE)***

Dibawah ini merupakan penjelasan dari *Activity Diagram* Berjalan pada Gambar 3.4:

1. Langkah pertama adalah *user* masuk ke halaman *Chatbot (ALICE)*.
2. *User* memasukan pertanyaan yang ingin ditanyakan.
3. *Chatbot (ALICE)* menerima *input* pertanyaan dari *User*.
4. Setelah *Chatbot (ALICE)* mendapatkan pertanyaan, *Chatbot (ALICE)* memberikan petanyaan kepada *API ChatGPT* untuk dianalisa.
5. *API ChatGPT* akan menganalisa pertanyaan yang telah diberikan oleh *Chatbot (ALICE)*, kemudian hasil analisa akan diberikan kembali ke *Chatbot (ALICE)* untuk memberikan jawaban yang telah dianalisa kepada *User*.
6. *Chatbot (ALICE)* menerima hasil pertanyaan yang telah dianalisa oleh *API ChatGPT*.
7. *User* mendapatkan jawaban dari *Chatbot (ALICE)*.
8. Terakhir, Jika jawaban telah sesuai dengan pertanyaan, *User* dapat memverifikasi kesesuaian tersebut. Jika ya, pengguna akan menerima jawaban yang sesuai dengan pertanyaan. Namun, jika belum sesuai, *User* dapat memberikan input pertanyaan yang lebih tepat kepada *Chatbot (ALICE)* untuk mendapatkan jawaban yang sesuai.

### 3.4.3 *Activity Diagram* Usulan *Chatbot (ALICE)*



**Gambar 3.5 *Activity Diagram* Usulan *Chatbot (ALICE)***

Dibawah ini merupakan penjelasan dari *Activity Diagram* Usulan pada Gambar 3.5:

1. Langkah awal adalah pengguna masuk ke halaman *Chatbot (ALICE)*.

2. *User* memasukkan pertanyaan yang ingin diajukan.

3. *Chatbot (ALICE)* menerima *input* pertanyaan dari *user*.

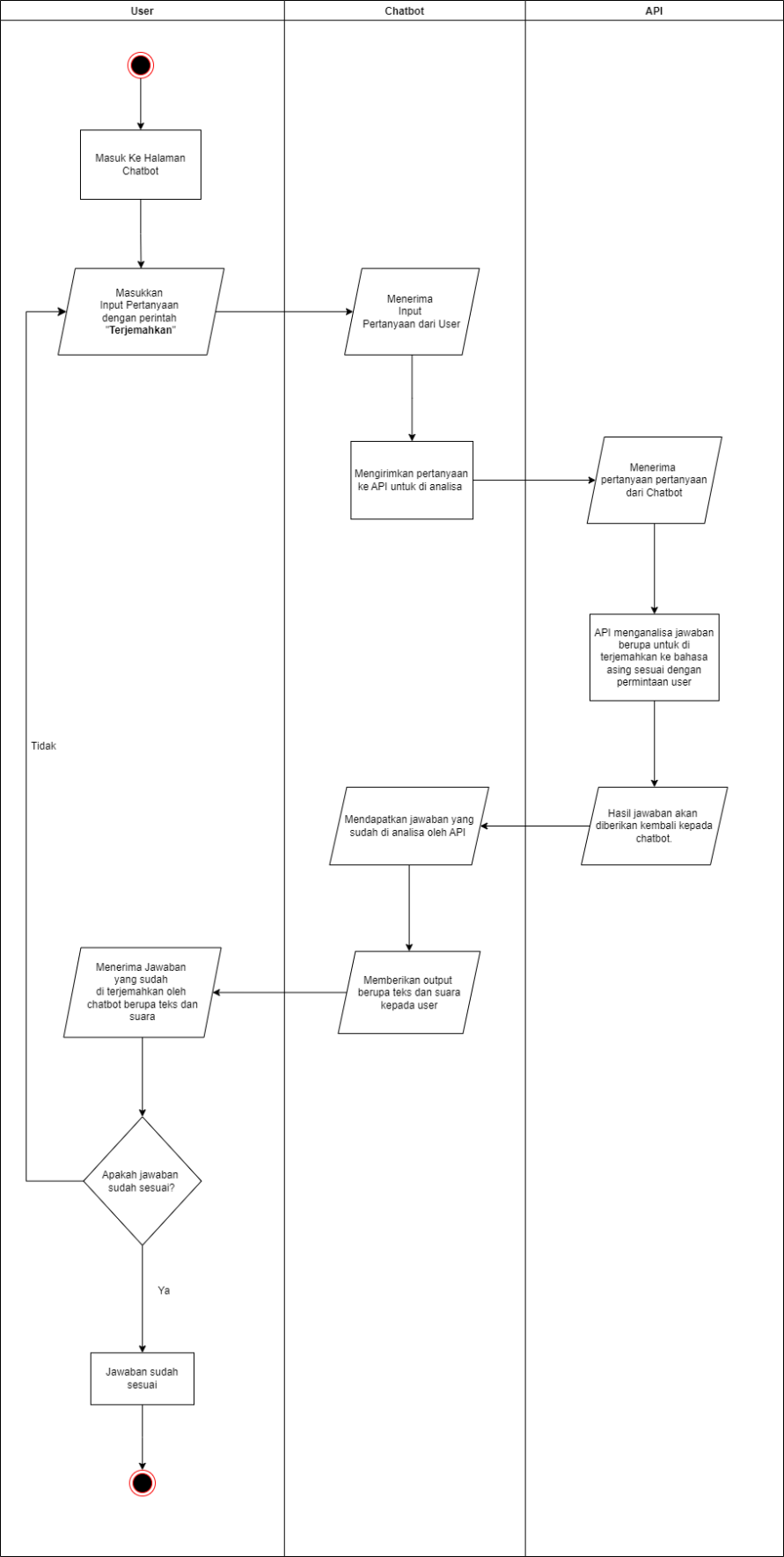
4. Setelah menerima pertanyaan, *Chatbot (ALICE)* mengirimkan pertanyaan ke *API* *ChatGPT* untuk dianalisis.

5. *API ChatGPT* menganalisis pertanyaan yang diberikan oleh *Chatbot (ALICE)*, kemudian hasil analisis dikirimkan kembali ke *Chatbot (ALICE)* untuk memberikan jawaban yang telah dianalisis kepada *user*.

6. *Chatbot (ALICE)* menerima hasil analisis pertanyaan dari *API ChatGPT* dan mengirimkan jawaban kepada *user* dalam bentuk teks dan suara.

7. Terakhir, jika jawaban sesuai dengan pertanyaan, *user* dapat memverifikasi kesesuaian tersebut. Jika ya, pengguna akan menerima jawaban yang sesuai. Namun, jika tidak sesuai, pengguna dapat memberikan *input* pertanyaan yang lebih spesifik kepada *Chatbot (ALICE)* untuk mendapatkan jawaban yang sesuai.

### 3.4.4 *Activity Diagram* Penerjemah Bahasa Asing *(Translate)*

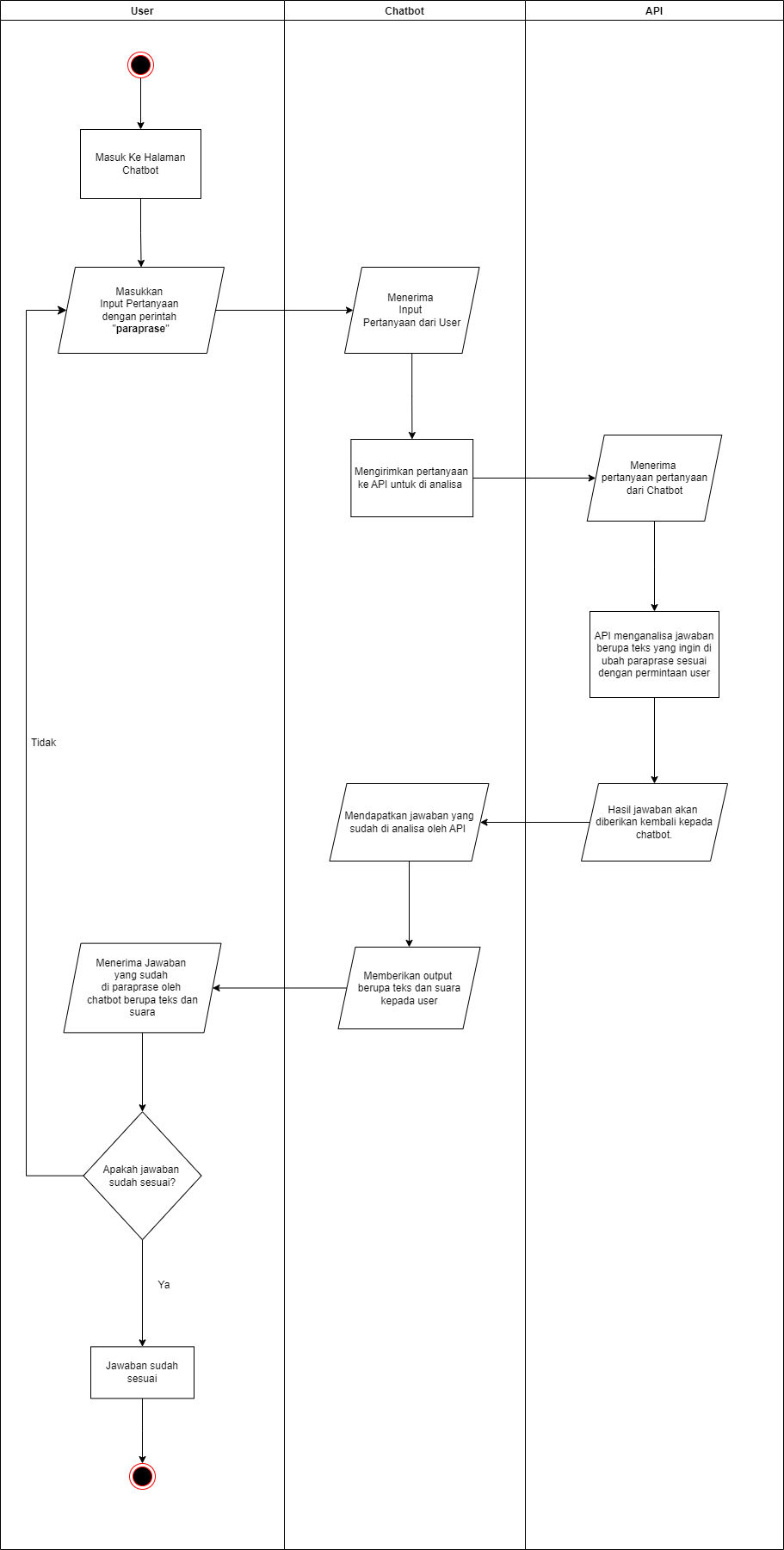


**Gambar 3.6 *Activity Diagram* Penerjemah Bahasa *(Translate) Chatbot (ALICE)***

Dibawah ini merupakan penjelasan dari *Activity Diagram* Penerjemah Bahasa *(Transalate)* pada Gambar 3.6:

1. Langkah awal adalah pengguna masuk ke halaman *Chatbot (ALICE)*.
2. *User* memasukkan pertanyaan yang ingin diajukan dengan menyertakan perintah “*Translate* atau Terjemahkan”.
3. *Chatbot (ALICE)* menerima *input* pertanyaan dari *user*.
4. Setelah menerima pertanyaan, *Chatbot (ALICE)* mengirimkan pertanyaan ke *API* *ChatGPT* untuk dianalisis.
5. *API ChatGPT* menganalisis pertanyaan yang diberikan oleh *Chatbot (ALICE)* sesuai perintah yang diajukan oleh user, kemudian hasil analisis dikirimkan kembali ke *Chatbot (ALICE)* untuk memberikan jawaban yang telah dianalisis kepada *user*.
6. *Chatbot (ALICE)* menerima hasil analisis pertanyaan dari *API ChatGPT* dan mengirimkan jawaban kepada *user* dalam bentuk teks dan suara.
7. Terakhir, jika jawaban sesuai dengan pertanyaan, *user* dapat memverifikasi kesesuaian tersebut. Jika ya, pengguna akan menerima jawaban yang sesuai. Namun, jika tidak sesuai, pengguna dapat memberikan *input* pertanyaan yang lebih spesifik kepada *Chatbot (ALICE)* untuk mendapatkan jawaban yang sesuai.

### 3.4.5 *Activity Diagram* Paraprase

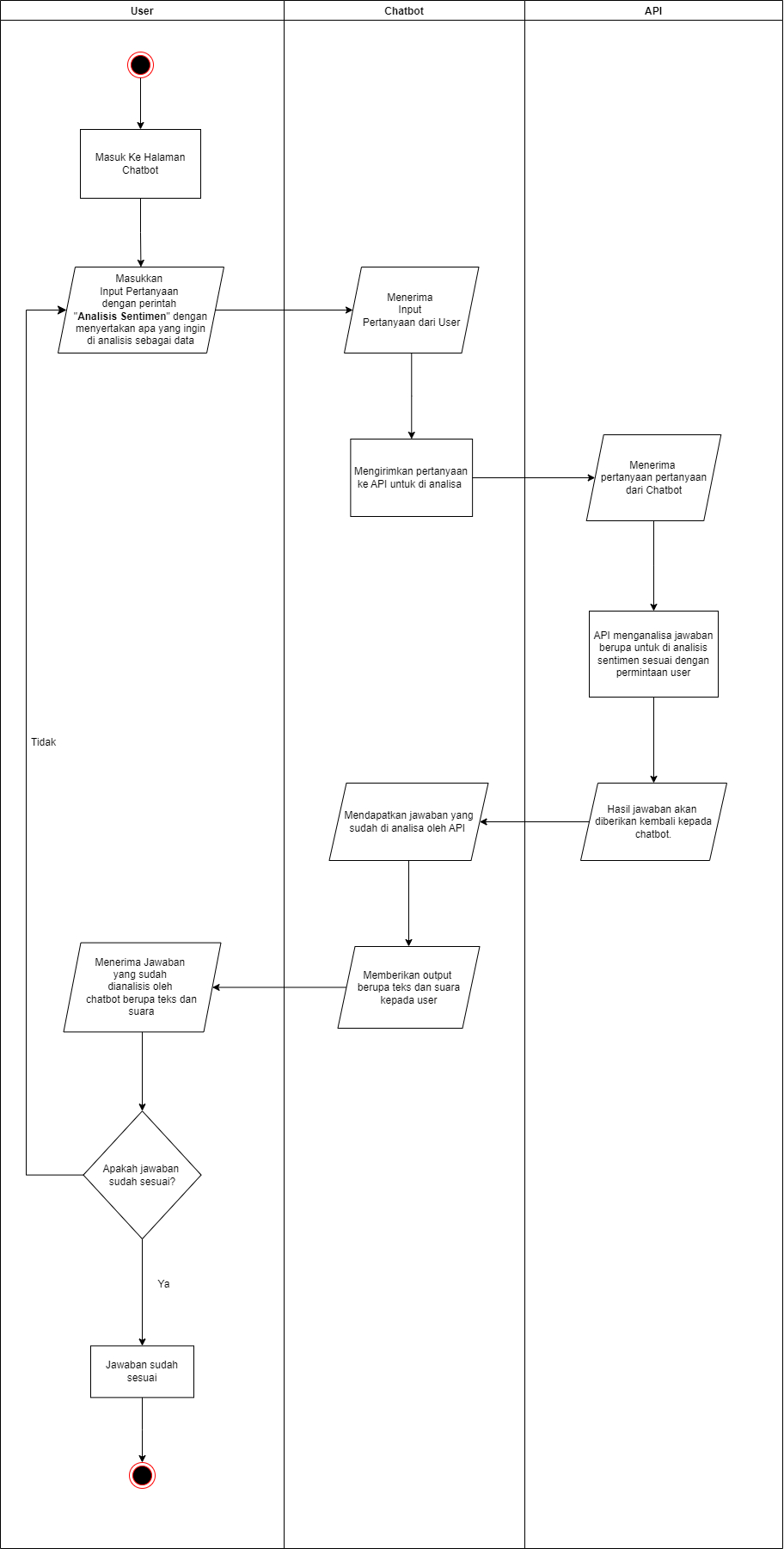


**Gambar 3.7 *Activity Diagram* Paraprase *Chatbot (ALICE)***

Dibawah ini merupakan penjelasan dari *Activity Diagram* Paraprase pada Gambar 3.7

1. Langkah awal adalah pengguna masuk ke halaman *Chatbot (ALICE)*.
2. *User* memasukkan pertanyaan yang ingin diajukan dengan menyertakan perintah “Paraprase”.
3. *Chatbot (ALICE)* menerima *input* pertanyaan dari *user*.
4. Setelah menerima pertanyaan, *Chatbot (ALICE)* mengirimkan pertanyaan ke *API* *ChatGPT* untuk dianalisis.
5. *API ChatGPT* menganalisis pertanyaan yang diberikan oleh *Chatbot (ALICE)* sesuai perintah yang diajukan oleh user, kemudian hasil analisis dikirimkan kembali ke *Chatbot (ALICE)* untuk memberikan jawaban yang telah dianalisis kepada *user*.
6. *Chatbot (ALICE)* menerima hasil analisis pertanyaan dari *API ChatGPT* dan mengirimkan jawaban yaitu teks yang sudah di ubah paraprase kepada *user* dalam bentuk teks dan suara.
7. Terakhir, jika jawaban sesuai dengan pertanyaan, *user* dapat memverifikasi kesesuaian tersebut. Jika ya, pengguna akan menerima jawaban yang sesuai. Namun, jika tidak sesuai, pengguna dapat memberikan *input* pertanyaan yang lebih spesifik kepada *Chatbot (ALICE)* untuk mendapatkan jawaban yang sesuai.

### 3.4.6 *Activity Diagram* Analisis Sentimen

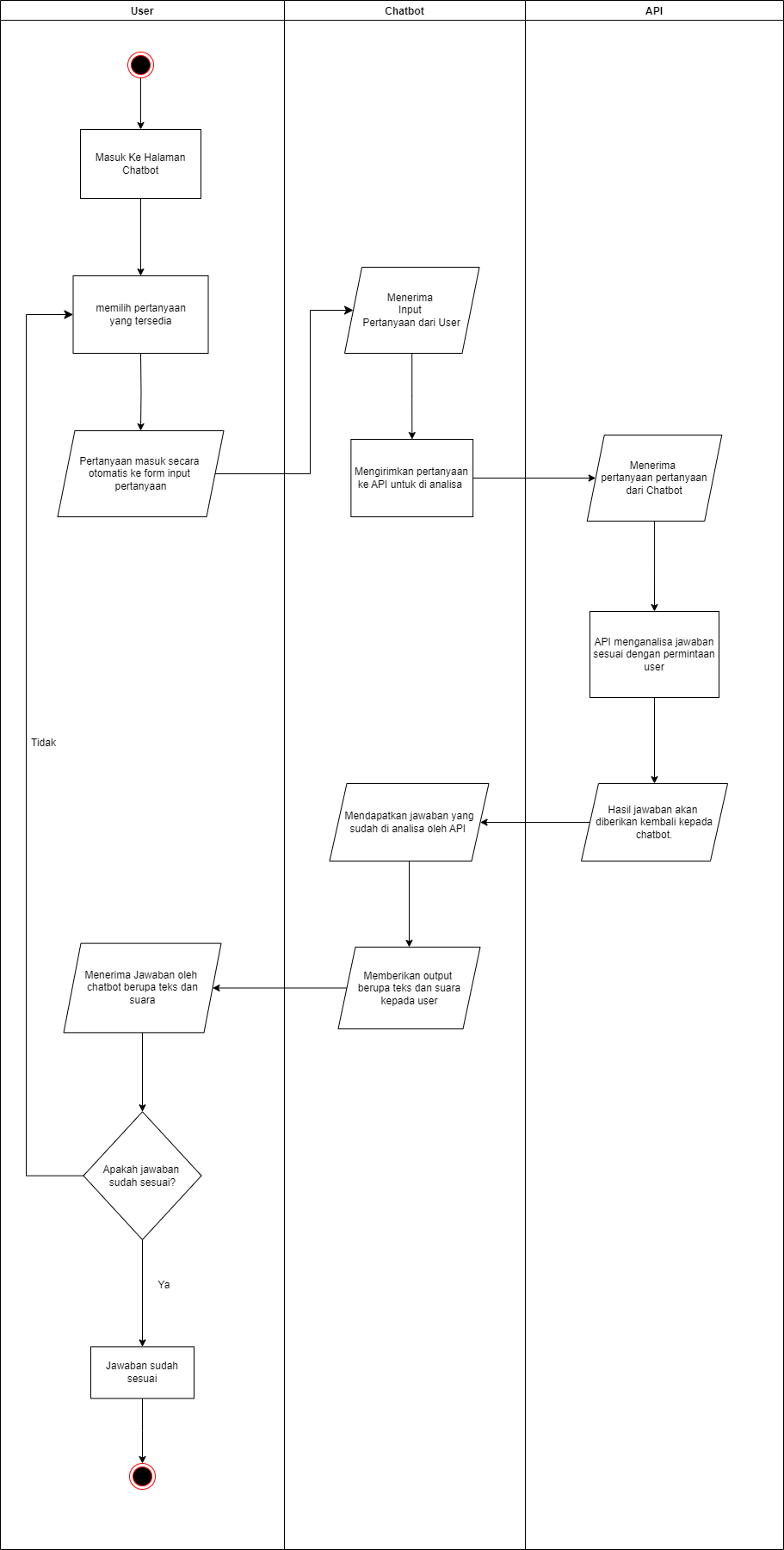


**Gambar 3.8 *Activity Diagram* Analisis Sentimen *Chatbot (ALICE)***

Dibawah ini merupakan penjelasan dari *Activity Diagram* Analisis Sentimen pada Gambar 3.8:

1. Langkah awal adalah pengguna masuk ke halaman *Chatbot (ALICE)*.
2. *User* memasukkan pertanyaan yang ingin diajukan dengan menyertakan perintah “Analisi Sentimen” dengan menyertakan apa yang ingin dianalisis sebagai data tambahan agar *Chatbot (ALICE)* bisa memberikan jawaban yang sesuai.
3. *Chatbot (ALICE)* menerima *input* pertanyaan dari *user*.
4. Setelah menerima pertanyaan, *Chatbot (ALICE)* mengirimkan pertanyaan ke *API* *ChatGPT* untuk dianalisis.
5. *API ChatGPT* menganalisis pertanyaan yang diberikan oleh *Chatbot (ALICE)* sesuai perintah yang diajukan oleh user, kemudian hasil analisis dikirimkan kembali ke *Chatbot (ALICE)* untuk memberikan jawaban yang telah dianalisis kepada *user*.
6. *Chatbot (ALICE)* menerima hasil analisis pertanyaan dari *API ChatGPT* dan mengirimkan jawaban kepada *user* dalam bentuk teks dan suara.
7. Terakhir, jika jawaban sesuai dengan pertanyaan, *user* dapat memverifikasi kesesuaian tersebut. Jika ya, pengguna akan menerima jawaban yang sesuai. Namun, jika tidak sesuai, pengguna dapat memberikan *input* pertanyaan yang lebih spesifik kepada *Chatbot (ALICE)* untuk mendapatkan jawaban yang sesuai.

### 3.4.7 *Activity Diagram* Pemilihan *Dataset*

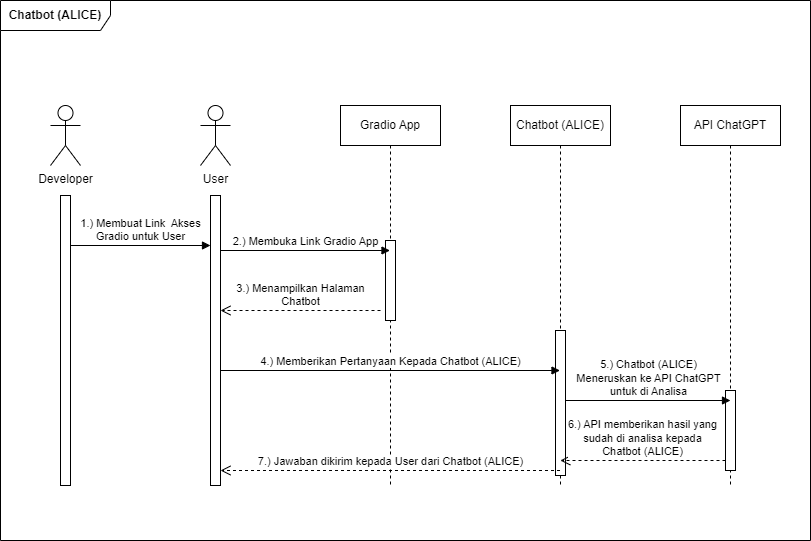


**Gambar 3.9 *Activity Diagram Dataset Chatbot (ALICE)***

Dibawah ini merupakan penjelasan dari *Activity Diagram* Dataset pada Gambar 3.9:

1. Langkah awal adalah pengguna masuk ke halaman *Chatbot (ALICE)*.
2. *User* memilih pertanyaan dari *dataset* yang sudah tersedia.
3. Pertanyaan *dataset* yang sudah dipilih akan berpindah ke *form input* pertanyaan secara otomatis.
4. *Chatbot (ALICE)* menerima *input* pertanyaan dari *user*.
5. Setelah menerima pertanyaan, *Chatbot (ALICE)* mengirimkan pertanyaan ke *API* *ChatGPT* untuk dianalisis.
6. *API ChatGPT* menganalisis pertanyaan yang diberikan oleh *Chatbot (ALICE)* sesuai pertanyaan yang dipilih, kemudian hasil analisis dikirimkan kembali ke *Chatbot (ALICE)* untuk memberikan jawaban yang telah dianalisis kepada *user*.
7. *Chatbot (ALICE)* menerima hasil analisis pertanyaan dari *API ChatGPT* dan mengirimkan jawaban kepada *user* dalam bentuk teks dan suara.
8. Terakhir, jika jawaban sesuai dengan pertanyaan, *user* dapat memverifikasi kesesuaian tersebut. Jika ya, pengguna akan menerima jawaban yang sesuai. Namun, jika tidak sesuai, pengguna dapat memberikan *input* pertanyaan yang lebih spesifik kepada *Chatbot (ALICE)* untuk mendapatkan jawaban yang sesuai.

### 3.4.8 *Sequence Diagram Chatbot (ALICE)*



**Gambar 3.10 *Sequence Diagram Chatbot (ALICE)***

Dibawah ini merupakan penjelasan dari *Sequence Diagram* pada Gambar 3.6:

1. Langkah Awal adalah *Developer* membuat *link gradio app* sebagai akses untuk pengguna agar dapat mengakses halaman *Chatbot (ALICE)*.

2. Pengguna menerima link yang telah dibuat oleh *developer* untuk mengakses halaman *Chatbot (ALICE)*.

3. *Link gradio* menampilkan halaman *Chatbot (ALICE)* bagi User.

4. Pengguna memasukkan pertanyaan yang ingin diajukan kepada *Chatbot (ALICE).*

5. *Chatbot (ALICE)* menerima *input* pertanyaan dari pengguna, kemudian meneruskannya kepada *API ChatGPT* untuk dianalisis.

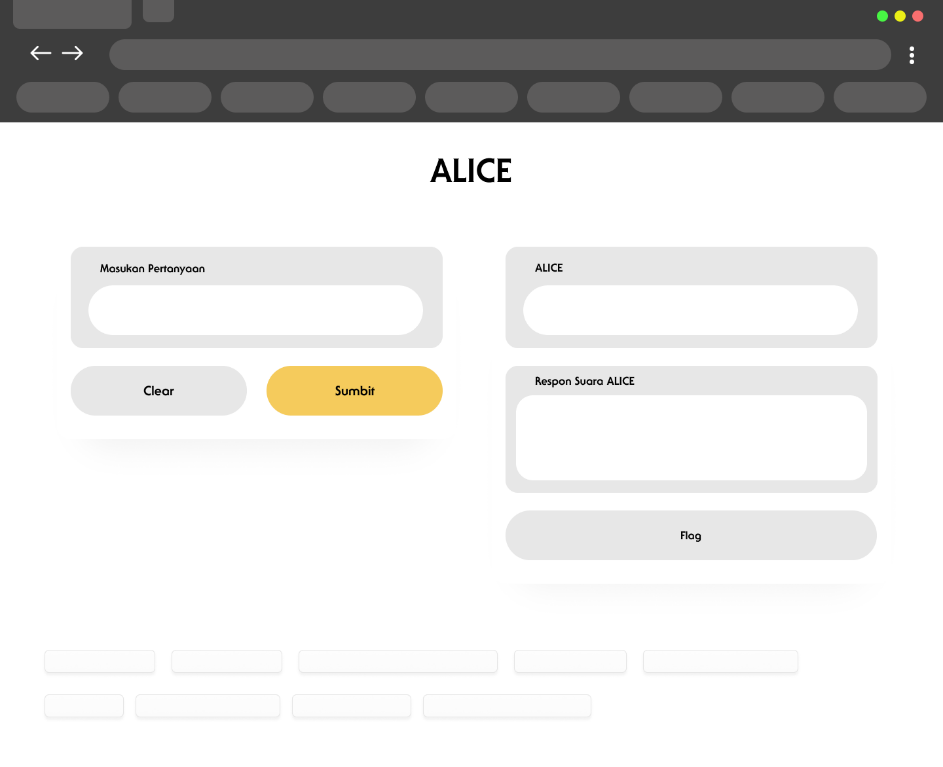
6. Setelah itu, *API ChatGPT* memberikan hasil analisis kepada *Chatbot (ALICE)* dalam bentuk jawaban atas pertanyaan tersebut.

7. Akhirnya, *Chatbot (ALICE)* menjawab pertanyaan yang telah dianalisis tersebut kepada *User*.

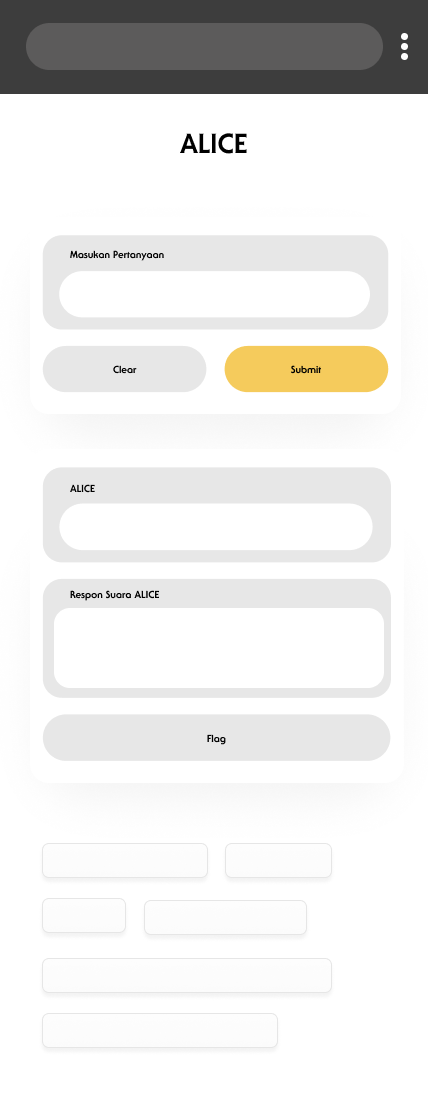
### 3.5 Perancangan Antarmuka *(User Interface)*

Menurut (Gamas, 2022) perancangan Antarmuka Pengguna *(User Interface)* adalah elemen komputer yang memfasilitasi interaksi pengguna dengan sistem.

Dibawah ini adalah perancangan Antamuka (*User Interface)* pada *Chatbot (ALICE)*:

****

**Gambar 3.11 Rancangan Tampilan Halaman *Web Browser Chatbot (ALICE)* pada Perangkat *Laptop***

****

**Gambar 3.12 Rancangan Tampilan Halaman *Web Browser Chatbot (ALICE)* pada Perangkat *Smartphone***

Gambar 3.7 dan gambar 3.8 adalah tampilan antarmuka *Gradio App* *Chatbot (ALICE)*, tampilan tersebut adalah tampilan yang di akses melalui *laptop* atau *smartphone* pengguna dan memberikan simulasi tanya jawab pada *Chatbot (ALICE).*

## BAB IV IMPLEMENTASI DAN PENGUJIAN

### 4.1 Spesifikasi

### 4.1.1 Spesifikasi Perangkat Lunak

Pengembangan perangkat lunak ini dengan menggunakan *Python, Gradio App, API ChatGPT, Visual Studio Code (VS Code)*, dan *Uptime Robot*. *Python* dipilih sebagai bahasa pemrograman utama, sementara *Gradio App* dipilih sebagai perangkat lunak untuk antarmuka dengan model *machine learning* yang dapat diakses secara luas oleh siapa pun, di mana pun. *API ChatGPT* sebagai pengembang perangkat lunak untuk berkomunikasi dengan model bahasa *GPT (Generative Pre-trained Transformer).* *Visual Studio Code (VS Code)* sebagai pengembang perangkat lunak untuk menulis, menguji, dan menjalankan kode komputer. Sementara *Uptime Robot* sebagai memantau apakah situs *web* atau *server* berfungsi dengan baik atau mengalami masalah. Adapun perangkat lunak yang digunakan sebagai berikut:

**Tabel 4.1 Perangkat Lunak**

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat Lunak** | **Keterangan** |
| 1. | Sistem Operasi | *Windows 10 64-bit* |
| 2. | Bahasa Pemprograman | *Python* |
| 3. | Kode *Editor* | *Visual Studio Code (VS Code)* |
| 4. | *Web Browser* | *Google Chrome* |
| 5. | *Server* | *Uptime Bot* |

### 4.1.2 Spesifikasi Perangkat Keras

Perangkat keras menjelaskan kebutuhan *minimum* perangkat keras yang diperlukan untuk mengimplementasikan program aplikasi yang dibuat, semakin tinggi spesifikasi komputer yang digunakan untuk menjalankan aplikasi, maka akan semakin baik. Dalam spesifikasi perangkat keras ini terdapat beberapa perangkat keras pendukung sebagai tempat untuk menerapkan *Chatbot* ini. Adapun perangkat keras yang digunakan sebagai berikut:

**Tabel 4.2 Perangkat Keras pada *Laptop***

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat Keras** | **Keterangan** |
| 1. | *Processor* | *Intel(R) Celeron(R) N4500 @ 1.10GHz 1.11 GHz* |
| 2. | *Memory RAM* | 4 *GigaByte* |
| 3. | *Memory HDD* | 500 *GigaByte* |
| 4. | *VGA* | *Intel HD Graphic* |
| 5. | *OS* | *Windows 7,8,10 & 11 Pro 32/64 bit* |

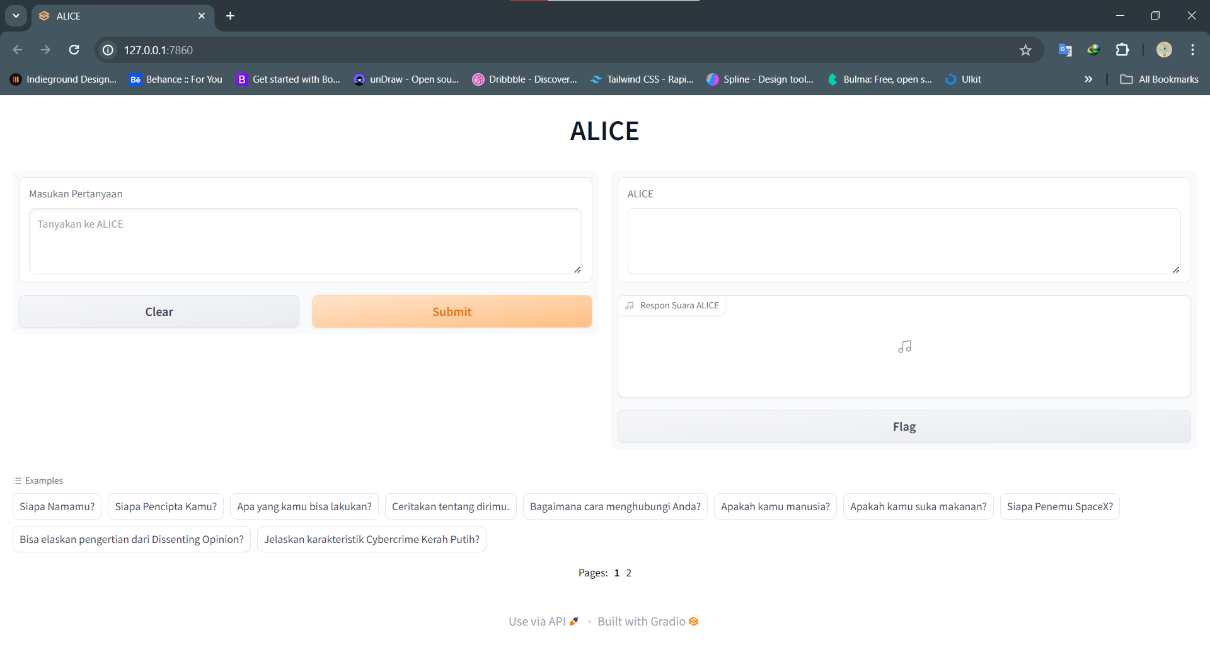
**Tabel 4.3 Perangkat Keras pada *Smartphone***

|  |  |  |
| --- | --- | --- |
| **No** | **Perangkat Keras** | **Keterangan** |
| 1. | *Processor* | *Snapdragon 439* |
| 2. | *Memory RAM* | 2 *GigaByte* |
| 3. | *Memory HDD* | 32 *GigaByte* |
| 4. | *OS* | *Funtouch OS* 9.1 |

### 4.2 Implementasi Program

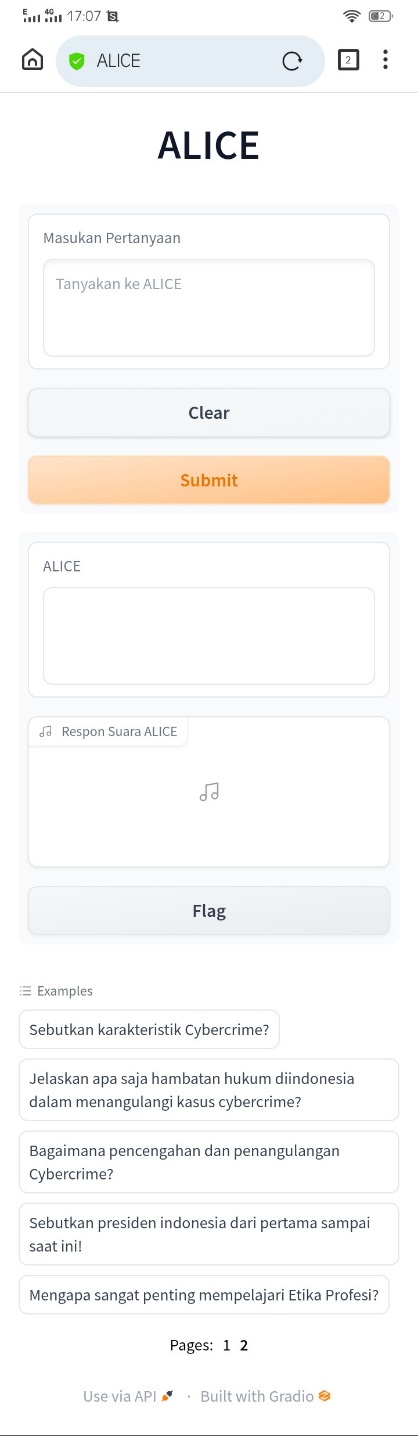
### 4.2.1 Halaman *Chatbot (ALICE)*

Dibawah merupakan gambaran dari tampilan halaman pada *Chatbot (ALICE)*:



**Gambar 4.1 Halaman *Web Browser Chatbot (ALICE)* pada Perangkat *Laptop***

Gambar 4.1, terlihat bahwa di sisi kiri terdapat *form input* yang memungkinkan *user* untuk mengajukan pertanyaan. Selain itu, terdapat tombol *"Clear"* yang digunakan untuk menghapus teks pertanyaan, dan tombol *"Submit"* untuk mengirim pertanyaan kepada *Chatbot (ALICE)*. Di sisi kanan, terdapat *form output* yang berfungsi sebagai area jawaban, yang mencakup teks dan suara. Selain itu, terdapat juga sebuah *dataset* yang memuat beberapa pertanyaan, bertujuan untuk mempermudah *user* dalam mengajukan pertanyaan.

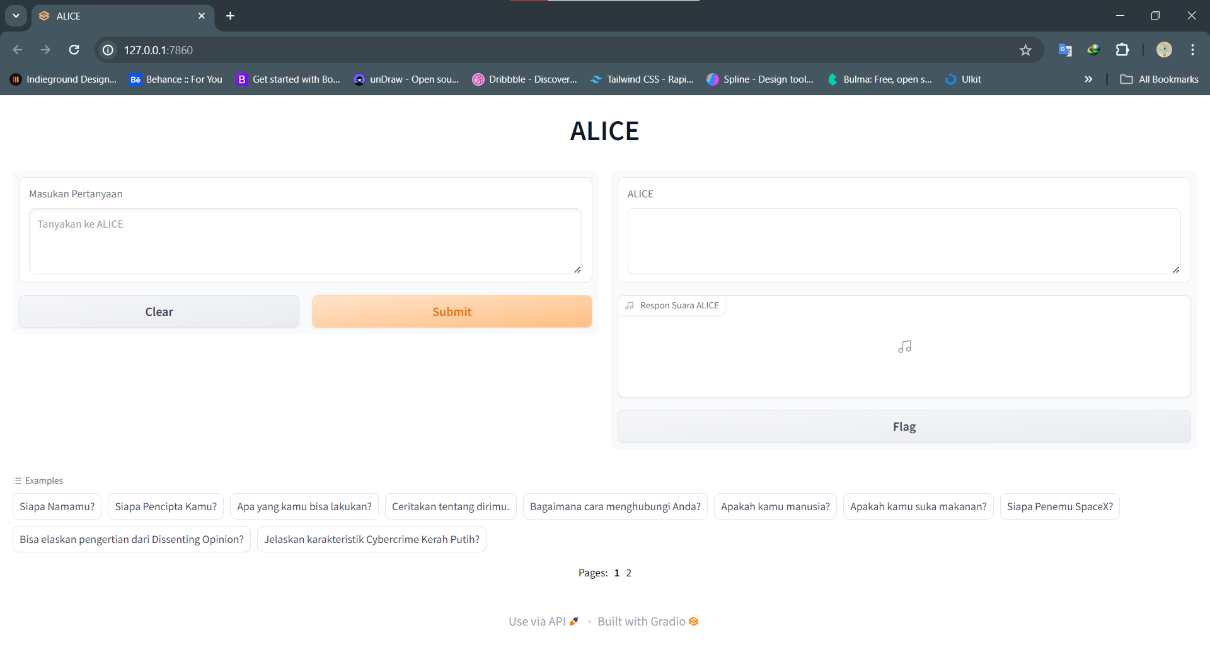


**Gambar 4.2 Halaman *Web Browser* *Chatbot (ALICE)* pada Perangkat *Smartphone***

Gambar 4.2, terlihat bahwa di bagian atas halaman terdapat sebuah *form input* yang memungkinkan *user* untuk mengajukan pertanyaan. Selain itu, terdapat tombol *"Clear"* yang digunakan untuk menghapus teks pertanyaan, dan tombol *"Submit"* yang berfungsi untuk mengirim pertanyaan kepada *Chatbot (ALICE)*. Di bagian bawah halaman, terdapat *form output* yang berfungsi sebagai area jawaban, mencakup teks dan suara. Terdapat juga sebuah *dataset* yang berisi beberapa pertanyaan, bertujuan untuk mempermudah *user* dalam mengajukan pertanyaan.

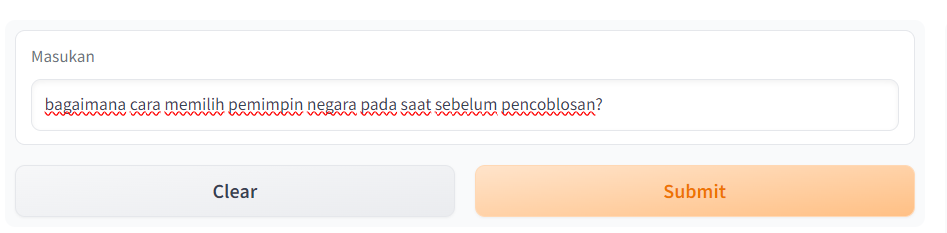
### 4.3 Implementasi Program

Analisis sistem merupakan suatu kegiatan yang melibatkan penemuan atau identifikasi masalah, evaluasi, pembuatan model, dan penyusunan spesifikasi sistem. Tujuan tersebut adalah untuk merancang sistem baru atau memperbaiki kelemahan yang ada dalam sistem yang sudah ada.



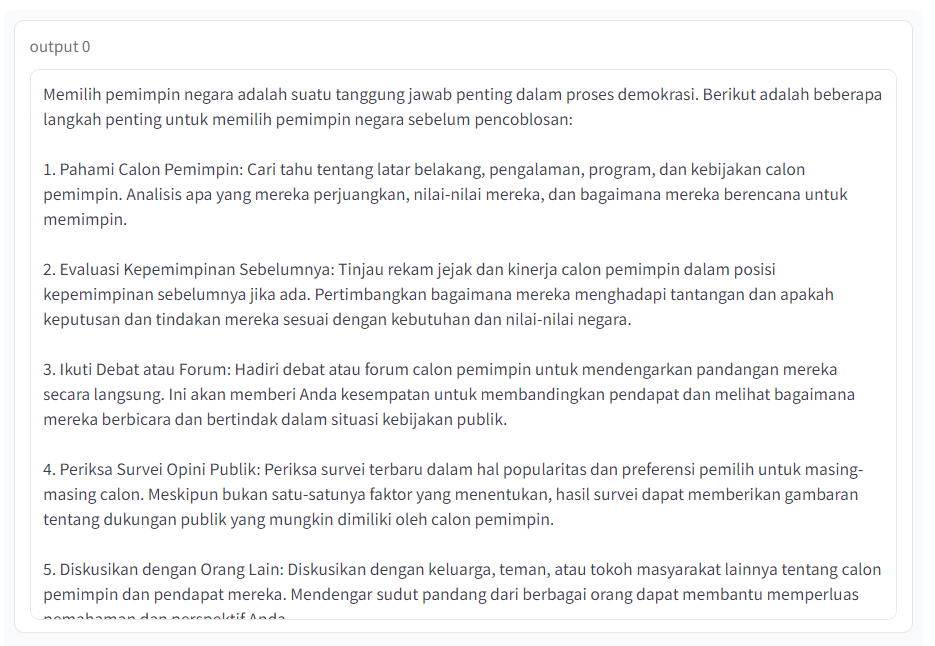
**Gambar 4.3 Halaman *Chatbot (ALICE) Demo* pada Perangkat *Laptop***

Gambar 4.3 menampilkan Tampilan Halaman pada *Chatbot (ALICE)* yang akan muncul ketika pengguna mengakses *link* yang telah diberikan oleh *Developer*.



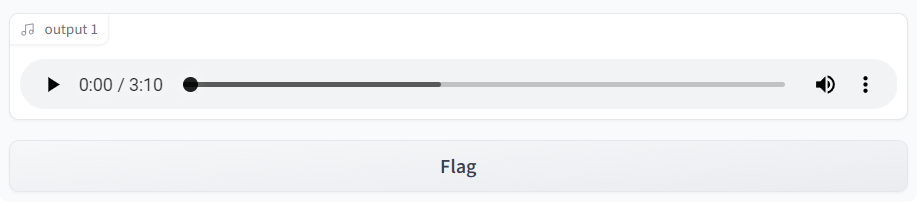
**Gambar 4.4 Tampilan *Form* Pertanyaan pada *Chatbot (ALICE) Demo* pada Perangkat *Laptop***

Gambar 4.4 menampilkan Tampilan *Form* Pertanyaan pada halaman *Chatbot (ALICE)*, di mana pengguna dapat mengajukan pertanyaan yang diinginkan kepada *Chatbot (ALICE)*.



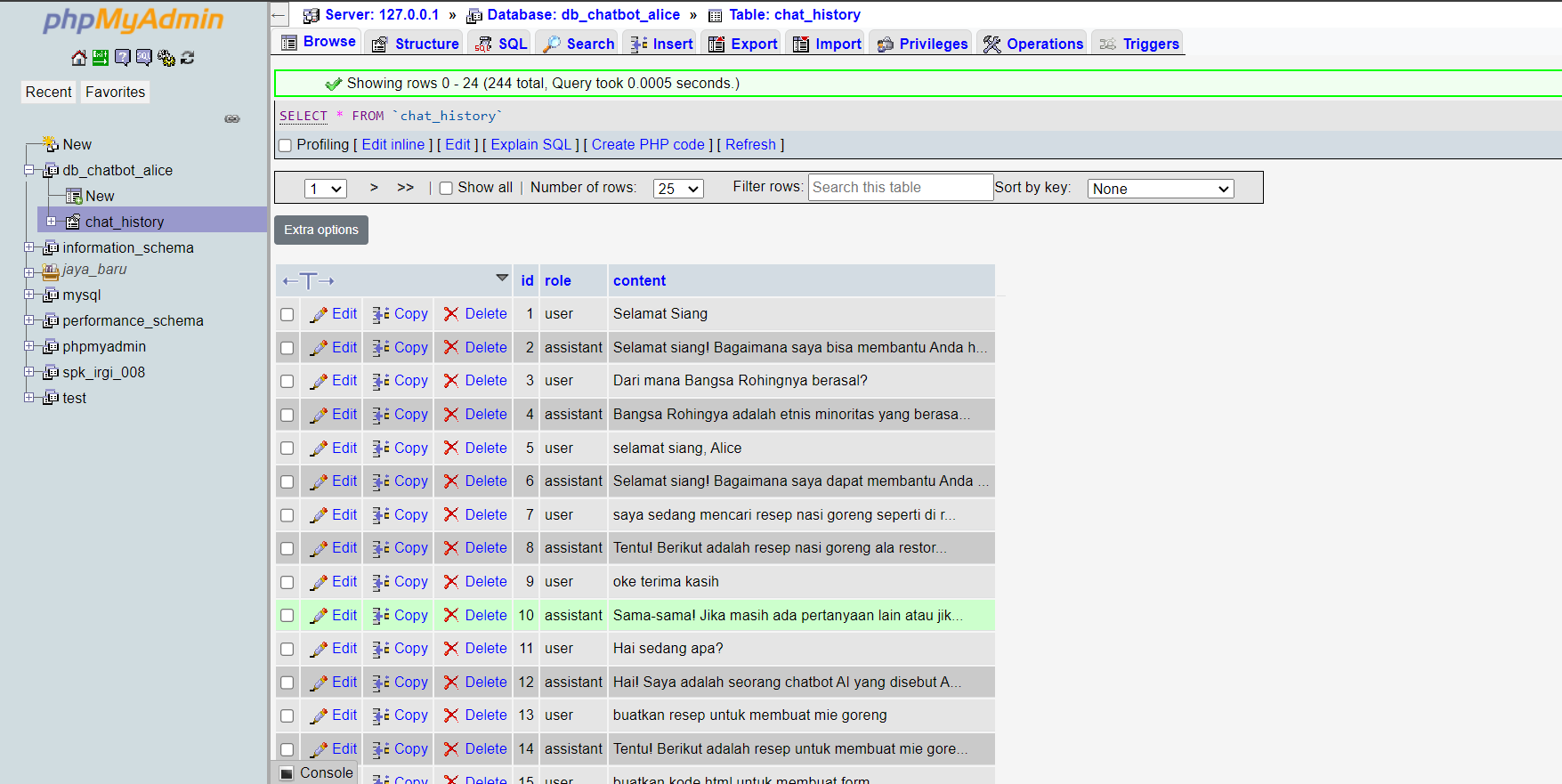
**Gambar 4.5 Tampilan *Form* Jawaban pada *Chatbot (ALICE) Demo* pada Perangkat *Laptop***

Gambar 4.5 menunjukkan Tampilan *Form* Jawaban pada halaman *Chatbot (ALICE)*, di mana pengguna akan menerima jawaban yang sesuai dengan input pertanyaan yang diajukan kepada Chatbot (ALICE).



**Gambar 4.6 Tampilan *Form Audio* jawaban pada *Chatbot (ALICE)***

Gambar 4.6 menampilkan *Form Audio* Jawaban pada halaman *Chatbot (ALICE)*. Dalam fitur ini, pengguna akan menerima jawaban dalam bentuk *audio*/suara yang dihasilkan dari teks jawaban, yang kemudian dibacakan kembali.



**Gambar 4.7 Tampilan Database Histori Chat**

Gambar 4.7 menampilkan *database* yang berisi riwayat percakapan antara pengguna dan *chatbot (ALICE)*, di mana setelah sesi tanya jawab antara pengguna dan *chatbot* selesai, sistem akan menyimpan pertanyaan yang sudah dijawab oleh *chatbot* ke dalam *database*.

4.3 Pengujian Sistem

### 4.3.1 Sistem *Black Box*

Menggunakan asumsi tidak kenal struktur internal dari program *(black box).* Berkonsentrasi untuk menemukan kondisi dimana program tidak berjalan susuai dengan spesifikasi (fungsional) menggunakan spesifikasi untuk *data test*.

Pengujian *Black Box* berfokus pada *input* dan *output* sistem tanpa mempertimbangkan bagaimana sistem menghasilkan *output* tersebut. Misalnya, jika sebuah fungsi seharusnya menghasilkan hasil penjumlahan dua angka, pengujian *Black Box* hanya akan memeriksa apakah fungsi tersebut menghasilkan hasil yang benar, tanpa memeriksa bagaimana fungsi tersebut melakukan penjumlahan.

Berikut di bawah ini adalah data pengujian *Black Box* pada *Chatbot (ALICE)* yang dikembangkan oleh peneliti:

**Tabel 4.4 Pengujian *Black Box pada Chatbot (ALICE)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Detail Pengujian** | **Output yang diharapkan** | **Output yang keluar** | **Kesimpulan** |
| 1. | Mengirim Pertanyaan | Apabila *Button Submit* pada *form input* ditekan, sistem akan mengajukan pertanyaan kepada *Chatbot (ALICE).* | Apabila *Button Submit* pada *form input* ditekan, sistem berhasil mengajukan pertanyaan kepada *Chatbot (ALICE).* | **Sesuai** |
| 2. | Mengapus semua pertanyaan dan jawaban | Apabila *Button Clear* ditekan, maka semua *input* dan *output* yang terisi akan dihapus. | Apabila *Button Clear* ditekan, maka semua *input* dan *output* yang terisi berhasil dihapus. | **Sesuai** |
| 3. | *Output* berupa jawaban | Setelah *User* mengirimkan pertanyaan dan menekan tombol "*Submit*", *Chatbot (ALICE)* akan memberikan jawaban yang sesuai dengan pertanyaan yang diajukan oleh *User*. | Berhasil, *User* mendapatkan jawaban yang sesuai dengan pertanyaan yang diajukan. | **Sesuai** |
| 4. | *Output* Jawaban *(Translate)* | Saat *User* ingin meminta teks jawaban menjadi bahasa asing. | Berhasil, *User* dapat meminta *Chatbot (ALICE)* agar jawaban menjadi bahasa, sesuai dengan *input* pertanyaan yang diminta. | **Sesuai** |
| 5. | *Output* Jawaban (Paraprase) | Saat *User* ingin menyusun ulang teks pertanyaan untuk memperbaiki gaya penulisan dengan paraprase. | Berhasil, saat *User* ingin menyusun ulang teks pertanyaan untuk memperbaiki gaya penulisan dengan paraprase. | **Sesuai** |
| 6. | *Output* Jawaban (Analisis Sentimen) | Ketika *User* ingin menentukan sentimen (*positif*, *negatif*, atau *netral*) dalam teks, hal tersebut dapat digunakan untuk menganalisis pandangan orang terhadap suatu topik, bergantung pada data yang diberikan kepada *Chatbot (ALICE)*. | Berhasil, *User* dapat menentukan sentimen (*positif*, *negatif*, atau *netral*) dalam teks, hal tersebut dapat digunakan untuk menganalisis pandangan orang terhadap suatu topik, bergantung pada data yang diberikan kepada *Chatbot (ALICE)*. | **Sesuai** |
| 7. | *Audio Output ALICE 1* | Saat *User* mengirim pertanyaan, *Chatbot (ALICE)* akan memberikan jawaban dalam format *audio* yang dapat diulang-ulang dan diunduh. | Berhasil, *User* mendapatkan jawaban dalam format *audio* yang dapat diulang-ulang dan diunduh. | **Sesuai** |
| 8. | *Audio Output ALICE 2* | *User* memberikan pertanyaan kepada sistem, *Chatbot (ALICE)* secara otomatis akan memberikan jawaban dalam bentuk *file* *audio* tanpa perlu mengunduh ulang. | Berhasil, *User* mendapatkan jawaban dalam bentuk *file* *audio* tanpa perlu mengunduh ulang. | **Sesuai** |
| 9. | Memilih Pertanyaan dari *dataset* | Saat *User* memilih pertanyaan dari dataset pada bagian bawah, maka pertanyaan yang sudah diklik akan ter-*input* secara otomatis ke *form* pertanyaan. | Berhasil, Saat *User* memilih pertanyaan dari dataset pada bagian bawah, maka pertanyaan yang sudah diklik akan ter-input secara otomatis ke *form* pertanyaan. | **Sesuai** |

### 4.3.2 Sistem *White Box*

Pengujian yang didasarkan pada detail prosedur dan alur logika kode program. Pada kegiatan *white box testing*, tester melihat *source code* program dan menemukan *bugs* dari kode program yang diuji. Intinya *white box testing* adalah pengujian yang dilakukan sampai kepada detail pengecekan kode program.

*White Box Testing* merupakan cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada dan menganalisis jika ada kesalahan. Jika ada modul yang menghasilkan output yang tidak sesuai dengan proses yang dilakukan maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki kemudian di *compile* ulang.

Pengujian *white box* dapat membantu dalam mengindetifikasi kesalahan dan cacat pada kode sumber aplikasi. Hal ini dapat membantu dalam meningkatkan kualitas kode sumber dan mengurangi jumlah kesalahan yang muncul pada aplikasi. Selain itu, pengujian *white box* dapat membantu dalam meningkatkan efisiensi pengujian karena dapat mengurangi waktu dan biaya pengujian secara keseluruhan.

Implementasi pengujian *white box* dapat dilakukan dengan menggunakan berbagai alat dan teknik, seperti pengujian mutasi, pengujian jalur kontrol, dan pengujian kriteria keputusan. Selain itu, pengujian *white box* dapat diintegrasikan dengan pengujian *Black Box* untuk meningkatkan kualitas pengujian secara keseluruhan.

Berikut di bawah ini adalah data pengujian *White Box* pada *Chatbot (ALICE)* yang dikembangkan oleh peneliti:

1. Modul Integrasi *API Chatgpt*

**Tabel 4.5 Modul Integrasi *API Chatgpt***

|  |  |
| --- | --- |
| ***Flowchart*** | ***Flowgraph*** |
|  |  |

Dari Gambar 4.5 diatas, menunjukan *flowchart* dan *flowgraph* sudah tersedia *cyclomatic complexity* dari modul integrasi *API Chatgpt*, dapat dibuat dengan menggunakn rumus di bawah ini:

V(G) = Jumlah *Predicate Node* + 1

= 0 + 1

= 1

V(G) = E – N + 2

= 8 – 9 + 2

= 1

V(G) = Jumlah *Region*

= 0

Keterangan:

E = Jumlah busur (*link*)

N = Jumlah Simpul (*node*)

Dari perhitungan diatas ditentukan jalur independen pada modul tersebut:

*Path* 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

1. Modul Intergrasi *Audio Player Automatic*

**Tabel 4.6 Modul Intergrasi *Audio Player Automatic***

|  |  |
| --- | --- |
| ***Flowchart*** | ***Flowgraph*** |
|  |  |

Dari Gambar 4.6 diatas, menunjukan *flowchart* dan *flowgraph* sudah tersedia *cyclomatic complexity* dari modul integrasi *Audio Player Automatic*, dapat dibuat dengan menggunakn rumus di bawah ini:

V(G) = Jumlah *Predicate Node* + 1

= 0 + 1

= 1

V(G) = E – N + 2

= 5 – 5 + 2

= 2

V(G) = Jumlah *Region*

= 0

Keterangan:

E = Jumlah busur (*link*)

N = Jumlah Simpul (*node*)

Dari perhitungan diatas ditentukan jalur independen pada modul tersebut:

*Path* 1 = 1 – 2 – 3 – 4 – 5

### 4.4 *User Response* (Kuesioner)

Pengujian *User Response* melibatkan pengguna yang sesuai dengan profil target pengguna sistem atau aplikasi. Pengujian dilakukan dengan memberikan tugas atau skenario kepada pengguna dan mengamati bagaimana mereka menggunakan sistem serta merespon interaksi dengan sistem tersebut. Metode pengumpulan dara yang umum digunakan dalam pengujian ini meliputi pengamatan langsung, kuesioner, dan catatan reflektif.

Berdasarkan data dari hasil kuesioner yang didapat, dapat dicari presentase masing-masing jawaban dengan menggunakan rumus:

X = (Y/Z) \* 100%

Dimana:

X = Banyaknya jawaban responden tiap soal

Y = Skor tertinggi

Z = Nilai Presentase

Kuesioner memiliki (banyak responden….) orang dengan 5 buah pertanyaan dan di setiap pertanyaan memiliki 5 buah pilihan jawaban. Perhitungan dilakukan dengan menggunakan skala likert 5 dimana data dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring setiap jawaban dari responden lalu kemudian dijumlahkan. Untuk mengetahui interpretasi skor dari setiap responden, dapat dilihat pada tabel berikut:

**Tabel 4.7 Skala Perhitungan**

|  |  |  |
| --- | --- | --- |
| Skala | Nilai Skala | Keterangan |
| 0% - 20% | 1 | Sangat Tidak Setuju |
| 21% - 40% | 2 | Tidak Setuju |
| 41% - 60% | 3 | Cukup Setuju |
| 61% - 80% | 4 | Setuju |
| 81% - 100% | 5 | Sangat Setuju |

Kemudian, setelah data responden terkumpul, nilai Skor Kriterium dapat dihitung dengan menggunakan rumus berikut:

Skor Kriterium = Nilai Skala x Jumlah Responden

Contoh perhitungan dengan total responden sebanyak 20 responden, sebagai berikut:

**Tabel 4.8 Contoh Perhitungan Persentase**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Keterangan | Nilai Skala | Responden | Skor | Persentase |
| 1. | Sangat setuju | 5 | 15 | 75 | (93\*100))/100 = 93% |
| 2. | Setuju | 4 | 3 | 12 |
| 3. | Cukup setuju | 3 | 2 | 6 |
| 4. | Tidak setuju | 2 | 0 | 0 |
| 5. | Sangat tidak setuju | 1 | 0 | 0 |
| Jumlah | | | 20 | 93 |

Berdasarkan contoh Tabel 4.8 di atas, total skor yang diperoleh adalah 93%, Hasil ini menunjukkan bahwa nilai presentasi responden mencapai 93% dari nilai yang diharapkan, yaitu 100%. Untuk pembuatan kesimpulan dapat dibuat berdasarkan hasil perhitungan kuesioner tersebut.

Berikut ini merupakan perhitungan hasil pengujian Implementasi *Chatbot* Dengan Memanfaatkan *API Chatgpt* Melalui *Gradio App* Menggunakan Metode *Natural Language Processing (NLP)* oleh (banyak responden…) responden yang diperoleh:

## BAB V PENUTUP

### 5.1 Kesimpulan

### 5.2 Saran

## DAFTAR PUSTAKA

Chandra, A. A., Nathaniel, V., Satura, F. R., Dharma Adhinata, F., & Studi, P. (n.d.). Pengembangan Chatbot Informasi Mahasiswa Berbasis Telegram dengan Metode Natural Language Processing. In *Jurnal ICTEE* (Vol. 3, Issue 1).

Furqan, M., & Naufal Shidqi, M. (2023). Chatbot Telegram Menggunakan Natural Language Processing. *Walisongo Journal of Information Technology*, *5*(1), 15–26. https://doi.org/10.21580/wjit.2023.5.1.14793

Pratama, R. A. (2022). Pemanfaatan Media Pembelajaran 3 Dimensi Untuk Materi Kecerdasan Buatan Dalam Mata KuliahKecerdasan Buatan. *https://jurnal.polsri.ac.id/*, 4.

Afriansyah, R., Sholeh, M., Andayati, D. (2021). Perancangan Aplikasi Pemrograman Antarmuka Berbasis Web Menggunakan Gaya Arsitektur Representasi Untuk Sistem Presensi Sekolah, 9(1), 84-93.

Setiawan, A. (2023). Penggunaan ChatGPT Untuk Pendidikan di Era Education 4.0. *PETISI*, 57.

Husamuddin, H. (2020). Otomatisasi Layanan *Frequensi Ask Questions* Berbasis *Natural Language Processing* pada Telegram Bot, 148.

Astiningrum, M. (2018). Implementasi NLP dengan Konversi Kata pada Sistem Chatbot Konsultan Laktasi. *Jurnal Informatika Polinema*, 47.

Clinton, R. M. (2019). Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini - Komputer Raspberry Pi. *Jurnal Teknik Elektro dan Komputer*, 183.

Permana, A. Y. (2019). Perancangan Sistem Informasi Penjualan Perumahan Mengunakan Metode Sdlc Pada Pt. Mandiri Land Prosperous Berbasis Mobile. *Jurnal Pelita Bangsa*, 155.

Gutama, R. (2021). Implementasi Scrum Pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring Dan Evaluasi Pembangunan (SMEP). Yogjakarta: Universitas Islam Indonesia.

Ningrum, F. C. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. Jurnal Informatika Universitas Pamulang, 126.

Kurniawati. (2021). Penerapan Metode Waterfall Untuk Perancangan Sistem Informasi Inventory Pada Toko Keramik Bintang Terang. *PROSISKO*, 48.

Murcahyanto, H. (2023). Penerapan Media Chat GPT pada Pembelajaran Manajemen Pendidikan terhadap Kemandirian Mahasiswa. *Edumatic*, 117.

Putra, D. W. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. *TEKNOIF*, 33.

Aliman, W. (2021). Perancangan Perangkat Lunak Untuk Menggambar Diagram Berbasis Android. *Syntax Literate*, 3094.

Achlison, U. (2020). Analisis Implementasi Pengukuran Suhu Tubuh Manusia dalam Pandemi Covid-19 di Indonesia. Jurnal Ilmiah Komputer Grafis, 103.

Arianti, T. (2022). Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram Uml (*Unified Modelling Language*). Jurnal Ilmiah Komputer Terapan Dan Informasi (Jikti), 20.

Prasetya, A. F. (2022). Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (*Unified Modelling Language*). Jurnal Ilmiah Komputer Terapan dan Informasi (JIKTI), 15.

Rosalina, A. (2020). Pengujian *Black Box* pada Sistem Informasi Penjualan HI Shoe Store Menggunakan Teknik *Equivalence Partitions*. Jurnal Informatika Universitas Pamulang, 26.

Purnomo, E. (2022). Pengembangan Aplikasi Alat Kesehatan Menggunakan Teknologi Realitas Tertambah Berbasis Android (Studi Kasus: Farouk Farisi Medika). *KALBISIANA*, 2908.

Kalsum, U. (2022). Pengenalan Kecerdasan Buatan (Artificial Intelligence) Kepada Para Remaja. *Jurnal Teknik Informatika*, 2.

Muhyidin, M. A. (2020). Perancangan *Ui/Ux* Aplikasi My Cic Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi *Figma*. *JURNAL DIGIT*, 211.

Sitinjak, D. D. (2020). Analisa Dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris Pada Intensive English Course Di Ciledug Tangerang. *Jurnal Ipsikom*, 7.

Firmansyah, M. D. (2021). Analisa dan Perancangan Web E- Commerce Berbasis Website pada Toko Ida Shoes. *Journal of Information System and Technology*, 68.

Hartiwati, E. N. (2022). Aplikasi Inventori Barang Menggunakan Java Dengan Phpmyadmin. *Cross-Border*, 602.

Prahast, i. K. (2022). Aplikasi Penilaian Sekolah Adiwiyata Pada Badan Lingkungan Hidup (BLH) Menggunakan Bahasa Pemrograman Basic dan Database MySQL. *Jurnal Media Infotama*, 376.

Gamas, A. W. (2022). Desain User Interface Website Pemetaan Tanaman Obat Dan Langka Di Kabupaten Kediri Dengan Menggunakan Figma. *Bulletin of Information Technology (BIT)*, 281.

Londjo, M. F. (2021). Implementasi *White Box* Testing Dengan Teknik Basis *Path* Pada Pengujian *Form Login*. *Jurnal Siliwangi*, 36.

Mardiana, Y. F. (2024). Implementasi Transformasi Digital dan Kecerdasan Sebagai Inovasi Untuk UMKM pada Era Revolusi Industri 4.0. *JUMANAGE*, 267.

Nurjanah, E. (2023). *Ai-Based Chatbot Application As Learning Media On Writing Text Description Material*. *ICOE*, 488.

Muhammad, I. R. (2024). Pengembangan Backend Server Berbasis Arsitektur *REST API* pada Sistem *Transfer* Dompet Digital. *SNATI*, 80.

*George*, H. G. (2023). *A Review of ChatGPT AI's Impact on Several Business Sectors*. *Partners Universal International Innovation Journal (PUIIJ)*, 9.

Aswiputri, M. (2021). *Literature Review Determinasi* Sistem Informasi Manajemen: *Database, Cctv Dan Brainware*. *JEMSI*, 315.

Simanullang, P. M. (2021). Pengaruh Perangkat Keras Komputer Dalam Sistem Informasi Manajemen. *OSF*, 34.